sensors

MDPI

Article

Spiral SAR Imaging with Fast Factorized Back-Projection: A Phase Error Analysis

Juliana A. Góes [1,*], Valquiria Castro [1], Leonardo Sant'Anna Bins [2] and Hugo E. Hernandez-Figueroa [1]

[1] School of Electrical and Computer Engineering, University of Campinas—UNICAMP, Campinas 13083-852, Brazil; v162627@g.unicamp.br (V.C.); hugo@unicamp.br (H.E.H.-F.)
[2] National Institute for Space Research—INPE, São José dos Campos 12227-010, Brazil; leonardo.bins@inpe.br
* Correspondence: jugoes@decom.fee.unicamp.br

Abstract: This paper presents a fast factorized back-projection (FFBP) algorithm that can satisfactorily process real P-band synthetic aperture radar (SAR) data collected from a spiral flight pattern performed by a drone-borne SAR system. Choosing the best setup when processing SAR data with an FFBP algorithm is not so straightforward, so predicting how this choice will affect the quality of the output image is valuable information. This paper provides a statistical phase error analysis to validate the hypothesis that the phase error standard deviation can be predicted by geometric parameters specified at the start of processing. In particular, for a phase error standard deviation of ~12°, the FFBP is up to 21 times faster than the direct back-projection algorithm for 3D images and up to 13 times faster for 2D images.

Keywords: synthetic aperture radar (SAR); fast factorized back-projection (FFBP); 3D imaging; drone

check for updates

Citation: Góes, J.A.; Castro, V.; Sant'Anna Bins, L.; Hernandez-Figueroa, H.E. Spiral SAR Imaging with Fast Factorized Back-Projection: A Phase Error Analysis. *Sensors* **2021**, *21*, 5099. https://doi.org/10.3390/s21155099

Academic Editors: Antonio Iodice and Gerardo Di Martino

Received: 30 May 2021
Accepted: 23 July 2021
Published: 28 July 2021

1. Introduction

In synthetic aperture radar (SAR) imaging, circular flight path surveys produce 2D images with very high resolution as data are collected over 360° around the imaged area. Circular SAR can also provide 3D scattering information, but the 3D images are deformed by strong cone-shaped sidelobes [1–3]. Multicircular SAR, or holographic SAR tomography (HoloSAR), creates another synthetic aperture in elevation that mitigates these undesirable sidelobes, thus providing complete 3D data reconstruction with very high resolution [4–9]. HoloSAR geometry acquisition consists of multiple circular flight paths at different fixed heights. The sparse nature of the elevation aperture in HoloSAR poses some difficulties for a system working in the THz band [10]. These issues are overcome with a cylindrical spiral flight pattern with constant vertical speed.

SAR image processing requires efficient algorithms in terms of both accuracy and processing time. Frequency-domain algorithms are fast, but they perform better when the flight path is linear and free of motion errors. The time-domain back-projection (BP) algorithm can process SAR data for any flight path with high focusing quality but with high computational costs. Fast factorized back-projection (FFBP) algorithms can significantly reduce the computational time while still maintaining the accuracy of the BP algorithm. However, the increase in the level of sophistication makes it difficult to formulate an FFBP algorithm for arbitrary trajectories. As a result, many FFBP algorithms either assume a linear flight path to simplify calculations [11–16] or are tailored for circular flight paths [2,17–19].

In [20], the authors proposed an FFBP algorithm that describes subapertures through a data mapping approach that does not depend on the flight path geometry, even though the algorithm assumes that the radar constantly illuminates the imaged area or volume. Moreover, the algorithm operates in cartesian coordinates and employs a flexible tree structure that can handle both 2D and 3D data.

For the HoloSAR presented by Ponce et al. [4], different image layers were processed with a 2D FFBP that is customized for circular trajectories [2]. Ponce et al. did not pursue 3D focusing with their FFBP due to practical reasons [4]. Other HoloSAR solutions have used the direct BP algorithm [6,8], sparse reconstruction models [4,5,7], adaptive imaging [6,9], or a combination of them. Apart from the choice in the algorithm, there are two common approaches:

- Process each circular flight independently and merge the outputs [4,6];
- Make radial slices of the cylindrical synthetic aperture, process them separately, and then combine the results [5,7–9].

For the spiral SAR presented in [10], the whole trajectory was processed with the direct BP algorithm. In [20], the initial version of our FFBP algorithm successfully processed simulated SAR data of a spiral trajectory. To the best of our knowledge, it was the first full 3D FFBP algorithm capable of processing nonlinear SAR data.

Although the preliminary version of the FFBP algorithm [20] is fully functional, it has proven inefficient when operating with real SAR data, both in processing time and memory consumption. Therefore, this paper presents a more consolidated version of the FFBP algorithm [21] that employs vectorized variables and parallel processing to mitigate these issues. Vectorization is essential for increasing efficiency, while parallel computing further decreases processing time and reduces memory consumption.

Processing SAR data with an FFBP algorithm is not as straightforward as with a BP algorithm because some FFBP input parameters can affect the quality of the output image. Thankfully, Ulander et al. [12] provided an error analysis that yielded a method to limit the phase error by controlling the processing setup.

This paper proposes a statistical phase error analysis inspired by [12] but with a key difference. Because the FFBP algorithm presented here works well with curved flight paths, the proposed analysis does not consider that deviations from a linear flight path will deteriorate the phase error.

The purpose was to test the hypothesis that geometric parameters at the beginning of processing can predict the phase error standard deviation of the output image. The data set for testing this hypothesis comprised processing results for a spiral flight path performed by a multiband drone-borne SAR system [22,23]. The collected P-band SAR data were processed with the BP and FFBP algorithms to produce 2D and 3D images. Different parameters were chosen for the FFBP to alter the response in phase error and processing time.

The other sections of the paper are structured as follows. Section 2 presents the FFBP algorithm, the phase error hypothesis, and the case study. Section 3 evaluates several 2D and 3D SAR images regarding the phase error versus the signal-to-noise ratio (SNR), geometric parameters, and processing time. Finally, discussion of the results is presented in Section 4 and the conclusion in Section 5.

2. Materials and Methods

2.1. Fast Factorized Back-Projection Algorithm

The BP algorithm integrates the information from all SAR positions for each pixel in the image in one go. If there are N SAR positions and the output image has N^2 pixels, the number of operations is $\mathcal{O}\left(N^3\right)$. Fortunately, FFBP can reduce computational cost to $\mathcal{O}\left(N^2 \log N\right)$ using a divide-and-conquer strategy, which is at the core of many FFBP algorithms. Before processing starts, each SAR pulse covers a large area. At each iteration, subapertures are merged as if building increasingly larger antenna arrays with more focused beam patterns to cover progressively smaller subimages. Figure 1b shows the steps in this iterative process.

