remote sensing

MDPI

Article

A Modified Iteration-Free SPGA Based on Removing the Linear Phase

Yi Xie [1,2], Yuchen Luan [1], Longyong Chen [1] and Xin Zhang [1,*]

[1] National Key Laboratory of Microwave Imaging Technology, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100190, China; xieyi20@mails.ucas.ac.cn (Y.X.); luanyc@aircas.ac.cn (Y.L.); chenly@aircas.ac.cn (L.C.)

[2] School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100094, China

* Correspondence: zhangxin000181@aircas.ac.cn

Abstract: In traditional Stripmap SAR imaging,the platform motion error will bring the phase error in the azimuthal direction to the image, which will have a series of effects on the imaging quality. The traditional autofocus algorithm—Stripmap Phase Gradient Algorithm (SPGA)—can estimate any order phase error above the second order in theory, but it is difficult to estimate the linear phase error, which leads to the discontinuity of the estimated phase error. It usually needs multiple iterations to focus an image, which is inefficient. Moreover, because the linear phase error cannot be estimated, the traditional SPGA cannot eliminate the target offset in the image, resulting in the distortion of the image in the azimuthal direction. According to the continuity of phase error, we propose a modified iteration-free SPGA based on removing the linear phase. Without iteration, the proposed autofocus algorithm can achieve comparable or even better results than traditional SPGA. In the simulation experiments, piecewise linear errors are added to the images of multiple targets. SPGA still fails to focus the image after six iterations. The average ILSR and ILSR are −7.11 dB and −3.99 dB, respectively, and the average number of point target drift is 8.42 pixels. The proposed algorithm optimizes the average ILSR and ILSR to −12.34 dB and −9.87 dB and reduces the average number of point target drift to 0.16 pixels. In the actual data processing, using image entropy as the evaluation criterion, the time consumption is only 19.25% of SPGA under the condition of achieving the same focusing quality.

Keywords: phase gradient algorithm; linear phase error; autofocus; iteration-free

Citation: Xie, Y.; Luan, Y.; Chen, L.; Zhang, X. A Modified Iteration-Free SPGA Based on Removing the Linear Phase. *Remote Sens.* **2023**, *15*, 5535. https://doi.org/10.3390/rs15235535

Academic Editor: Timo Balz

Received: 20 August 2023
Revised: 20 November 2023
Accepted: 21 November 2023
Published: 28 November 2023

1. Introduction

Synthetic Aperture Radar (SAR) has been widely used in military reconnaissance, terrain mapping, environmental monitoring, and other fields [1,2] because of its all-weather ability. In the Stripmap SAR imaging algorithm, it is considered that the radar platform moves uniformly in a straight line, but in reality, due to many factors such as airflow disturbance and piloting error, the track will be an irregular curve.

Thus, phase errors are added in azimuth and distance direction, causing defocusing, distortion, and other phenomena affecting image quality [3,4]. Flight trajectory and attitude data provided by INS and GNSS are usually used for motion compensation [5,6] to eliminate serious errors. However, due to the limitation of the accuracy of the inertial data and vibration of the load itself, motion compensation cannot eliminate all errors. In order to solve this problem, it is necessary to use autofocus technology; that is, to use the characteristics of echo data itself for motion error compensation so that the image can be further focused [7]. Figure 1 shows where motion compensation (MoCo) [8] and autofocus technology are typically used, and to what extent they improve image quality.

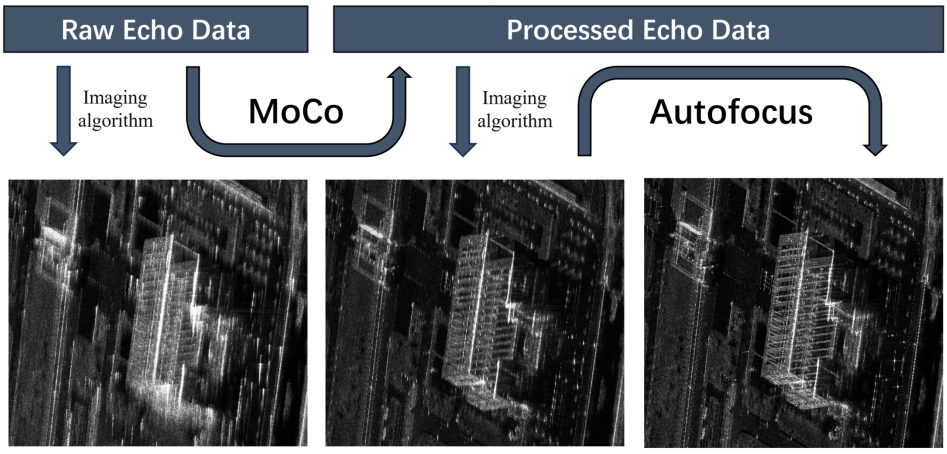

Figure 1. MoCo and autofocus using scene and effect comparison. (**a**) Raw Image; (**b**) Image after Moco; (**c**) Image after Autofocus.

According to the modeling method of motion error, standard autofocus algorithms are usually divided into two categories: parametric method and non-parametric method.

(1) Parametric autofocus method: Parametric autofocus methods usually build a parametric model for motion errors and then use algorithms to estimate the parameters of the model, mainly including Map-Drift (MD) [9] and Shift-And-Correlate (SAC) [10]. Their algorithms are simple and fast. However, it is only effective for quadratic phase errors, but not for linear phases and higher-order phases (above the second order) [11]. In addition, there is an autofocus algorithm based on image optimization, which uses specific criteria to search a set of optimal parameters for the motion error model according to different actual requirements to achieve the best image quality. The criteria mainly include contrast optimization (CO) [12], sharpness [13–15], and minimum entropy (ME) [16–18]. Since there is no analytical mapping relationship between image quality and model parameters, the iterative search process is needed to meet the constraints in image optimization methods. The larger the phase error order, the greater the computational complexity and the longer the time cost.

(2) Non-parametric autofocus method: The non-parametric method does not require a parametric model of the motion error because it extracts the phase or phase gradient of the motion error directly from the raw radar data. Its main representative algorithm is the Phase Gradient Algorithms (PGA) [19–21], which does not depend on the specific model and can estimate the phase error from the characteristics of the data itself. Theoretically, the phase error of the second order or any order above can be estimated through iteration. Because of this advantage, various improved algorithms based on it have been widely studied in recent years, but there are still some problems that have not been overcome. For example, the Stripmap Phase Gradient Method (SPGA) [22,23], which is suitable for strip mode SAR. It introduces the PGA algorithm into strip SAR imaging. Although it has the advantages of being simple and easy to implement, it still does not make up for the shortcomings of PGA in essence, and it is still unable to directly estimate the linear phase error. The existence of linear phase error will lead to the drift of the target [24,25]. The SPGA algorithm will extract and splice the phase error of the point target at different positions. When the phase history of the image is too long, the linear terms of the phase error in different regions will be different, resulting in discontinuities at the phase stitching and large errors in the coincidence part, which further leads to unfocused images and requires several iterations to eliminate this effect [26]. On the one hand, multiple iterations will consume a lot of computing resources and reduce the focusing efficiency, making it difficult to achieve real-time processing. On the other hand, it is still impossible to ensure that the global linear error residual is at the same level after iteration. As a result, although

the image is focused, it will produce distortion in the azimuthal direction, which leads to a series of problems in some applications, such as SAR scene matching navigation technology [27,28], where image distortion will affect the navigation accuracy. For the above problems, under normal circumstances, SPGA first subtracts the average value of the phase error gradient of the adjacent two points, takes the average value of the phase error of the coincidence part when stitching the phase error, and then eliminates the influence of linear error as much as possible through iteration. However, the effectiveness of this approach is very limited.

In this paper, we analyze the effect of linear phase on SPGA; that is, it will cause image distortion and increase the number of iterations. We propose a modified autofocus algorithm based on SPGA that can remove linear phase error. We use phase continuity to estimate and restore the real relative position of the strong point target so that the linear error can be preserved in the phase error extraction stage. Because the real position of the strong point can be estimated, the phase error can be restored to the greatest extent so that there will be no phase discontinuity in the phase splicing stage, which theoretically through one iteration can achieve the ideal effect. Compared with the traditional SPGA algorithm, the proposed algorithm only uses a small computational cost to achieve the goal of no iteration and no image distortion, which is helpful to SAR real-time autofocus imaging.

2. Theory

As shown in Figure 2, the trajectory of the aircraft will be curved due to airflow disturbance or self-vibration, which will bring phase error to the azimuth data, and it will change with the azimuth time. Assume that the residual phase error after MoCo is $\varphi(t)$.

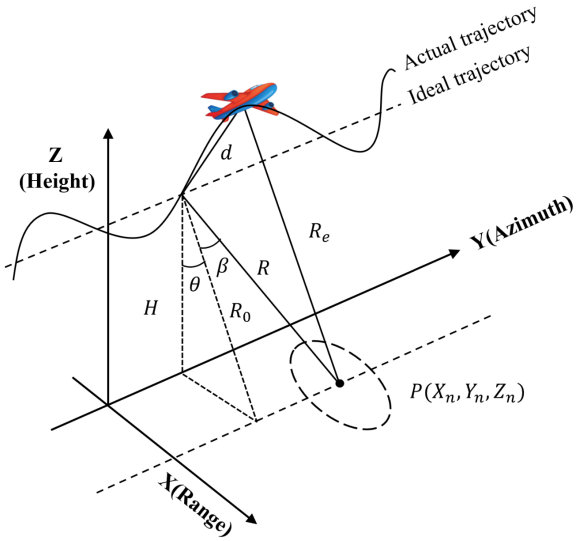

Figure 2. Schematic of the ideal flight trajectory and the actual trajectory of the SAR system.