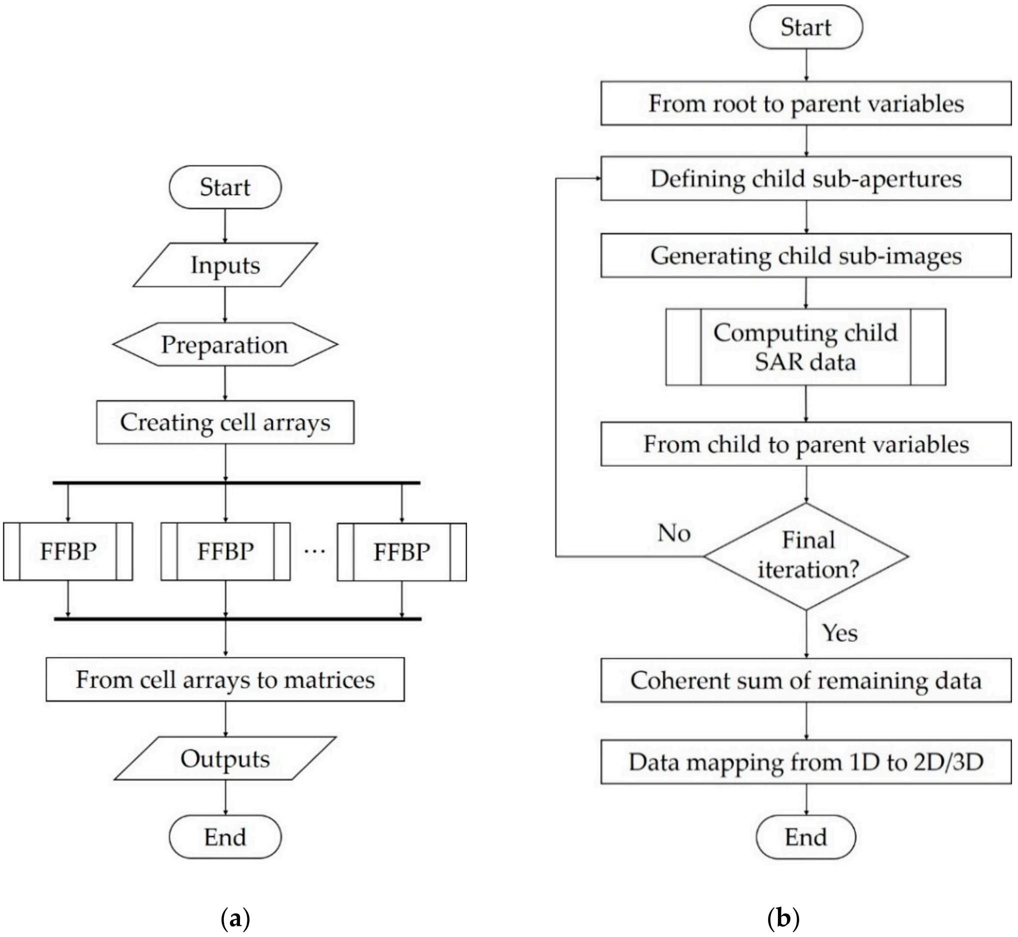

Start
Inputs
Preparation
Creating cell arrays
FFBP
FFBP
···
FFBP
From cell arrays to matrices
Outputs
End
(a)
Start
From root to parent variables
Defining child sub-apertures
Generating child sub-images
Computing child SAR data
From child to parent variables
No
Final iteration?
Yes
Coherent sum of remaining data
Data mapping from 1D to 2D/3D
End
(b)

Figure 1. Flowchart of the algorithm: (**a**) parallel processing and (**b**) fast factorized back-projection (FFBP), an iterative process.

The FFBP algorithm is parallelizable, which means that the computation can be distributed among different processing units that work simultaneously. This is accomplished by dividing the imaged volume into blocks to be processed independently of one another. The data are managed by creating a cell array for each output matrix, i.e., processed SAR data and voxel coordinates. All cell arrays have the same number of elements, and each cell index is associated with an image block. When an image block is processed, its results are stored in the corresponding cells. Then, after processing all image blocks, each cell array is converted into a matrix that combines data for the whole output image. This process is illustrated in Figure 1a.

The next sections use the following terminology (see Figure 1):

- Root variables: either inputs to the algorithm or defined in the preparation step;
- Child variables: calculated within each FFBP iteration and then become parent variables at the end of the iteration;
- Parent variables: inputs to the current iteration.

The proposed algorithm is also vectorized, so matrix indices are written within parentheses to distinguish them from other types of indices. In addition, variables representing positions in the (x, y, z) space are written in bold letters.

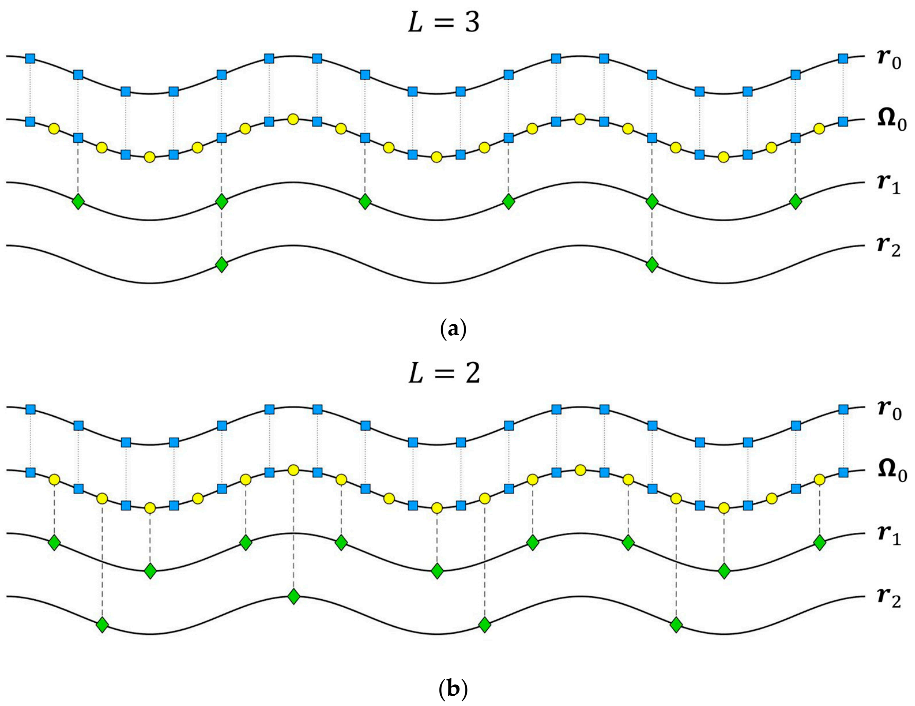
2.1.1. Defining Child Subapertures

The method for defining child subapertures was first proposed in [20]. It takes a data mapping approach and does not depend on the flight pattern. Let $\boldsymbol{r}_0$ be the set of radar positions at the root node, let L be number of parent subapertures that are combined to form a child subaperture at each iteration, and let $\boldsymbol{r}_n$ be the set of the phase centers of all child subapertures at the n^{th} node.

Case 1. *When L is odd, $\boldsymbol{r}_n$ is always a subset of $\boldsymbol{r}_0$.*

Case 2. *When L is even, each point in $\boldsymbol{r}_n$ falls halfway between two consecutive points in $\boldsymbol{r}_0$.*

Cases 1 and 2 are depicted in Figure 2a,b, respectively. Blue squares represent the actual radar root positions, yellow circles represent the midpoints between them, and green diamonds represent the subaperture phase centers.



(**a**)

(**b**)

Figure 2. Definition of child subapertures for (**a**) $L = 3$ and (**b**) $L = 2$. The blue squares, yellow circles, and green diamonds represent the radar root positions, the midpoints between them, and the child subapertures phase centers, respectively. Reprinted with permission from ref. [20]. Copyright 2020 IEEE.

Now, let $\boldsymbol{\Omega}_0$ be defined as

$$\boldsymbol{\Omega}_0(i) = \boldsymbol{r}_0\left(\frac{i}{2}\right), \tag{1}$$

where $i = 0,\ 1,\ \ldots,\ 2(K_0 - 1)$, with K_0 being the number of radar root positions. Then, $\boldsymbol{\Omega}_0$ is the union between $\boldsymbol{r}_0$ and the set of midpoints between two consecutive radar root positions (see Figure 2).

General Case. *For any value of L, $\boldsymbol{r}_n$ is always a subset of $\boldsymbol{\Omega}_0$.*

For the general case, $\boldsymbol{r}_n$ is determined by [20]

$$\boldsymbol{r}_n(k) = \boldsymbol{\Omega}_0((2k+1)L^n - 1), \tag{2}$$

where $k = 0,\ 1,\ \ldots,\ K_n - 1$, with $K_n = K_0/L^n$ being the number of child subapertures at the n^{th} node. Note that for all k, if L is odd, then the argument on the right will always be even, and vice versa.

2.1.2. Generating Child Subimages

Child subimages are generated using a flexible space-filling tree structure called the modified Morton curve [20]. It arranges multidimensional data into 1D following a Z pattern, much like the original Morton order curve [24,25]. The modification, however, allows for different partition schemes beyond dividing by two in each direction in every recursion.

The partition scheme for all iterations is defined in the preparation step (Figure 1a). It consists of a matrix whose columns contain the number of partitions in the x, y, and z dimensions (D_x, D_y, and D_z); the number of lines equals the number of iterations. These quantities are obtained from the output image dimensions and resolution, the initial subdivision into image blocks, and the number of combining subapertures L. Figure 3 shows the modified Morton order curve with a (3 × 3 × 2) partition on the first and second recursions. When working with 2D data, i.e., images with zero thickness, the partition scheme sets $D_z = 1$ for all iterations.

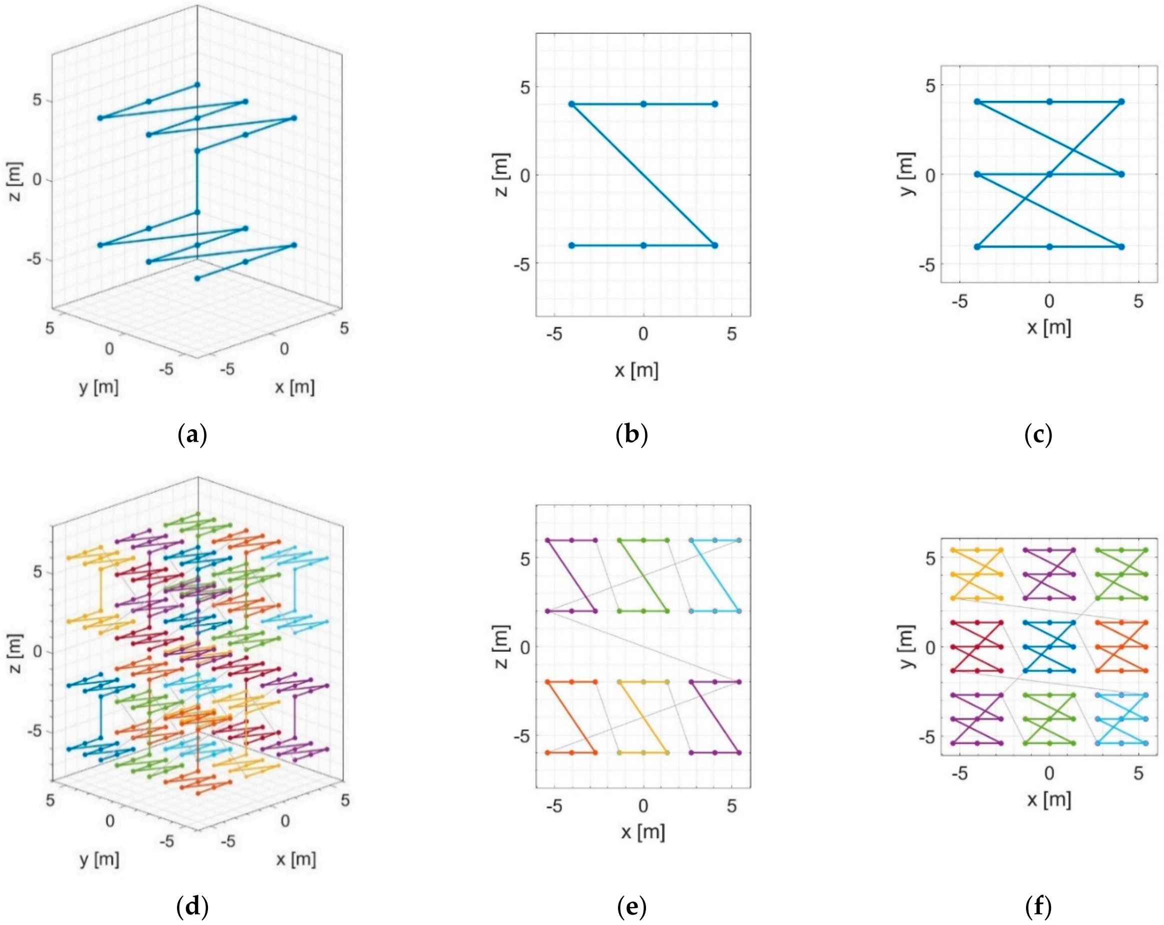


Figure 3. The modified Morton order curve with a (3 × 3 × 2) partition: (**a**) perspective, (**b**) front, and (**c**) top views for the first recursion; (**d**) perspective, (**e**) front, and (**f**) top views for the second recursion. Reprinted with permission from ref. [20]. Copyright 2020 IEEE.

After retrieving the partition scheme for the current iteration, the algorithm finds all possible values of x, y, and z coordinates for the center of the child subimages in a local coordinate system with the parent subimage center $\boldsymbol{h}_{n-1}(p)$ at the origin. Next, the possible values of x, y, and z are arranged in a pattern similar to a truth table in digital systems theory to construct a Z-shaped curve of coordinates $\tilde{x}$, $\tilde{y}$, and $\tilde{z}$ (see Table 1). Then, the position of each child subimage center $\boldsymbol{h}_n(c)$ is given by

$$\boldsymbol{h}_n(c) = \begin{bmatrix} \tilde{x}(d) & \tilde{y}(d) & \tilde{z}(d) \end{bmatrix} + \boldsymbol{h}_{n-1}(p), \tag{3}$$

$$c = pD_n + d, \tag{4}$$

where $d = 0,\ 1,\ \ldots,\ D_n - 1$, $D_n = D_x D_y D_z$ is the number of children generated by each parent, p refers to a parent subimage, and c indicates a child subimage.

Table 1. Order of arrangement of the x, y, and z coordinates of the child subimage centers in a modified Morton order curve with a (3 × 3 × 2) partition.

d	$\tilde{x}(d)$	$\tilde{y}(d)$	$\tilde{z}(d)$	d	$\tilde{x}(d)$	$\tilde{y}(d)$	$\tilde{z}(d)$
0	$x(0)$	$y(0)$	$z(0)$	9	$x(0)$	$y(0)$	$z(1)$
1	$x(1)$	$y(0)$	$z(0)$	10	$x(1)$	$y(0)$	$z(1)$
2	$x(2)$	$y(0)$	$z(0)$	11	$x(2)$	$y(0)$	$z(1)$
3	$x(0)$	$y(1)$	$z(0)$	12	$x(0)$	$y(1)$	$z(1)$
4	$x(1)$	$y(1)$	$z(0)$	13	$x(1)$	$y(1)$	$z(1)$
5	$x(2)$	$y(1)$	$z(0)$	14	$x(2)$	$y(1)$	$z(1)$
6	$x(0)$	$y(2)$	$z(0)$	15	$x(0)$	$y(2)$	$z(1)$
7	$x(1)$	$y(2)$	$z(0)$	16	$x(1)$	$y(2)$	$z(1)$
8	$x(2)$	$y(2)$	$z(0)$	17	$x(2)$	$y(2)$	$z(1)$

The positions $\boldsymbol{h}_{n-1}(p)$ and $\boldsymbol{h}_n(c)$ do not contain information about the terrain topography. Thus, the terrain height H_{DEM} needs to be interpolated from a digital elevation model (DEM). Finally, the actual position of the child subimage $\widetilde{\boldsymbol{h}}_{n,c}$ is

$$\widetilde{\boldsymbol{h}}_{n,c} = \boldsymbol{h}_n(c) + [0\;0\;H_{DEM}(\boldsymbol{h}_n(c))]. \tag{5}$$