To explore why SPGA requires multiple iterations and cannot estimate linear phase, start with a mathematical process of point target focusing and defocusing. Let the peak of a point target be at t_0 and its expression before pulse compression be:

$$S(t) = \exp\left\{-j\pi K(t - t_0)^2\right\} \tag{1}$$

When it is disturbed by a phase error $\varphi(t)$, there is:

$$S_e(t) = \exp\left\{-j\pi K(t - t_0)^2 + j\varphi(t)\right\} \tag{2}$$

As shown in Figure 3, the introduction of phase error will have many effects on pulse compression results, such as main lobe broadening, side lobe rising, side lobe dissymmetry,

and peak point drift. These phenomena are reflected in the image as defocusing and distortion. When we know the exact working parameters such as t_0 and K, this process is reversible; that is, we can extract $\varphi(t)$ through inverse compression and dchirp operation and compensate it to the original formula. You can obtain the focused pulse compression signal again. ($PC\{\}$ indicates the pulse compression operation).

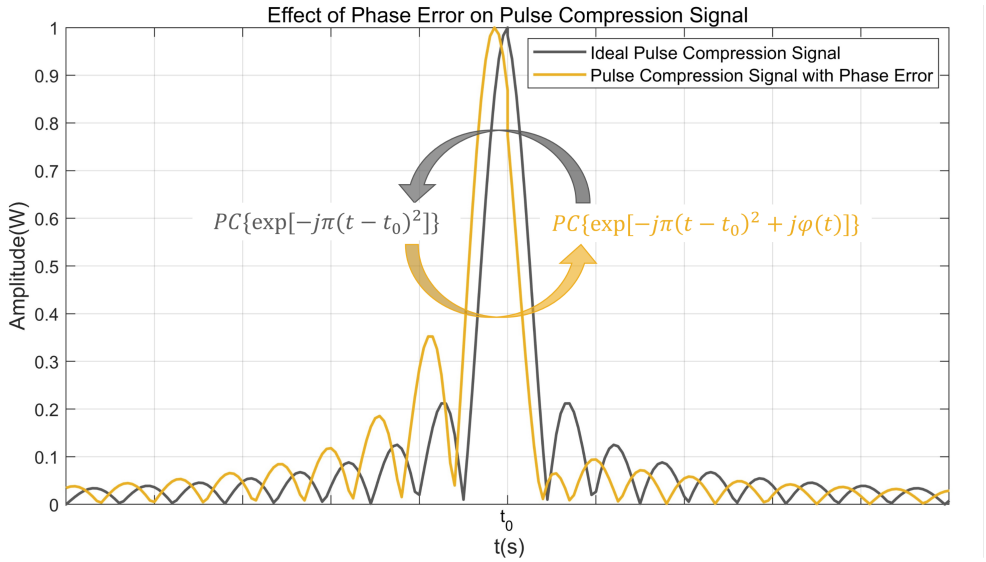

Figure 3. Schematic diagram of the principle of SPGA.

It is fundamental is to inversely solve the phase error gradient by using the point target characteristics in the image, obtain the phase error surface through integration and fitting, and focus the image by compensating the phase errors.

In theory, it can estimate any error above the second order through iteration, but it is difficult to estimate the linear phase because the linear phase is reflected in the shift of the point target in the orientation, which prevents us from getting the right t_0. When carrying out the Legendre polynomial expansion on the phase error and listing the first-order term only, the higher-order terms represented by $\varphi_0(t)$, we can obtain:

$$\varphi(t) = a_0(t - t_0) + \varphi_0(t - t_0) \tag{3}$$

substituting the above Equation (3) into Equation (2), we obtain:

$$S_e(t) = \exp\left\{ j\left[-\pi K\left(t - t_0 + \frac{a_0}{2\pi K} \right)^2 + \frac{a_0^2}{4\pi K} + \varphi_0(t - t_0) \right] \right\} \tag{4}$$

By observing Equations (2) and (4), we see that the real position of the peak point is t_0, but due to the existence of the linear component of phase error, its peak point will be shifted to $t_0 - \frac{a_0}{2\pi K}$. This will cause us to be unable to confirm the true value of t_0. Since a_0 is an indeterminate constant, this will cause us to be unable to confirm the true value of t_0. This will further cause the extracted phase error information to change from $\varphi(t)$ to $\varphi(t) - a_0(t - t_0) + \frac{a_0^2}{4\pi K}$, thus losing the linear phase information. That is, if the correct point target position cannot be selected, the linear error cannot be extracted, and the true position of the origin cannot be returned if the linear error cannot be estimated. The two problems affect each other.

As shown in Figure 4, suppose there are two point targets whose positions are t_1 and t_2, respectively, and their synthetic aperture overlapped part is AB. $\varphi(t)$ (green solid line) is the true phase error passing through the two point targets. According to Equations (3) and (4), the estimated phase errors by SPGA are $\frac{a_1^2}{4\pi K} + \varphi_1(t - t_1)$ (blue dotted line) and $\frac{a_2^2}{4\pi K} + \varphi_2(t - t_2)$ (yellow dotted line), respectively. The concatenated phase error

curves are $\varphi_\varepsilon(t)$ (red solid line). It can be seen that due to the absence of the linear phase, there will be a large difference between the estimated phase error curve and the actual phase error.

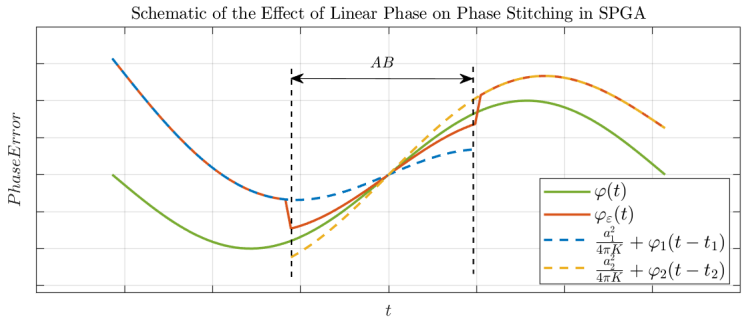

Figure 4. Schematic of the effect of linear phase on phase stitching in SPGA.

3. Method

In the process of processing a lot of real data, we found that under most conditions, due to the influence of linear phase error, the point target will only offset within a certain range of azimuthal direction, which means that the real position of the point target is around the selected point. As long as we go through all the points within a certain range around the selected point, the point closest to the real position can always be screened out under certain conditions. According to the continuity of the phase error, the motion error of the overlapped parts should be the same. When the selected point is correct, the coincidence degree of the phase error of the overlapped parts of two adjacent synthetic apertures should be the highest; that is, the phase difference in the overlapped part of the phase error function is the smallest. In other words, the phase error function has the greatest correlation. Therefore, for each point target selected, the point within a certain range of its azimuthal direction is taken as the alternative point of its real position. Then, through the coincidence part of the phase error of adjacent points, the error matching is carried out and the point closest to the real position is screened out from the alternative points so as to extract the linear phase error of this point. Figure 5 is a flow chart of our approach.

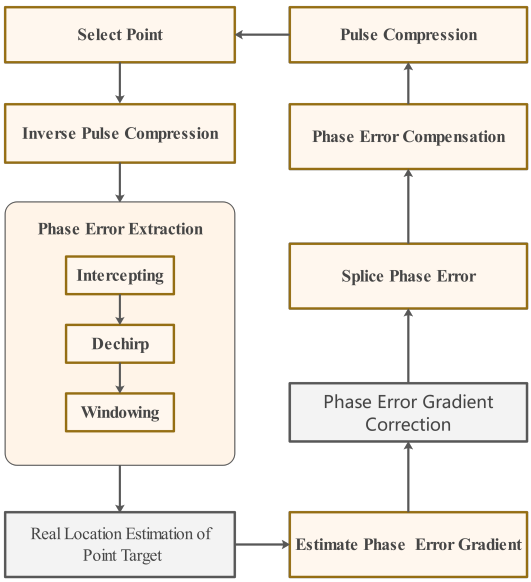

Figure 5. Main steps of our approach.

3.1. Point Selection

The extraction of the linear phase depends on the phase error function of the coincidence part of the synthetic aperture of two adjacent points. Therefore, in the process of point selection, in addition to ensuring the good characteristics of the point target, it should be evenly distributed in the entire imaging region as far as possible so that the two adjacent points have overlapping parts as far apart as possible, so as to meet the conditions of phase error matching.

In many cases, due to the different scattering characteristics of ground objects, the strong points in the image may be concentrated in a few range gates or several range units. Using the maximum peak point of the range gate or range unit as the point target will filter out part of the ideal point target so that point targets are not evenly distributed in the azimuthal direction. Therefore, this paper uses $2D$ CFAR detection results [29,30] instead of the maximum range element for preliminary screening and uses the contrast rule for secondary filtering. That is, the amplitude of the LFM signal generated by the ideal point target is considered consistent within the synthetic aperture, as shown in Equation (5).

$$Q_k = 1 - \frac{\left[\frac{1}{T_S}\int_{-\frac{T_S}{2}}^{\frac{T_S}{2}} |S_k(t)|dt\right]^2}{\frac{1}{T_S}\int_{-\frac{T_S}{2}}^{\frac{T_S}{2}} |S_k(t)|^2 dt} \tag{5}$$

$S_k(t)$ is the LFM signal, T_S is the synthetic aperture time, less than a synthetic aperture according to the LMF signal duration at this point. k is the number of point selection results of 2D $CFAR$.