To convert the serial index c into subscripts of a 3D matrix (u, v, w), recurrent sequences are necessary. These sequences are also built in a parent–child dynamic to allow for flexible partition schemes. Let q_{xn}, q_{yn}, and q_{zn} be the recurrent sequences of the n^{th} iteration, then

$$q_{x0}(0) = q_{y0}(0) = q_{z0}(0) = 0, \tag{6a}$$

$$q_{xn}(uD_x + d_x) = D_n q_{xn-1}(u) + d_x, \tag{6b}$$

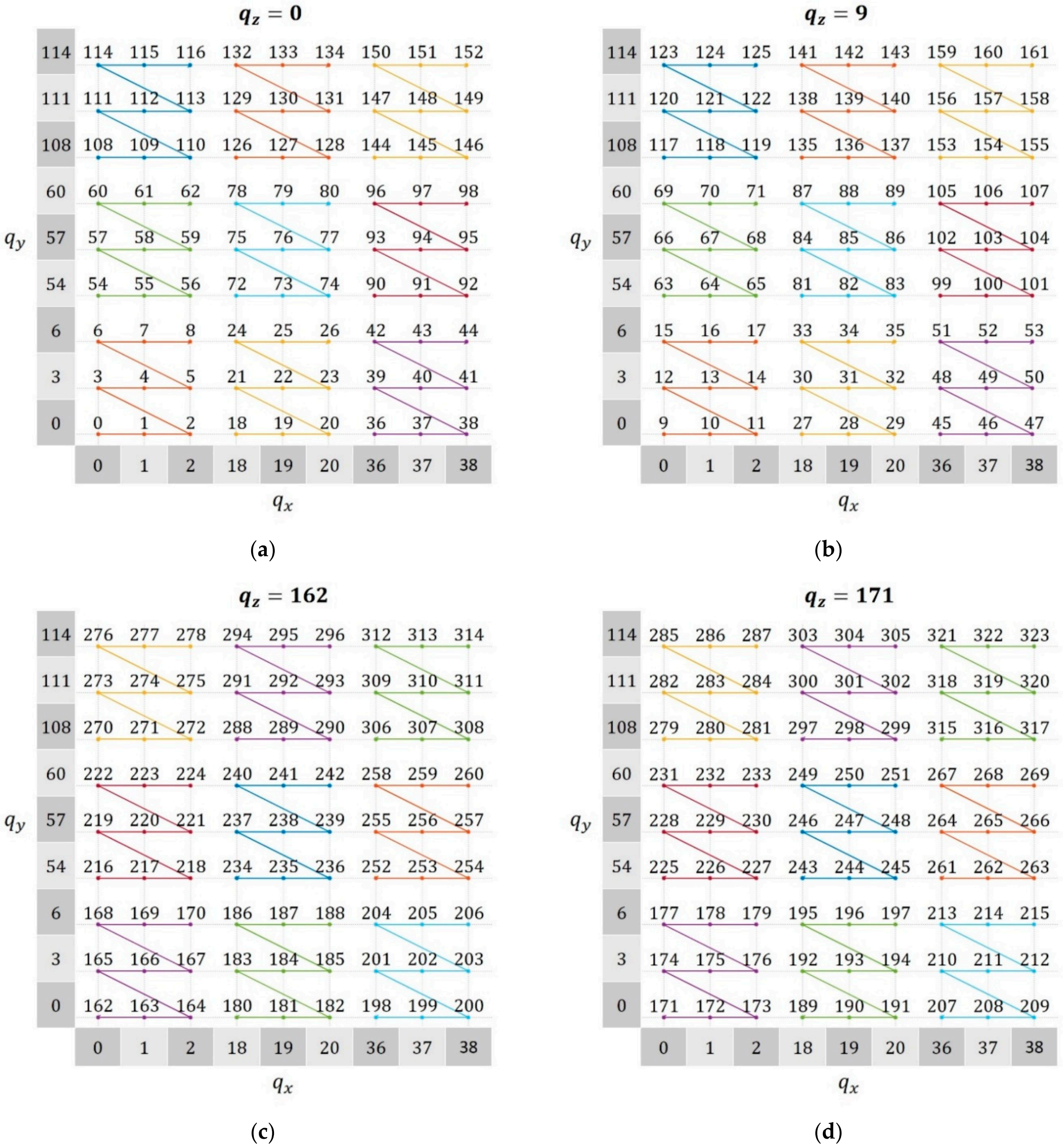
$$q_{yn}(vD_y + d_y) = D_n q_{yn-1}(v) + d_y D_x, \tag{6c}$$

$$q_{zn}(wD_z + d_z) = D_n q_{zn-1}(w) + d_z D_x D_y, \tag{6d}$$

where $d_x = 0, 1, \ldots, D_x - 1$, and the same for d_y and d_z. Therefore, the mapping $c \to (u, v, w)$ from the modified Morton order curve into a 3D matrix can be carried out with the following relationship:

$$c = q_{xn}(u) + q_{yn}(v) + q_{zn}(v). \tag{7}$$

Figure 5 demonstrates how Equations (6) and (7) correspond to the curve shown in Figure 3d–f. The sequences q_x and q_y are indicated on the axes, and each panel corresponds to a different element of q_z. The child subimage index starts with $c = 0$ at the bottom left corner of Figure 4a, then moves back and forth between the layers $q_z = 0$ and $q_z = 9$ until reaching $c = 161$ at the top right corner of Figure 4b. Then, it continues at the bottom left corner of Figure 4c, going back and forth between $q_z = 162$ and $q_z = 171$ up until the end, at the top right corner of Figure 4d.



$q_z = 0$

114	114	115	116	132	133	134	150	151	152
111	111	112	113	129	130	131	147	148	149
108	108	109	110	126	127	128	144	145	146
60	60	61	62	78	79	80	96	97	98
57	57	58	59	75	76	77	93	94	95
54	54	55	56	72	73	74	90	91	92
6	6	7	8	24	25	26	42	43	44
3	3	4	5	21	22	23	39	40	41
0	0	1	2	18	19	20	36	37	38
	0	1	2	18	19	20	36	37	38

q_y

q_x

(**a**)

$q_z = 9$

114	123	124	125	141	142	143	159	160	161
111	120	121	122	138	139	140	156	157	158
108	117	118	119	135	136	137	153	154	155
60	69	70	71	87	88	89	105	106	107
57	66	67	68	84	85	86	102	103	104
54	63	64	65	81	82	83	99	100	101
6	15	16	17	33	34	35	51	52	53
3	12	13	14	30	31	32	48	49	50
0	9	10	11	27	28	29	45	46	47
	0	1	2	18	19	20	36	37	38

q_y

q_x

(**b**)

$q_z = 162$

114	276	277	278	294	295	296	312	313	314
111	273	274	275	291	292	293	309	310	311
108	270	271	272	288	289	290	306	307	308
60	222	223	224	240	241	242	258	259	260
57	219	220	221	237	238	239	255	256	257
54	216	217	218	234	235	236	252	253	254
6	168	169	170	186	187	188	204	205	206
3	165	166	167	183	184	185	201	202	203
0	162	163	164	180	181	182	198	199	200
	0	1	2	18	19	20	36	37	38

q_y

q_x

(**c**)

$q_z = 171$

114	285	286	287	303	304	305	321	322	323
111	282	283	284	300	301	302	318	319	320
108	279	280	281	297	298	299	315	316	317
60	231	232	233	249	250	251	267	268	269
57	228	229	230	246	247	248	264	265	266
54	225	226	227	243	244	245	261	262	263
6	177	178	179	195	196	197	213	214	215
3	174	175	176	192	193	194	210	211	212
0	171	172	173	189	190	191	207	208	209
	0	1	2	18	19	20	36	37	38

q_y

q_x

(**d**)

Figure 4. Child subimage indices c at their corresponding locations over the modified Morton order curve with the recurrent sequences and q_y displayed on the axes and with (**a**) $q_z = 0$, (**b**) $q_z = 9$, (**c**) $q_z = 162$, and (**d**) $q_z = 171$.

2.1.3. Computing Child SAR Data

The child SAR data are both an output of the current iteration and an input for the next. For this reason, multiple range samples are required until the second to last iteration. Additionally, the child SAR data are a function of two slant range distances instead of one. Except for these differences, computing child SAR data is the step that most resembles the direct BP algorithm. Its process is illustrated in Figure 5.

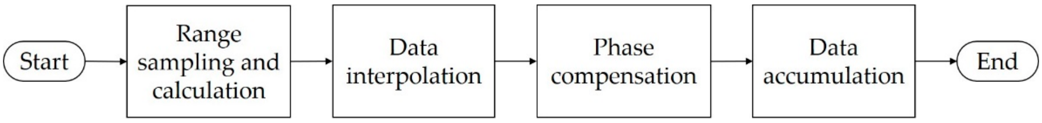


Figure 5. Flowchart of the process for computing child synthetic aperture radar (SAR) data.

Range samples are collected along a line defined by the center of the child subaperture $\boldsymbol{r}_n(k)$ and the center of the child subimage $\widetilde{\boldsymbol{h}}_{n,c}$. A sample is always taken at $\widetilde{\boldsymbol{h}}_{n,c}$; except for the last iteration, other samples are taken along the diameter of the sphere that circumscribes the child subimage, as depicted in Figure 6. The range sampling interval is the same for all iterations. It is calculated in the preparation step, shown in Figure 1a, and is equal to the resulting range bin spacing after upsampling the root SAR data.

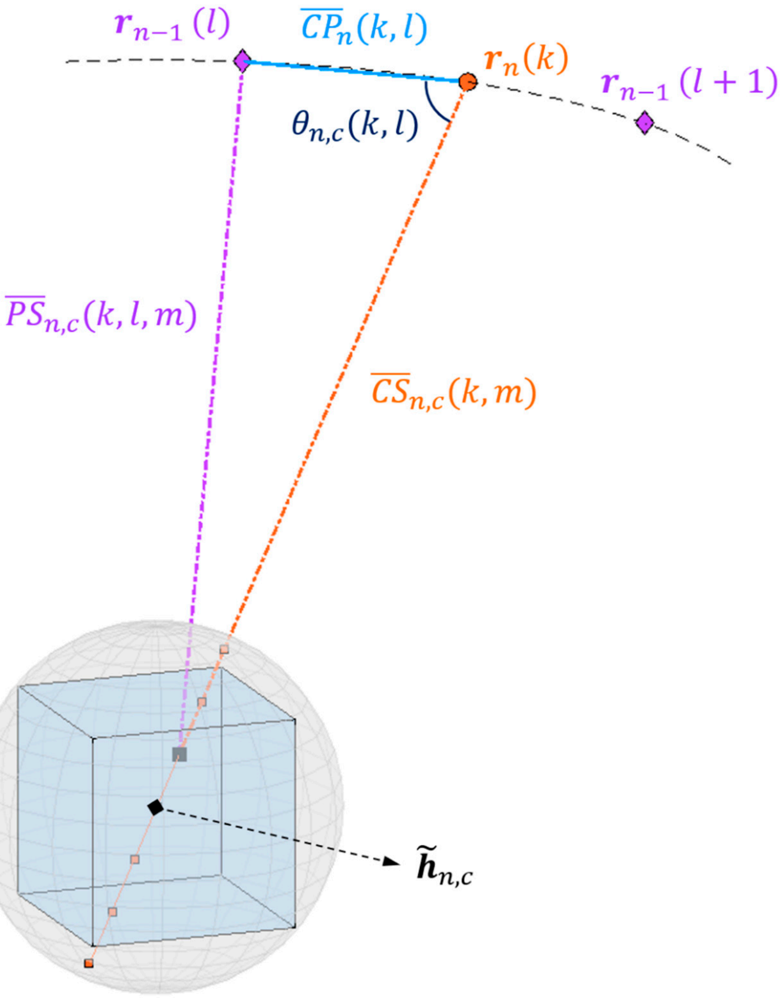

$\boldsymbol{r}_{n-1}(l)$
$\overline{CP}_n(k,l)$
$\boldsymbol{r}_n(k)$
$\boldsymbol{r}_{n-1}(l+1)$
$\theta_{n,c}(k,l)$
$\overline{PS}_{n,c}(k,l,m)$
$\overline{CS}_{n,c}(k,m)$
$\widetilde{\boldsymbol{h}}_{n,c}$

Figure 6. Range samples at a child subimage and geometry for calculating distances between a range sample, a child subaperture, and a parent subaperture.

Figure 6 also shows how the required slant range distances are obtained. The depicted triangle is composed of the following vertices:

- (C) the child subaperture center $\boldsymbol{r}_n(k)$;
- (P) the parent subaperture center $\boldsymbol{r}_{n-1}(l)$;
- (S) the m^{th} data sample within a child subimage centered at $\widetilde{\boldsymbol{h}}_{n,c}$.

The sides $\overline{CP}_n(k,l)$ and $\overline{CS}_{n,c}(k,m)$, as well as the angle $\theta_{n,c}(k,l)$ between them, are found by analytic geometry. Then, the side $\overline{PS}_{n,c}(k,l,m)$ is calculated with the law of cosines.

The child datum $s_n(k,m,c)$ is computed by the coherent sum of the parent data [11]:

$$s_n(k,m,c) = \sum_{l \in \Lambda_{n,k}} s_{n-1}(l, \nu_{n,c}(k,l,m), p) e^{j\varphi_{n,c}(k,l,m)}, \tag{8}$$

where $l \in \Lambda_{n,k} = \{kL + b | b = 0,\ 1,\ \ldots,\ L-1\}$ is the set of parent subapertures associated with the k^{th} child subaperture. The fractional index $\nu_{n,c}(k,m,l)$ is given by [11]

$$\nu_{n,c}(k,l,m) = \frac{\overline{PS}_{n,c}(k,l,m) - \overline{CS}_{n-1,p}(l,0)}{\alpha}, \tag{9}$$

where α is the range sampling interval and $\overline{CS}_{n-1,p}(l,0)$ is the slant range from the parent subaperture to the first sample in the parent data. The value $s_{n-1}(l, \nu_{n,c}(k,l,m), p)$ is determined via linear interpolation, and the phase compensation term $\varphi_{n,c}(k,l,m)$ in (8) is given by [13]

$$\varphi_{n,c}(k,l,m) = \frac{4\pi}{\lambda_0}\left[\overline{PS}_{n,c}(k,l,m) - \overline{CS}_{n,c}(k,m)\right], \tag{10}$$

where λ_0 is the radar frequency.

Each of the indices k, m, and l correspond to a different matrix dimension. Note that none of the variables denoting position (indicated in bold letters) are dependent on the data sample index m, so there is no need for a fourth matrix dimension to account for the (x, y, z) triplets.

After reaching the final iteration, the remaining subapertures are coherently combined. Finally, the resulting serial data are mapped into a 2D or 3D data matrix using (6) and (7).