3.2. Phase Error Extraction

This part is not the key point of the article, but aims to briefly introduce the basic concepts and functions of the related operations of phase error extraction and describe the brief process from the point target to the phase error. For a more detailed introduction, see paper [7].

(1) Intercepting: When the point target is determined, the data of one synthetic aperture length are intercepted in the azimuthal direction with the point target as the center. When the data are less than one synthetic aperture, the excess part is discarded.

(2) Inverse pulse compression: It is the inverse process of pulse compression. Let the azimuthal coordinate of the image be u, the range coordinate be v, and the coordinates of the selected point be (u_0, v_0). The signal intercepted centered at this point is $X_0(u)$. The reference function of the range gate is $H(f_a)$, and f_a is the azimuth frequency domain coordinate. The signal after inverse pulse compression is $S_0(t)$, where t is the azimuth time domain coordinate. They have the following equation:

$$S_0(t) = IFFT\{FFT\{X_0(u)\} \cdot H(f_a)\} \tag{6}$$

(3) Dechirp: $S_0(t)$ is the superposition of the chirp signal with phase error and the chirp signal of the surrounding point target. "Dechirp" is to remove the LFM signal and keep the phase error information. Let the signal after "Dechirp" be $\Phi_0(t)$:($conj\{A\}$ means take the conjugate of A)

$$\Phi_0(t) = S_0(t) \cdot conj\{FFT\{H(f_a)\}\} \tag{7}$$

(4) Windowing: $\Phi_0(t)$ contains all the phase error information of the selected point target, but due to the influence of the information of other nearby point targets and the influence of noise, it needs "Windowing" in order to eliminate these effects as much as possible.

3.3. Real Location Estimation of Point Target

There are N selected point targets in the point selection step. In order to facilitate the analysis, only two adjacent selected point targets (hereinafter referred to as the main points) in the same range gate are considered, and their synthetic apertures overlap. It is assumed that the two main points have been shifted by within $\pm W/2$ points in the azimuthal direction due to the influence of linear phase error (W is set as an even number for easy calculation). The synthetic aperture time is T_s, the range of azimuth time is $[-T, T]$, and the azimuth tuning frequency is K. Without the influence of the linear phase error component, the azimuth time of the two main points P_1 and P_2 are t_1 and t_2, and the azimuth peaks of them fall at $u1$ and $u2$. AB is the phase overlap region of two main points. When it experiences a certain phase error $\varphi(t)$, the two main points can be obtained after interception and inverse compression:

$$\begin{cases} \mathrm{S}_1(t) = \exp\left\{-j\pi K(t-t_1)^2 + \varphi(t)\right\}, t \in \left[t_1 - \frac{T_s}{2}, t_1 + \frac{T_s}{2}\right] \\ \mathrm{S}_2(t) = \exp\left\{-j\pi K(t-t_2)^2 + \varphi(t)\right\}, t \in \left[t_2 - \frac{T_s}{2}, t_2 + \frac{T_s}{2}\right] \end{cases} \tag{8}$$

Assume that the time corresponding to the true position of the point $u1$ and $u2$ is t_{u1} and t_{u2}. The ideal reference signals generated based on them are:

$$\begin{cases} \mathrm{S}_{ref1}(t, t_{u1}) = \exp\left\{-j\pi K(t-t_{u1})^2 + \varphi(t)\right\}, t \in \left[t_1 - \frac{T_s}{2}, t_1 + \frac{T_s}{2}\right] \\ \mathrm{S}_{ref2}(t, t_{u2}) = \exp\left\{-j\pi K(t-t_{u2})^2 + \varphi(t)\right\}, t \in \left[t_2 - \frac{T_s}{2}, t_2 + \frac{T_s}{2}\right] \end{cases} \tag{9}$$

After "dechirp" [7] using Equations (8) and (9), let $\Delta t_1 = t_1 - t_{u1}$ and $\Delta t_2 = t_2 - t_{u2}$, the phase error function can be obtained:

$$\begin{cases} \Phi_1(t) = \exp\left\{j\pi K\left[2\Delta t_1(t-t_1) + \Delta t_1^2\right] + \varphi(t)\right\}, t \in \left[t_1 - \frac{T_s}{2}, t_1 + \frac{T_s}{2}\right] \\ \Phi_2(t) = \exp\left\{j\pi K\left[2\Delta t_1(t-t_2) + \Delta t_2^2\right] + \varphi(t)\right\}, t \in \left[t_2 - \frac{T_s}{2}, t_2 + \frac{T_s}{2}\right] \end{cases} \tag{10}$$

Conjugating the multiplication of overlapped parts is denoted as:

$$H_{12}(t) = \Phi_1(t) * \Phi_2^*(t), t \in \{\text{ overlapping parts }\} \tag{11}$$

According to the continuity of the phase error, $Phase\{\Phi_1(t)\}$ and $Phase\{\Phi_2(t)\}$ should be highly coincident. The interrelationship among $Phase\{\Phi_1(t)\}$, $Phase\{\Phi_2(t)\}$, and $\varphi(t)$ is shown in Figure 6.

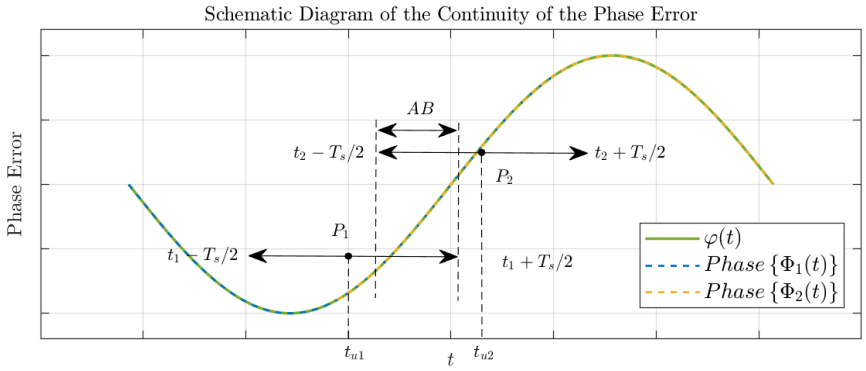

Figure 6. Schematic diagram of the continuity of the phase error.

Which also implies the following relationship: Phase$\{\cdot\}$ is the phase-taking operation.

$$\text{Phase}\left\{H_{12}(t)\right\} = 0, t \in \{\text{ overlapping parts }\} \tag{12}$$

That is:

$$2\pi K(\Delta t_1 - \Delta t_2)t + 2\pi K\left[\Delta t_1^2 - \Delta t_2^2 - 2t_1\Delta t_1 + 2t_2\Delta t_2\right] = 0, t \in \{\text{ overlapping parts }\} \tag{13}$$

As can be seen from the above equation, it is a line with slope $2\pi K(\Delta t_1 - \Delta t_2)$. In order to satisfy the above equation, it needs to satisfy $\Delta t_1 = 0$ and $\Delta t_2 = 0$. In fact, in most cases, due to azimuth sampling, t_{u1} is discontinuous, whereas t_1 is a continuous variable. This causes Δt_1 to float between $[0, 1/prf]$. It is very rare that $\Delta t_1 = 0$ and $\Delta t_2 = 0$. According to the above assumptions, when $u1$ and $u2$ are the real positions, let:

$$r_{12} = 2\pi K(\Delta t_1 - \Delta t_2) \tag{14}$$

Then:

$$|r_{12}| \leq \frac{4\pi K}{prf} \tag{15}$$

where prf is pulse repetition frequency, namely azimuth sampling rate.

That is, after conjugate multiplication of the phase error of the overlapped parts of two adjacent points, its phase is linear, and the slope should meet the above equation. According to this condition, taking P_1 as the origin of the pixel, the $u1$ traversal $[u1 - W/2, u1 + W/2 - 1]$ is taken, and the step length is set as 1 (W is called the redundant range of selection points), and they are, respectively, denoted by $u1(1), u1(2), u1(3)$, $...u1(W)$, and the corresponding azimuth time is denoted by $t_{u1(1)}, t_{u1(2)}, t_{u1(3)}, ...t_{u1(W)}$. Then we repeat R the same operation for P_2 and select the combination of $u1(i)$ and $u1(i)$ that satisfies Equation (15) and where $|r_{12}|$ is minimum $u1(i), u2(j)$ is the estimated true position of the point. The matrix can be expressed as:

$$R_{12} = R\left\{ \text{Phase}\left\{ \begin{bmatrix} S^*_{ref1}\left(t, t_{u1(1)}\right) \\ S^*_{ref1}\left(t, t_{u1(2)}\right) \\ \vdots \\ S^*_{ref1}\left(t, t_{u1(W)}\right) \end{bmatrix} S_1(t)S_2^*(t) \begin{bmatrix} S^*_{ref2}\left(t, t_{u2(1)}\right) \\ S^*_{ref2}\left(t, t_{u2(2)}\right) \\ \vdots \\ S^*_{ref2}\left(t, t_{u2(W)}\right) \end{bmatrix}^H \right\} \right\} \tag{16}$$

where R$\{\cdot\}$ is defined as finding the element slope and Phase$\{\cdot\}$ is defined as finding the element phase.