2.2. The Phase Error Hypothesis

According to Ulander et al. [12], the phase error is proportional to the range error averaged over all subapertures and iterations. The range error, in turn, is introduced by the FFBP algorithm and can be estimated for each iteration. For a linear flight path, the estimated range error is proportional to the child subaperture length and the child subimage width and inversely proportional to the distance between those two entities. For nonlinear flight paths, the across-track deviation has to be taken into account. Based on this analysis, they proposed a method for keeping the phase error below a given threshold:

- Calculate the maximum subimage size for the first iteration;
- Balance the increase in subaperture length with an equivalent decrease in subimage width to keep the range error constant.

This paper investigates if the phase error standard deviation $\sigma_{\Delta\varphi}$ can be predicted by the geometric parameters at the first iteration. Moreover, instead of the subimage width, the subimage diagonal is considered as it is more relevant to the FFBP algorithm detailed in the preceding sections. Specifically, the goal is to test the following hypothesis:

$$\sigma_{\Delta\varphi} \propto \beta = \frac{4\pi}{\lambda_0} \cdot \frac{\delta_k \Delta_h}{R_{min}}, \tag{11}$$

where δ_k and Δ_h are the child subaperture length and the child subimage diagonal at the first FFBP iteration, respectively, and R_{min} is the shortest distance from the radar to the imaged volume.

In [12], the across-track deviation is inserted into the estimated range error equation to account for phase error degradation in nonlinear flight paths. However, the FFBP algorithm proposed in this paper does not suffer from such degradation thanks to the phase compensation term (10), as noted in [20]. That is why hypothesis (11) does not take into consideration any deviations from a linear flight path.

2.3. The Case Study

The case study comprised SAR data from a drone-borne SAR system [22,23] that flew over a eucalyptus plantation with a spiral flight pattern. Figure 7 displays a Google Earth image of the drone trajectory over the imaged area; the eucalyptus plantation can be seen on the bottom left. The spacing between the trees was around 3 m. The survey took place on 13 November 2019, in Mogi Guaçu, São Paulo, Brazil. The drone-borne SAR system works with three different frequency bands, but only the results for the P-band are presented here. Table 2 shows the radar acquisition parameters.

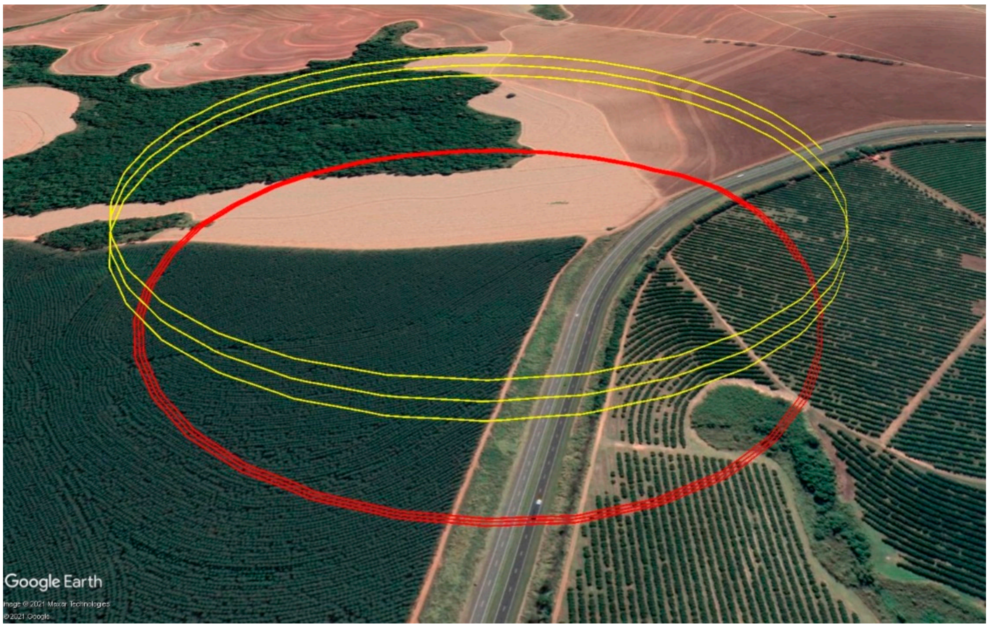


Figure 7. Google Earth image of the spiral flight path over the imaged area.

Table 2. Radar acquisition parameters.

Radar Parameters	Values	Units
Carrier wavelength	70.54	cm
Bandwidth	50	MHz
Range resolution	2.4	m
Pulse repetition frequency	64.95	Hz
Mean velocity	8.5	m/s
Mean flight radius	338	m
Height above ground level	79–120	m
Number of turns	3	-

The purpose of this case study was to investigate the hypothesis (11) by varying δ_k, Δ_h, and R_{min} for the first iteration. This was accomplished by the following steps:

- Setting different values for the number of subapertures that are combined at each iteration (L);
- Choosing different schemes for the initial partition into image blocks;
- Selecting two image blocks for analysis, one close to the edge and one close to the center of the output image (see Figure 8).

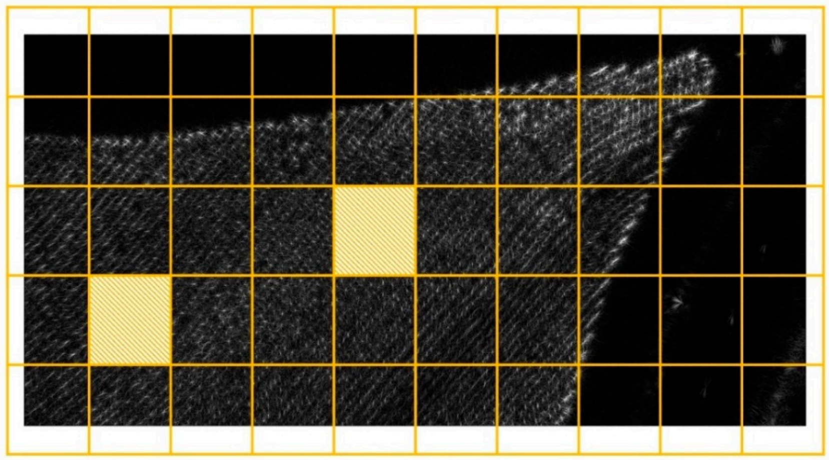


Figure 8. An illustration of the selected image blocks for analyzing the phase error as a function of geometric parameters.

Table 3 shows the selected set of input parameters.

Table 3. Set of processing parameters for the case study.

Processing Parameter		**Values**
Output image dimension	2D	$300 \times 150\ \mathrm{m}^2$
	3D	$300 \times 150 \times 2.4\ \mathrm{m}^3$
Output image resolution	2D	$0.2 \times 0.2\ \mathrm{m}^2$
	3D	$0.2 \times 0.2 \times 0.2\ \mathrm{m}^3$
Number of subapertures combined at each iteration (L)		2
		3
		4
		5
Initial partition of image blocks		$8 \times 4 \times 1$
		$12 \times 6 \times 1$
		$16 \times 8 \times 1$
		$20 \times 10 \times 1$
		$24 \times 12 \times 1$

For each setup, the partition scheme was the same for all image blocks. Because of that, the resulting number of pixels or voxels might not match the expected value calculated from the output image dimension and resolution. There were two options: either process the image with a different resolution or let the output image size be distinct from what is required. Because the FFBP images needed to be compared with the BP to carry out the analysis, the second option was adopted. Moreover, to minimize the waste of computing undesired pixels or voxels, the actual number of image blocks might be larger than the one provided as an input. Ultimately, this would result in a wider variation for Δ_h and R_{min}. A function executed this process at the preparation step (Figure 1a). The outcome is figuratively represented in Figure 8.

Both BP and FFBP algorithms were written in MATLAB R2018a with vectorized variables and parallel computing functions. All data were processed on an Intel(R) Core (TM) i7-7700 CPU (3.60 GHz) with 64 GB RAM.

3. Results

3.1. FFBP vs. BP

Figures 9 and 10 present the 3D output images processed by the direct BP algorithm and the FFBP algorithm, respectively. They depict isosurfaces at −15 dB normalized magnitude, clearly showing that the radar detects every single eucalyptus tree. The processing setup for Figure 10 uses $L = 5$ and an $(8 \times 4 \times 1)$ initial partition. Although this setup produced the highest phase error of the case study, a qualitative comparison suggested that the differences between the two images were quite subtle. Indeed, the degree of coherence between them was 0.9916; the magnitude error had a −0.3 dB mean and a 2.5 dB standard deviation; the mean phase error was 0.0007 rad (0.04°); and the phase error standard deviation was 0.35 rad (19.9°), slightly below $\pi/8$ rad.

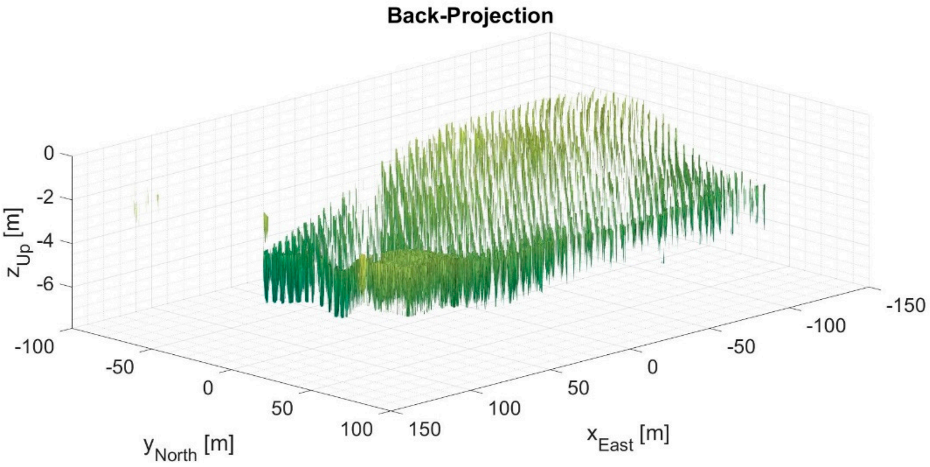


Figure 9. 3D output image processed by the back-projection (BP) algorithm. Perspective view of isosurfaces at −15 dB normalized magnitude.

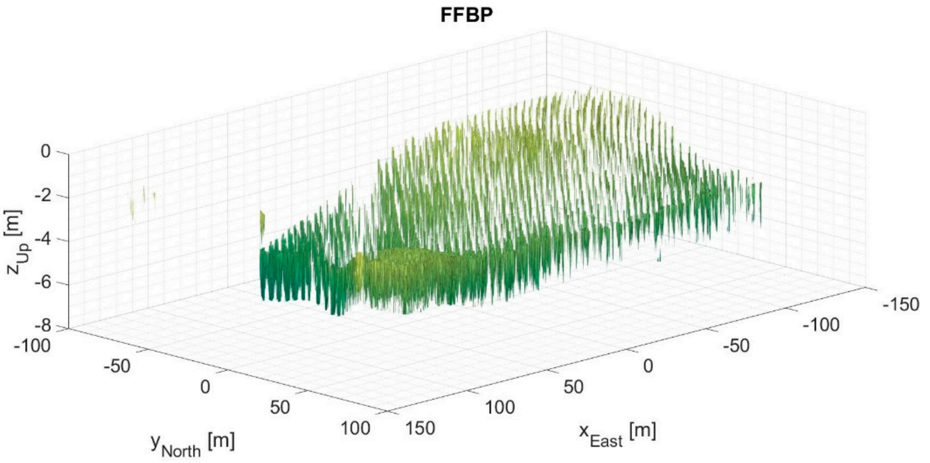


Figure 10. 3D output image processed by the FFBP algorithm with $L = 5$ and an $(8 \times 4 \times 1)$ initial partition. Perspective view of isosurfaces at −15 dB normalized magnitude.

Figure 11 presents the 2D output image processed by the direct BP algorithm, and Figure 12 shows the 2D output image processed by the FFBP algorithm using $L = 5$ and an $(8 \times 4 \times 1)$ initial partition. Again, this setup produced the highest phase error of the case study. However, as in the previous case, the differences between the two images were barely perceptible. The degree of coherence between them was 0.9942; the magnitude error had a −0.2 dB mean and a 2.3 dB standard deviation; the mean phase error was 0.0004 rad (0.02°); and the phase error standard deviation was 0.33 rad (18.8°), also somewhat below $\pi/8$ rad. Lastly, the lines of trees of the eucalyptus plantation can be easily seen in both Figures 11 and 12.

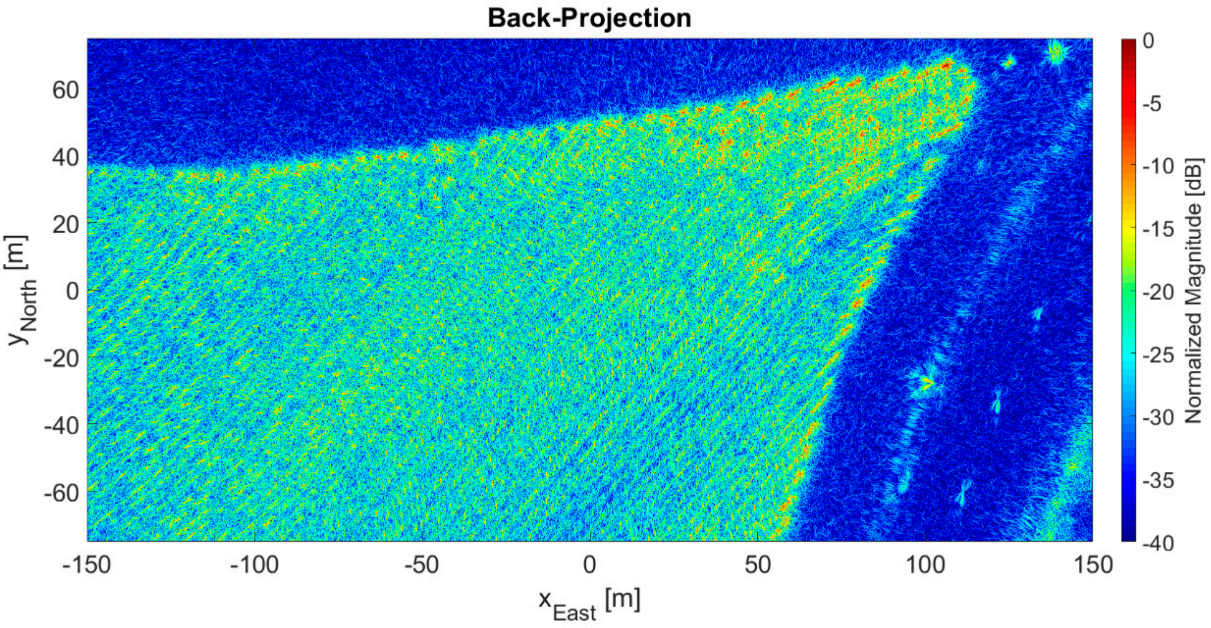


Figure 11. 2D output image processed by the BP algorithm.