In the matrix R_{12}, r_{12} is the smallest element satisfying Equation (15), i and j are the row index and column index, respectively, and $u1(i)$ and $u2(j)$ are the real positions of P_1 and P_2. For the sake of understanding, Figure 7 shows the typical relationship of R_{12} between t_{u1} and t_{u1}. We can find the minimum value of R_{12} in curve $t_{u2}(3)$, whose $t_{u1} - coordinate$ is 22, and its value satisfies Equation (15), so here i is 22 and j is 3. If there is no element satisfying Equation (15) in R_{12}, it means that the range W is too small, and W and the traversal step size can be increased appropriately.

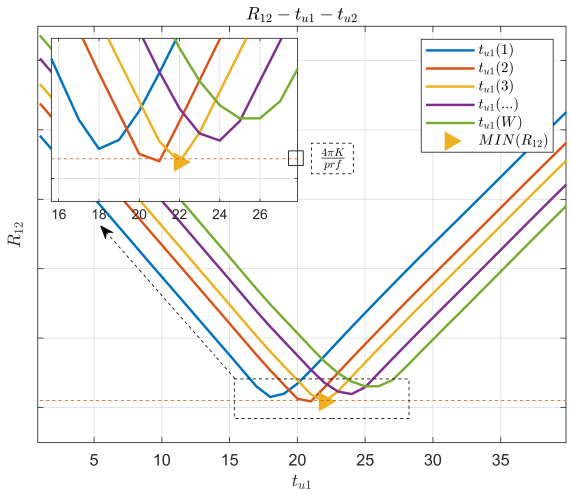

Figure 7. Typical relation of R_{12} to t_{u1} and t_{u2}.

3.4. Phase Error Estimation

Taking P_1 as an example, the Legendre polynomial expansion of $\phi(t)$ in Equation (10) at t_1 is performed [31] to list the terms only once.

Let:

$$\varphi(t) = a_n \cdot (t - t_n) + \varphi_n(t - t_n), n = 1, 2, 3, \ldots, N \tag{17}$$

We can figure out:

$$\Phi_1(t) = \exp\left\{ j\pi K \left[2\Delta t_1 (t - t_1) + \Delta t_1^2 \right] + a_1(t - t_1) + \varphi_1(t - t_1) \right\} \tag{18}$$

The phase error estimator of P_1 can be obtained by differentiating and integrating the phase of the above equation:

$$\varphi_{\varepsilon(1)}(t) = (2\pi K \Delta t_1 + a_1)(t - t_1) + \varphi_1(t - t_1) \tag{19}$$

It can be seen that the phase error estimated by selecting the real position of the point target can completely retain the linear error in the system, but a new linear error $2\pi K \Delta t_1 (t - t_1)$ is introduced. By compensating the above formula into P_1, we can obtain:

$$S_1(t) = \exp\left\{ -j \left[\pi K (t - t_1 + 2\Delta t_1)^2 \right] \right\} \tag{20}$$

The above algorithm only focuses on the error matching between two adjacent points. In actual processing, the main points need to be sorted from the largest to the smallest according to the azimuth index, and then the above operations should be carried out on each two adjacent points. Except for the first and last points of the figure, all the other points will be estimated twice; that is, two sets of phase error gradients will be obtained when the estimation is accurate. The two sets of phase error gradients should be roughly the same, so the average can be taken in this case.

3.5. Phase Error Correction

It can be observed from Equation (14) that although the original linear phase error is eliminated, the main points will shift by at most two points extra. If the shift value of the origin is much larger than two points, the effect of our method is obvious. If the original shift value is less than two points, it will bring unnecessary errors. In addition, if N main points are selected, the specific azimuth time of the n_{th} main points is recorded as t_n, and the azimuth time of the peak point is t_{un}. Because $\Delta t_n = t_n - t_{un}$, it is easy to know that the shift value of each part will be inconsistent due to the difference of Δt_n after the phase error is combined.

Therefore, two correction methods are proposed:

(1) Interpolation: According to Equation (20), the extra part of the estimated phase error is positively correlated with Δt_n, whereas the maximum value of Δt_n is inversely proportional to the azimuth sampling rate. Therefore, after estimating the true position of the point target, the origin target can be interpolated, and the above operation can be repeated again for more precise estimation, thus reducing the upper limit of Δt_n. This reduces the additional shift value to an acceptable range.

(2) Iterative correction: According to Equations (17) and (19), the phase error gradient of point P_n is:

$$\Delta\varphi_{\varepsilon(n)}(t) = 2\pi K\Delta t_n + a_n + \varphi_n'(t - t_n) \tag{21}$$

By Equation (14), we can obtain $R_{(n)(n+1)} = 2\pi K(\Delta t_n - \Delta t_{n+1})$, which is the slope of the phase curve multiplied by the conjugate of P_n and P_{n+1} phase error curves. The phase error gradient can be corrected by gradually calculating the data.

$$\begin{aligned}\Delta\varphi^*_{\varepsilon(n)}(t) &= \Delta\varphi_{\varepsilon(n)}(t) + \sum_{i=1}^{n} R_{(i)(i+1)} \\ &= 2\pi K(\Delta t_n + \Delta t_1 - \Delta t_2 + \Delta t_2 - \Delta t_3 \cdots + \Delta t_{n-1} - \Delta t_n) + a_n + \varphi_n'(t - t_n) \\ &= 2\pi K\Delta t_1 + a_n \cdot (t - t_n) + \varphi_n'(t - t_n)\end{aligned} \tag{22}$$

It can be observed from the above equation that the corrected error gradient is fixed at $2\pi K\Delta t_1$, which can ensure that the overall offset of the image is consistent in theory and will not distort the target positions of different strong points.

In summary, the key steps of the algorithm in this paper is given in Table 1.

Table 1. The Key Step of Our Approach.

The Key Step of Our Approach
Input: Image with azimuthal phase error $S_{err}(u, v)$, main imaging parameters, redundancy selection point range W.
Output: The focused image $S_{com}(u, v)$.
Description of operator:.
① $chirp\{u_n, v_n\}$: Reference chirp signal generated with $P_n(u_n, v_n)$ as the center point.
② $Phase\{\cdot\}$: Take the phase of the signal.
③ $R\{\cdot\}$: Finding the slope of the line.
④ $Decmp_{az}\{\cdot\}$: Azimuth inverse compression.
⑤ $Cmp_{az}\{\cdot\}$: Azimuth compression.
Calculate global optimal solution:
Step 1: Select N points and arrange them according to the azimuth index from large to small and name them $P_1, P_2, ..., P_N$ in turn. Their coordinates are $(u_1, v_1), (u_2, v_2), ..., (u_N, v_N)$.
Step 2: Take the azimuth echo data $S_n(u)$ of P_n and add Windowing.
Step 3: Estimate the true position of the point target.
Loop the following $(i = -W/2 : W/2 - 1; j = -W/2 : W/2 - 1)$
$\varphi 1_i(u) = S_n(u) \cdot chirp^*\{u_n + i, v_n\}$
$\varphi 2_j(u) = S_{n+1}(u) \cdot chirp^*\{u_{n+1} + j, v_{n+1}\}$
$R(i, j) = R\{Phase\{\varphi 1_i(u). * \varphi 2_j^*(u)\}\}$.
Step 4: Find the maximum value of $\lvert R(i, j)\rvert$ and record its index (i_{max}, j_{max})
$R_n = R(i_{\max}, j_{\max})$.
Step 5: The phase error at each point will be calculated twice and averaged
$\phi_{relay}(u) = \cdot\varphi 2_{j_{max}}(u)$
$\varphi_n(u) = \left[\varphi 1_{i_{max}}(u) + \varphi_{relay}(u)\right]/2$
$\Delta\varphi_n(u) = diff\{\varphi_n(u)\}$.
Step 6: Correct the phase error iteratively.
Loop the following $(m = -1 : n - 1)$
$\Delta\varphi_n(u) = \Delta\varphi_n(u) + R_m$.
Step 7: Loop Step 2 to Step 6 $(n = 1 : N - 1)$.
Concatenate all $\Delta\varphi_n(u)$ and integrate them to obtain $\varphi_n(u)$.
Step 8: Phase error compensation and pulse compression
$S_{com}(u, v) = Cmp_{az}\{Decmp_{az}\{S_{err}(u, v)\} \cdot \exp(-j\varphi(u))\}$.

4. Experimental Results and Analysis

4.1. Simulation Experiments

In this paper, point target simulation data are used to verify the effectiveness of the proposed algorithm and are compared with the traditional SPGA algorithm. The simulation parameters are shown in Table 2.

In the simulation experiment, seven point targets were uniformly generated at different range gates (the image size is 4096 × 4096), and piecewise linear phase errors were added in the whole azimuth (blue line in Figure 8).

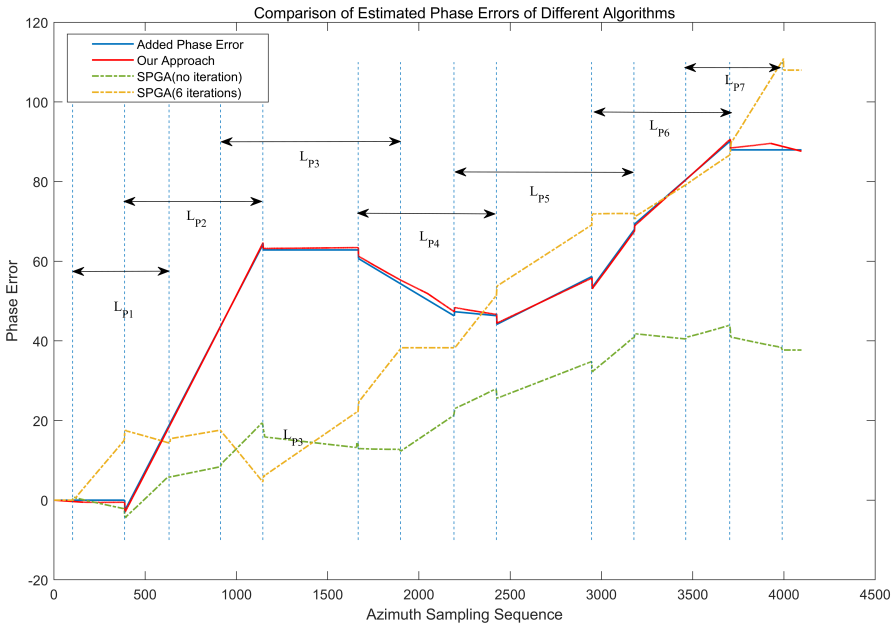

Figure 8. Main process of SPGA.

L_{P_1} to L_{P_7} were the synthetic aperture lengths of points 1 to 7, making the point targets offset by different pixel points, respectively, (see the first row of Table 1). The synthetic aperture of two adjacent main points overlap. It is worth noting that the piecewise linear function is not a linear function, and there are still higher-order components after the Legendre Polynomial Expansion. It will not only bring the drift of the point but also bring the defocusing effect of the point target. The piecewise linear phase error is a relatively extreme case, just to verify the superiority of the algorithm in the processing of linear components in this paper. In Figure 8, the green dotted line is the phase error estimated by SPGA without iteration, whereas the yellow dotted line is the phase error estimated by SPGA after six iterations. The reason why we choose six iterations as the reference object is that under general circumstances [7], six iterations can generally bring good results. It can be seen that the estimated phase error curve is quite different from the added curve, because SPGA cannot estimate the target position of the real point. On the contrary, the red solid line is the phase error extracted by our approach, which can better fit with the added phase error.

Figure 9 illustrates the effects of a representative point target before and after recovery. It can be seen that the traditional SPGA did not estimate the piecewise linear phase error of the point without iteration; not only did it not return the point target to the original position, but it even failed to achieve the focusing effect. After six iterations, although it achieved the focusing effect, it did not restore the original position. In fact, in more cases, it still failed to achieve the focusing effect after multiple iterations. See Table 2 for details. However, the algorithm in this paper not only makes the point target focus but also restores the point target to the original position without iteration. Figure 10 shows the

azimuthal slice, and Table 3 shows the comparison of specific indicators. It can be seen that the proposed method can better estimate the added linear error.

Table 2. The parameters of simulation experiments.

Parameter	Symbol	Value
Pulse repetition frequency	prf	312.5 Hz
Azimuth bandwidth	B_a	285.73 Hz
Speed of flight	V	30.44 m/s
Pulse width	τ	2 us
Beam width	B_w	5 deg
Range direction bandwidth	B_r	400 MHz
Rate of sampling	f_s	600 MHz
Center frequency	f_0	14.6 GHz
Redundant range of selection points	W	40

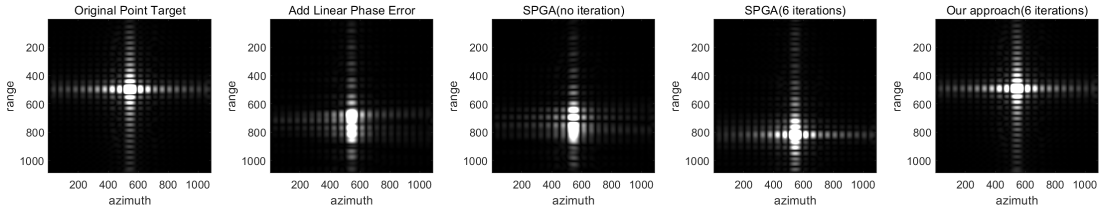

Figure 9. Comparison of focus effect in point target simulation (32 times interpolation).

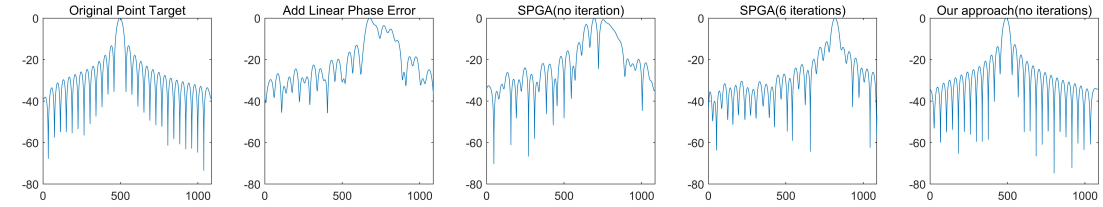

Figure 10. Comparison of simulation focusing effect of point target (azimuth) (32 times interpolation).

Table 3. Comparison of relevant parameters between different algorithms and our approach.

Indicators	Method	P1	P2	P3	P4	P5	P6	P7	Avg
Shift Pixels	A	+0	+8	+0	−4	+4	+8	+0	2.29
	B	+1.15	+7.25	+0.59	−4.65	+1.44	+5.25	+1.96	1.86
	C	+6.37	+13.21	+6.00	+6.00	+10.28	+10.03	+7.04	8.42
	D	−0.02	−0.19	−0.16	+0.03	−0.28	−0.16	−0.34	−0.16
PLSR (dB)	A	−1.75	−13.26	−5.25	−4.00	−4.73	−3.62	−2.34	−4.99
	B	−0.46	−2.18	−5.98	−3.51	−0.93	−0.74	−0.40	−2.03
	C	−3.13	−13.10	−2.09	−0.60	−8.15	−11.34	−11.34	−7.11
	D	−13.10	−13.24	−11.86	−11.68	−12.76	−12.74	−11.06	−12.34
ILSR (dB)	A	−0.18	−10.48	−0.37	−2.40	+0.23	−2.41	−0.61	−2.32
	B	−0.71	+4.75	−3.57	+7.82	−7.42	−4.13	−2.84	−0.87
	C	−0.72	+0.07	−10.09	+7.72	−6.92	−8.46	−9.56	−3.99
	D	−10.30	−10.07	−9.82	−9.73	−9.98	−9.97	−9.22	−9.87

The operation denoted by A is to add a piecewise linear phase error. The operation denoted by B is that the image is processed by SPGA without iteration. The operation denoted by C is that the image is processed by SPGA with six iterations. The operation denoted by D is that the image is processed in our approach without iteration.

Table 3 shows the comparison of point target positions restored by the two methods. It can be seen that the method in this paper can accurately restore the relative positions between the origin targets and ensure that the deviation before and after the point target is basically consistent and within two points. At the same time, the point target characteristics can be better restored.

4.2. Real Data Experiments

In order to further improve the effectiveness of the proposed algorithm, the traditional SPGA and the proposed algorithm are compared and tested. The measured data are from an airborne X-band SAR system. Its working parameters are in Table 4.

Table 4. The parameters of real data experiments.

Parameter	Symbol	Value
Pulse repetition frequency	prf	1000 Hz
Azimuth bandwidth	B_a	346.12 Hz
Speed of flight	V	58.81 m/s
Pulse width	τ	20 us
Beam width	β	8.0 deg
Range direction bandwidth	B_r	100 MHz
Rate of sampling	f_s	400 MHz
Center frequency	f_0	5.4 GHz
Redundant range of selection points	W	40

Figure 11a is the imaging results of an area after working with the working parameters shown in the table above (Caesar window with $\beta = 6$ added). As shown in Figure 11a , after motion compensation, due to the existence of azimuth residual phase error, point target A, point target B, and area C have defocus and offset to different degrees, and area C is between A and B in the azimuthal direction. A and B are selected as reference point targets. Since the point targets in region C are too dense and the available point targets in this scene are sparse, the phase compensation of region C can only depend on the part where the phase errors of point A and point B overlap.

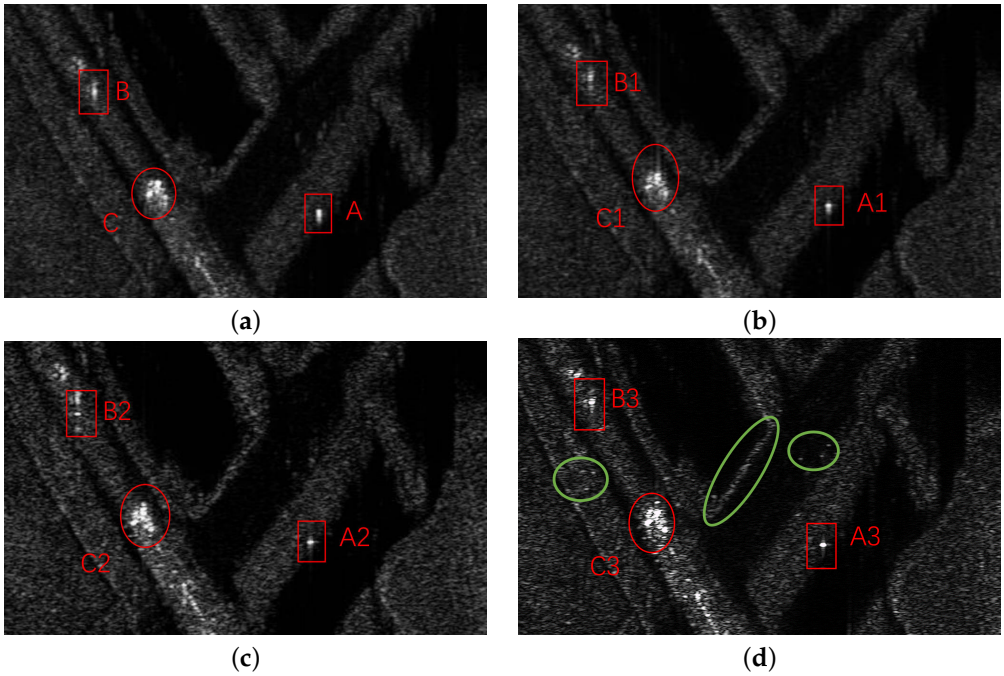

Figure 11. Comparison of focusing effects of different methods. (**a**) Original unfocused image; (**b**) SPGA (no iterations); (**c**) SPGA (6 iterations); (**d**) our approach.

Figure 11b shows the traditional SPGA processing results without iteration, with six iterations, and A1, B1, and C1 are the recovered images of A, B, and C, respectively. It is easy to find that the focusing effect is very limited; A1, B1, and C1 still have a serious defocusing effect.

Figure 11c shows the effect of six iterations of traditional SPGA. It can be seen that, although A2 has a good focusing effect, point B2 does not effectively focus on a point, there are redundant sidelobes around it, and the defocusing phenomenon still exists in area C1.

Figure 11d shows the processing results of the algorithm in this paper. It can be seen that the focusing effect of C3 is obviously better than that of C1 and C2, B3 also has a good result, and the focusing effect of A3 is also better than that of A1 and A. It can be seen that the overall quality is better than Figure 11c.

The targets in the figure are numerous and complex, so we cannot analyze the overall focusing effect from a single point target-related index. In order to quantitatively analyze the performance of the algorithm, as a supplement, Equation (23) is used to calculate the image entropy of different results. Image entropy is a statistical form of image features, reflecting the average amount of information in the image. The smaller the entropy is, the better the focusing effect is. Table 5 shows the image entropy of the four images. At the same time, in order to reflect the computational efficiency of the algorithm, we also show the time cost of the algorithm in the above table. The CPU is the Intel core i7-13700K processor with the memory capacity of 64 GB (Made by Intel, California, USA).

$$\begin{cases} \mathrm{A(I)} = \sum_{u=1}^{U} \sum_{v=1}^{V} |\mathrm{I(u,v)}|^2 \\ \mathrm{P(u,v)} = \frac{|\mathrm{I(u,v)}|^2}{\mathrm{A(I)}} \\ \mathrm{E(I)} = -\sum_{u=1}^{u} \sum_{v=1}^{v} \mathrm{P(u,v)} \ln(\mathrm{P(u,v)}) \end{cases} \tag{23}$$

where $\mathrm{A(I)}$ is the total energy of image I, $P(\mathrm{u,v})$ is the normalized intensity of pixel, and $\mathrm{E(I)}$ is the image entropy of image I.

Table 5. Comparison of result parameters of different processing methods.

Parameter	Figure 11a	Figure 11b	Figure 11c	Figure 11d
Image entropy	5.88	5.84	5.50	5.46
Time consumption (s)	\	13.08	82.16	15.82

As can be seen from the above table, the proposed method does not need multiple iterations, avoiding a lot of time overhead. On the premise of achieving nearly the same image entropy, the time used is only 19.25% of the SPGA algorithm. In the case of no iteration, the time consumption is only 20.95% more than SPGA, but the image entropy is increased by 7.6%.

Figure 12, respectively, show the phase error of point A and the comparison of recovery results.

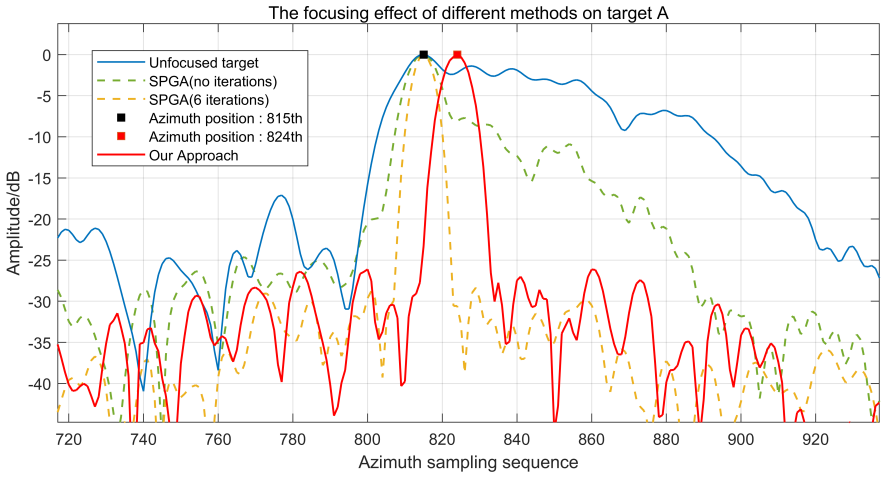

Figure 12. The focusing effect of different methods on target A.

As can be seen from Figure 11, the green dotted line and yellow dotted line are the results of SPGA iterations of different times. With the increase in iterations, the peak sidelobe ratio and integral sidelobe ratio both increase significantly. However, due to the limitations of the algorithm itself, the position of the point target cannot be estimated, and the linear phase component cannot be extracted. Therefore, the position of the peak point hardly changes, and only changes the main lobe width, side lobe peak, and other indicators of the point target. Figure 12 reveals the state of the SPGA algorithm before and after splicing and the difference between the phase error obtained by the algorithm in this paper after splicing. The light blue dashed line is the phase error gradient extracted from point A, and the light purple dashed line 2 is the phase error extracted from point B. After being superimposed and filtered, the two lines are yellow solid lines (the mean value of the overlapped part is taken). As shown in Figure 12, the black dashed ellipse E1 and E2 shows that incoherence occurs when the two points are spliced due to errors in the selection of points A and B. Meanwhile, the red solid line in the figure is the phase error gradient curve of the algorithm splicing in this paper. It can be seen that there is a reservation of linear phase error and no incoherent linearity is generated during splicing.

Figure 13 shows the difference between the phase error gradient curve estimated by SPGA six iterations and the phase error gradient curve estimated in this paper. As a supplement to Figure 13, L1, L2, and L3 in Figure 14 show obvious steep increases at E3 and E4. Meanwhile, it can be seen that their slopes are not zero, indicating that there is a small part of the phase component of quadratic error more than the phase error estimated by SPGA. This is also the reason why the main lobe is slightly widened and the side lobe is slightly increased compared with the sixth iteration of SPGA; however, this does not affect the focusing effect. It can be seen that the proposed algorithm can estimate the azimuth linear phase error component and can accurately splice the phase error to improve the focusing quality.

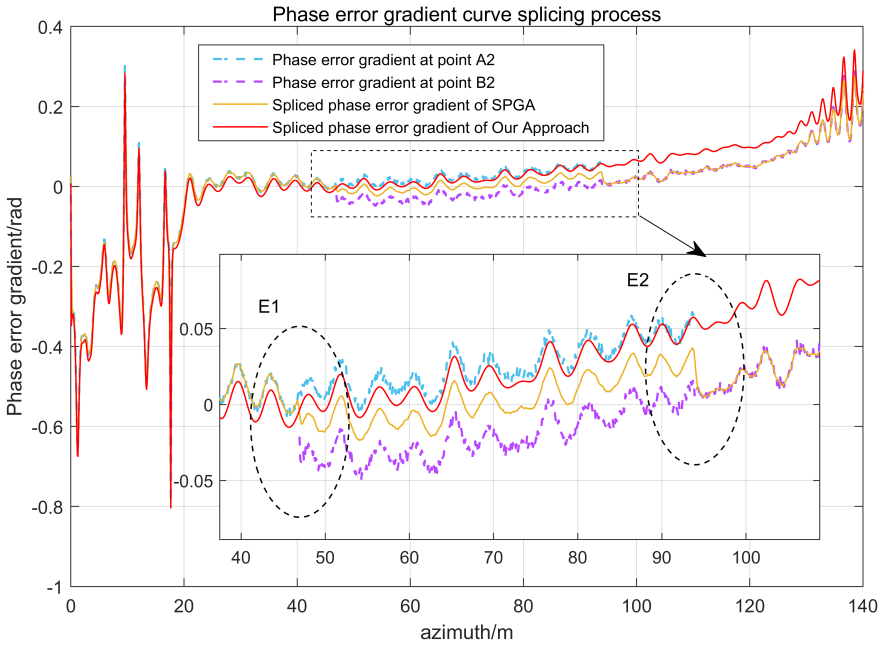

Figure 13. Phase error gradient curve splicing process.

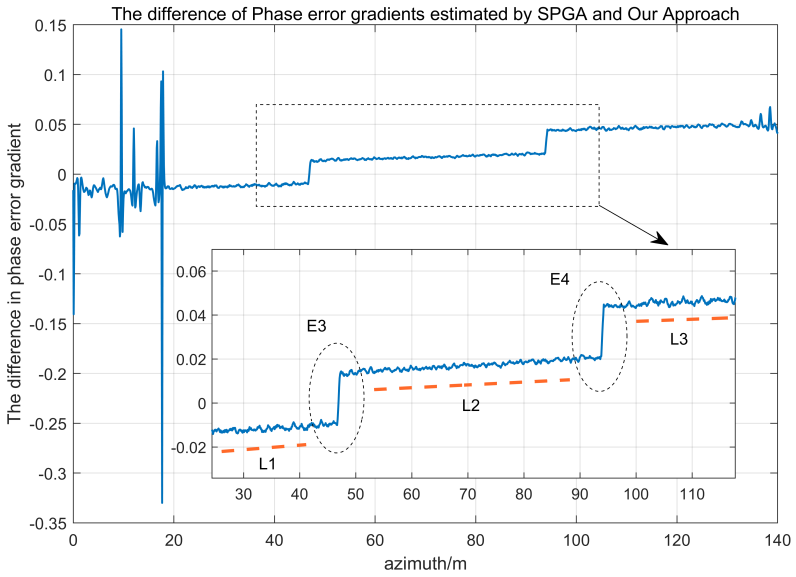

Figure 14. The difference of phase error gradients estimated by SPGA and our approach.

5. Discussion

5.1. Comparison of the Computational Cost

Let the image-sampling points in the azimuth direction be N_a, the sampling points in the range direction be N_r, the size of 2D $CAFR$ protection unit is α, the final number of selected point targets is K, the range of redundant selection points is W (step size is 1), and the number of iterations is N_{int}. Table 6 shows the computational cost for all steps of SPGA and our approach.

The meaning represented by the process number is in Figure 5.

Table 6. Statistical table of computational cost components.

Process	Computational Cost	SPGA	Our Aproach	Simulation Value
1	$\alpha N_r N_a$	✓	✓	2.68×10^8
2	$(2K + N_r)N_a \log_2 N_a + KN_a$	✓	✓	2.02×10^8
3	WKN_a	\	✓	1.15×10^6
4	KN_a	✓	✓	2.87×10^4
5	$K^2 N_a/2$	\	✓	1.00×10^5
6	KN_a	✓	✓	2.87×10^4
7	$N_r N_a$	✓	✓	1.68×10^7
8	$N_r N_a \log_2 N_a$	✓	✓	2.01×10^8

It can be seen that our approach only has two more processes (NO.3 process and NO.5 process) than SPGA in exchange for avoiding algorithm iterations. The last column of Table 6 shows the number of operations consumed by each process under simulation conditions. Although the two algorithms have slight differences in the same step, the difference is not large.

We can see that the computation of SPGA is concentrated in the NO.1 process (Select Point), NO.2 process (Phase Error Extraction), and NO.8 process (Pulse Compression). As long as the computational cost of the extra processes NO.3 and NO.5 is less than the cost

of repeating $N_{int} - 1$ times the above three main processes, our approach can achieve the purpose of improving efficiency, which means it must satisfy the following equation:

$$N_{int}(\alpha N_r + 2N_r \log_2 N_\alpha + 2K \log_2 N_\alpha + 3K + N_r) > WK + K^2/2 \tag{24}$$

We can see that the extra computational cost of our approach is proportional to the square of K. In general, K is much smaller than N_a and N_r, so Equation (24) can be satisfied in most cases. However, in the scenario where some point targets are dense, if too many point targets are selected, there is a possibility that the computational cost will increase. In order to explore the boundary conditions of our approach, that is, the upper limit on the number of selected point targets K, we can draw Figure 15 based on Equation (24) (Suppose that $N_a = N_r$).

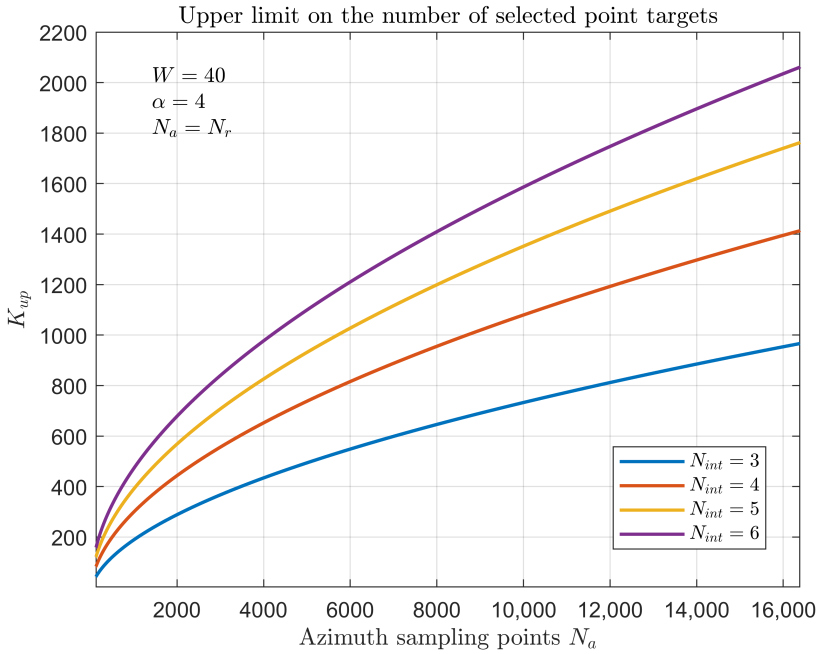

Figure 15. Upper limit on the number of selected point targets.

Figure 15 shows how the upper limit of the number of selected points K_{up} changes with the direction of sampling points N_a. That is, when the azimuth sampling point of the SAR image is N_a, and the sampling point K exceeds the upper limit K_{up}, the computational cost of our approach is equivalent to that of traditional SPGA with N_{int} times of iterations. At the same time, through experiments, it is found that the above curve is not sensitive to W and α in a certain range, so let it be an empirical value. In order to calculate K_{up} quickly, change ">" in Equation (24) to "=" and simplify appropriately, and the following empirical formula is given:

$$K_{up} = \left\lfloor \sqrt{2} N_{int}^{3/4} \sqrt{N_a \log_2 N_a} \right\rfloor \tag{25}$$

If the number of points selected according to Equation (5) exceeds K_{up}, some points with relatively lower quality Q_k can be appropriately filtered to reduce the number of selected points.

5.2. Summary of Experimental Results.

The simulation results in Figure 9 show that our approach can effectively solve the problem that SPGA cannot estimate the linear phase. Figure 9 shows the estimation effect of SPGA and our approach on the piecewise linear phase for the case of multi-point targets. It can be seen that the traditional SPGA algorithm cannot correctly extract the linear phase without iteration or multiple iterations, whereas our approach can extract the correct linear

phase at one time. At the same time, it can be seen from Figures 9 and 10 and Table 3 that in the case of a multi-target experiment, after adding the piecewise linear phase error, the average deviation of the seven point targets is 2.29 pixels, PLSR drops to −4.99 dB on average, and ILSR drops to −2.32 dB on average. After being processed by SPGA without iterations, the average offset of the target was reduced to 1.86 pixels, PLSR further deteriorated to −2.03 dB, and ILSR further deteriorated to −0.87 dB. After six iterations of SPGA, although the ILSR and ILSR indicators are improved to −7.11 dB and −3.99 dB, respectively, the average offset points of point targets are increased to 8.42 pixels. In other words, although the point target focus quality is improved, the point target position is shifted more severely. On the contrary, our approach improves ILSR and ILSR to −12.34 dB and −9.87 dB without iteration, and the average drift points are controlled at −0.16 pixels, which is basically restored to the original position. The above results reflect that our approach is much better than the original SPGA algorithm.

In addition, we processed and compared the actual data, and Figure 11 shows the focusing results of different processing methods. From the comparison of Figure 11a–d, it can be seen that SPGA cannot play a good focusing effect in the case of no iterations, but with the increase in the number of iterations, the focusing effect has a certain improvement. However, this improvement is only obvious in the selected points, and the overall improvement of the image is still not good, as can be seen from the focusing effect of small targets in the image. In contrast, the effect of our approach without iterations is much better than that of the traditional SPGA without iterations. At the same time, the overall focusing effect of the image is better than the effect of SPGA with six iterations (see the image entropy of various processing methods in Table 5 for details). At the same time, as shown in Figure 12, SPGA cannot change the position of the selected point target in the image, indicating that it cannot estimate and compensate for the linear phase. However, the proposed method can estimate the true position of the point target through phase continuity, and prevent the candidate point from being locked in a fixed position, thus limiting SPGA to estimate the phase error more accurately.

Figures 13 and 14 show the reason for the above result: the linear components of the phase error extracted by SPGA are different according to different point target positions, and the jump of the phase gradient will occur at the phase splicing, which will cause the phase gradient discontinuity after splicing. The proposed method, when extracting the phase error from the point target, will estimate the true point target position according to the difference in linear components, and then extract the phase error with continuous phase gradient. However, when extracting the phase error from the point target, the proposed method estimates the true point target position according to the difference in linear components of adjacent point targets and extracts the phase error with continuous phase gradient.

In addition, compared with traditional parametric autofocus algorithms such as MD and SAC, the proposed algorithm inherits the advantages of SPGA, which does not depend on specific model parameters, can estimate high-order phase errors, can directly obtain phase error parameters from images, and the algorithm is easy to converge. Compared with the non-parametric self-focusing algorithm, the proposed algorithm overcomes the disadvantages of SPGA and exchanges the advantages of non-iteration and non-distortion with a small computational cost. The efficiency and effect of the algorithm are greatly improved.

6. Conclusions

According to the continuity of the phase error, this paper proposes a modified iteration-free Stripmap Phase Gradient Algorithm based on removing the linear phase. Through the analysis of the mathematical process of phase error extraction, this paper finds a method to estimate the real position of the point target, which retains linear error in phase error extraction, thus eliminating the error generated in the phase stitching process, making the estimated phase error curve more accurate and smooth, avoiding the frequent iteration of traditional SPGA. At the same time, our approach can keep the global linear phase error

level consistent, making the image scene as a whole without relative offset.This algorithm overcomes the defects of SPGA that make it difficult to remove linear phase errors and requires iteration, resulting in low operational efficiency.Under normal circumstances, our approach without iteration can achieve the focusing effect of SPGA for six iterations. The validity and efficiency of our approach are verified by the processing of simulation data and real data. By analyzing the time cost of the two algorithms, this paper obtains the use boundary of our approach. It provides a solution for SAR real-time self-focusing applications and has a certain significance. It is worth noting that the algorithm is suitable for small squint angle conditions; when the squint angle is too large, the algorithm needs further improvement.

Author Contributions: Conceptualization, Y.X. and X.Z.; formal analysis, Y.X. and X.Z.; funding acquisition, L.C.; methodology, Y.X. and X.Z.; project administration, Y.L.; resources, L.C. and Y.L.; validation, Y.X., Y.L. and X.Z.; software, Y.X. and X.Z.; writing—original draft preparation, Y.X.; writing—review and editing, Y.X., L.C. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key R&D Program of China, 2022YFB3901601.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to restrictions eg privacy or ethical.

Acknowledgments: The authors thank all colleagues who participated in the SAR system design and the acquisition of measured data. Furthermore, the authors would like to express their gratitude to the anonymous reviewers and the editor for their constructive comments on the paper.

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

Abbreviations

The following abbreviations are used in this manuscript:

SAR	Synthetic Aperture Radar
INS	Inertial Navigation System
GNSS	Global Navigation Satellite System
MoCo	Motion Compensation
2D	Two-Dimensional
LFM	Linear frequency modulation
CFAR	Constant False Alarm Rate Detector
AIRCAS	Aerospace Information Research Institute, Chinese Academy of Sciences
POS	Position and Orientation System
ISLR	Integral Sidelobe Ratio
PSLR	Peak Sidelobe Ratio

References

1. Deng, Y.; Yu, W.; Zhang, H.; Wang, W.; Liu, D.; Wang, Y. Forthcoming spaceborne SAR development. *J. Radars* **2020**, *9*, 1–33.
2. Zhang, L.; Qiao, Z.; Xin, M.; Yang, L.; Bao, Z. A robust motion compensation approach for UAV SAR imagery. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 3202–3218. [CrossRef]
3. Chen, Z.; Zhang, Z.; Zhou, Y.; Wang, P.; Qiu, J. A Novel Motion Compensation Scheme for Airborne Very High Resolution SAR. *Remote Sens.* **2021**, *13*, 2729. [CrossRef]
4. Ren, Y.; Tang, S.; Dong, Q.; Sun, G.; Guo, P.; Jiang, C.; Han, J.; Zhang, L. An Improved Spatially Variant MOCO Approach Based on an MDA for High-Resolution UAV SAR Imaging with Large Measurement Errors. *Remote Sens.* **2022**, *14*, 2670. [CrossRef]
5. Ren, Z.; Zhu, D. A GPU-Based Two-Step Approach for High Resolution SAR Imaging. In Proceedings of the 2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP), Nanjing, China, 22–24 October 2021; pp. 376–380.
6. Yi, T.; He, Z.; He, F.; Dong, Z.; Wu, M.; Song, Y. A Compensation Method for Airborne SAR with Varying Accelerated Motion Error. *IEEE Trans. Geosci. Remote Sens.* **2018**, *10*, 1124. [CrossRef]
7. Chen, J.; Xing, M.; Yu, H.; Liang, B.; Peng, J.; Sun, C. Motion Compensation/Autofocus in Airborne Synthetic Aperture Radar: A Review. *IEEE Geosci. Remote Sens. Mag.* **2022**, *10*, 185–206. [CrossRef]
8. Meng, Z.; Zhang, L.; Li, J.; Lu, J.; Li, Y. Time-Domain Azimuth-Variant MOCO Algorithm for Airborne SAR Imaging. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]

9. Wang, W.; An, D.; Luo, Y.; Zhou, Z.; Huang, X. A Modified Map-Drift Algorithm for SAR Autofocusing. In Proceedings of the 2018 Asia-Pacific Microwave Conference (APMC), Kyoto, Japan, 6–9 November 2018; pp. 815–817.
10. Li, B.; Xie, Y.; Liu, X.; Deng, Y. A new reconfigurable methodology to implement the shift-and-correlate (SAC) algorithm for real-time SAR autofocusing. In Proceedings of the IET International Radar Conference, Hangzhou, China, 14–16 October 2015; pp. 1–6.
11. Gao, M.; Qiu, X.; Cheng, Y.; Lv, J.; Ding, C. A Robust Track Error Estimation Method for Airborne SAR Based on Accuracy Analysis Model. *Remote Sens.* **2022**, *14*, 5769. [CrossRef]
12. Klare, J.; Weiss, M.; Peters, O.; Brenner, A.; Ender, J. Two-dimensional autofocus technique for high-resolution spotlight synthetic aperture radar. *IET Signal Process.* **2016**, *10*, 699–707
13. Gao, Y.; Yu, W.; Wang, R. Autofocus algorithm for SAR imagery based on sharpness optimisation. *Electron. Lett.* **2014**, *50*, 830–832. [CrossRef]
14. Schulz, T.J. Optimal sharpness function for SAR autofocus. *IEEE Signal Process. Lett.* **2007**, *14*, 27–30. [CrossRef]
15. Morrison, R.L.; Do, M.N.; Munson, D.C. SAR image autofocus by sharpness optimization: A theoretical study. *IEEE Trans. Image Process.* **2007**, *16*, 2309–2321. [CrossRef] [PubMed]
16. Yang, L.; Xing, M.; Zhang, L.; Sheng, J.; Bao, Z. Entropybased motion error correction for high-resolution spotlight SAR imagery. *IET Radar Sonar Navig.* **2012**, *6*, 627–637. [CrossRef]
17. Zeng, T.; Wang, R.; Li, F. SAR image autofocus utilizing minimum-entropy criterion. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 1552–1556. [CrossRef]
18. Zhang, S.; Liu, Y.; Li, X. Fast entropy minimization based autofocusing technique for ISAR imaging. *IEEE Trans. Signal Process.* **2019**, *63*, 3425–3434. [CrossRef]
19. Eichel, P.H.; Ghiglia, D.C.; Jakowatz, C.V. Speckle processing method for synthetic-aperture-radar phase correction. *Opt. Lett.* **1989**, *14*, 1–3. [CrossRef] [PubMed]
20. Liao, Y.; Shao, H.; Chen, H.; Wang, W. A modefied PGA motion compensation method for circular trace scanning SAR. In Proceedings of the 2016 CIE International Conference on Radar (RADAR), Guangzhou, China, 10–13 October 2016; pp. 1–4.
21. Zhang, R.; Wang, H.; Hu, C.; Chen, X. An Optimization Algorithm of Moving Targets Refocusing Via Parameter Estimation Dependence of Maximum Sharpness Principle After BP Integral. In Proceedings of the Data Processing of Airborne FMCW SAR Sliding Spotlight, 2021 2nd China International SAR Symposium (CISS), Shanghai, China, 3–5 November 2021; Volume 14, pp. 1–4.
22. Meng, D.; Ding, C. A New Approach to Autofocus Considering Strip Map SAR. *J. Electron. Inf. Technol.* **2005**, *27*, 1349–1352.
23. Zhang, X.; Wu, Y.; Ding C. A Easy Implementation Real Time Autofocus Algorithm for High Resolution Airborne SAR. *J. Electron. Inf. Technol.* **2006**, *28*, 923–926.
24. Chen, J.; Liang, B.; Yang, D. Two-Step Accuracy Improvement of Motion Compensation for Airborne SAR With Ultrahigh Resolution and Wide Swath. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 7148–7160. [CrossRef]
25. Xiang, D.; Tang, T.; Zhao, L.; Su, Y. Superpixel Generating Algorithm Based on Pixel Intensity and Location Similarity for SAR Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 1414–1418. [CrossRef]
26. Ye, Y.; Shen, L.; Hao, M.; Wang, J.; Xu, Z. Robust Optical-to-SAR Image Matching Based on Shape Properties. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 564–568. [CrossRef]
27. Zheng, X.; Sun, W.; Li, Q.; Guo, Y. INS/Scen matching intergrated navigation based on evidence reasoning. *J. Chin. Inert. Technol.* **2014**, *3*, 176–182.
28. Wang, J.; Huang, Z.; Yin, J.; Yin, K.; Wang, Z. A Navigation Method of The SAR Platform Combining The INS with The Scene-Matching Points. In Proceedings of the 2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 17–19 June 2022; pp. 640–645.
29. Smith, M.E.; Varshney, P.K. Intelligent CFAR processor based on data variability. *IEEE Trans. Aerosp. Electron. Syst.* **2000**, *3*, 837–847. [CrossRef]
30. Liu, Y.; Zhang, J.; Suo, J.; Zhang, J.; Yao, T. Research on a New Comprehensive CFAR (Comp-CFAR) Processing Method. *IEEE Access* **2019**, *7*, 19401–19413. [CrossRef]
31. Gao, Y.; Yu, W.; Feng, J.; Zheng, S.; Yang, L. A SAR Back Projection Autofocusing Algorithm Based on Legendre Approximation. *J. Radars* **2014**, *3*, 176–182. [CrossRef]