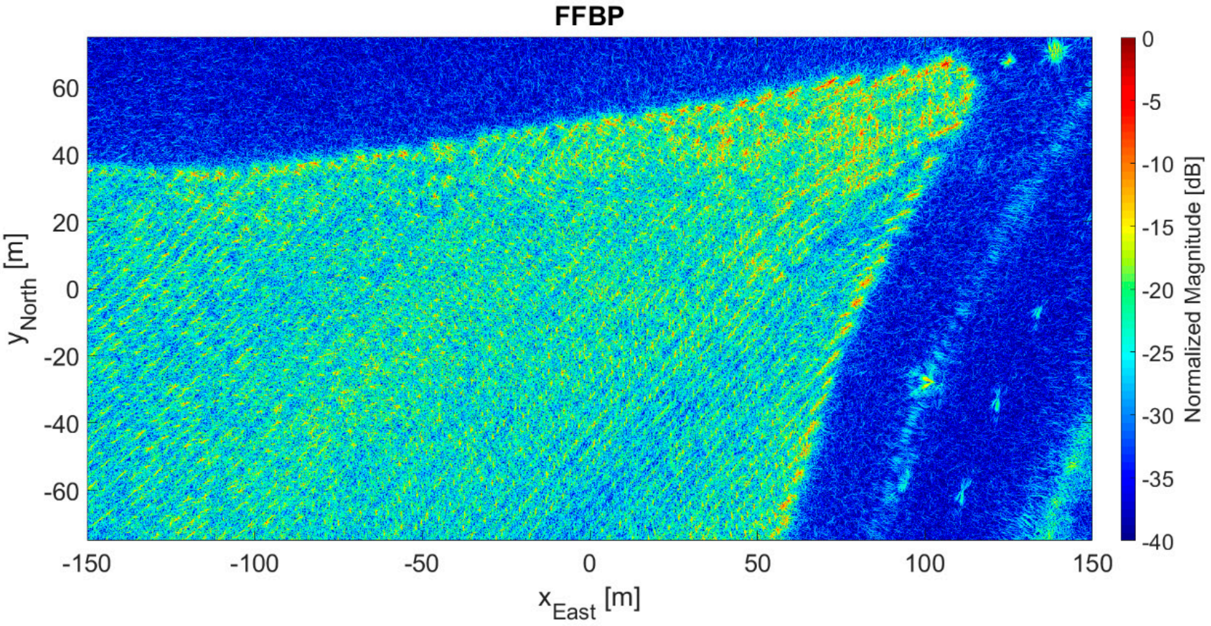


Figure 12. 2D output image processed by the FFBP algorithm with $L = 5$ and an $(8 \times 4 \times 1)$ initial partition.

3.2. Phase Error vs. SNR

Figure 13 presents the phase error response between the 2D images shown in Figures 11 and 12. Notice that the darkest area of Figure 11 corresponds to an increase of phase error in Figure 13, which indicates a noisy behavior. The mean normalized magnitude at a 30×30 m^2 square in the northwestern-most corner of Figure 11 is close to -40 dB. Thus, this value was considered the noise floor level for calculating the SNR for the following analysis.

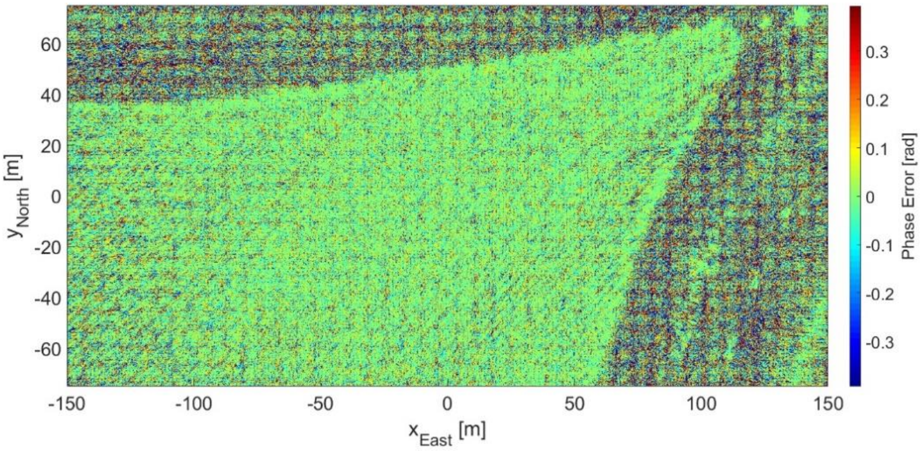


Figure 13. Phase error for the 2D FFBP image with $L = 5$ and an $(8 \times 4 \times 1)$ initial partition.

Figure 14 displays three histograms for the phase error. The first (Figure 14a) had no SNR threshold, i.e., all pixels are taken into account; the second (Figure 14b) had a 0 dB SNR threshold; and the last one (Figure 14c) had a 10 dB SNR threshold. As can be seen, between Figure 14a,b, there is a subtle change of less than 2% in relative probability for each bin. The corresponding decrease in phase error standard deviation was from 0.33 (18.8°) to 0.20 (11.4°) rad. On the other hand, between Figure 14a,c, there is a perceptible change of more than 8% in relative probability for the central bins, which made the phase error standard deviation decrease even more to 0.10 rad (5.8°).

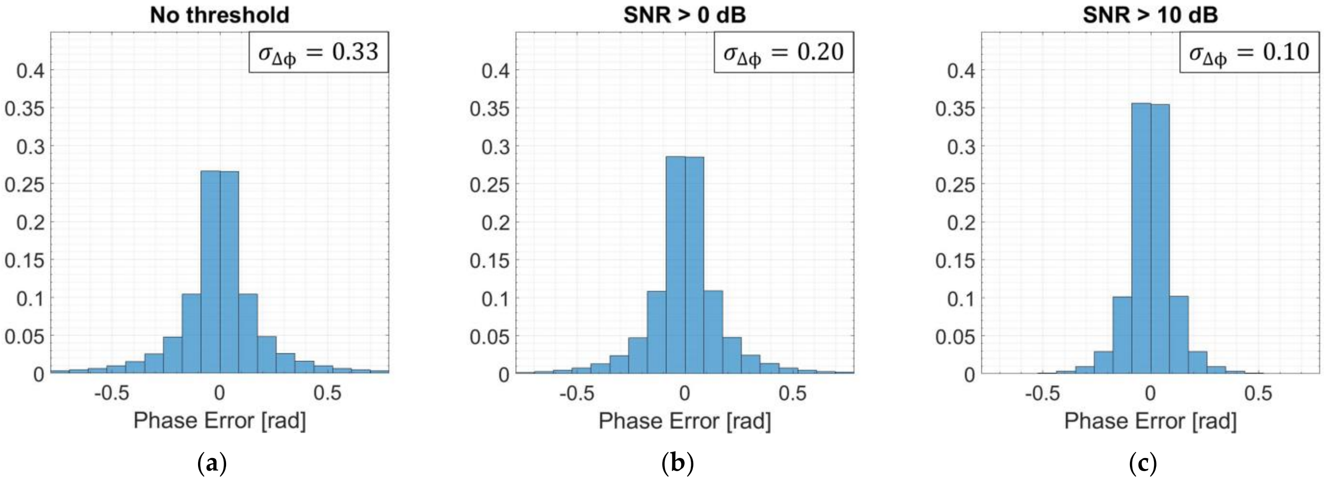


Figure 14. Histogram of the phase error for the 2D FFBP image with $L = 5$ and an $(8 \times 4 \times 1)$ initial partition for (**a**) no signal-to-noise ratio (SNR) threshold, (**b**) SNR >0 dB, and (**c**) SNR >10 dB.

As the 10 dB SNR threshold might have eliminated valuable information, the chosen threshold for the subsequent analysis was 0 dB SNR. By applying the selected SNR threshold to Figure 10, the resulting phase error standard deviation became 0.22 rad (12.7°).

3.3. Phase Error vs. Geometric Parameters

Figures 15 and 16 present scatter plots of the phase error standard deviation $\sigma_{\Delta\phi}$ versus β, defined by (11). Figure 15 shows two separate linear regressions for 2D and 3D images, while Figure 16 shows a unique linear regression for all results. In Figure 15, the slopes indicate lower phase errors for the 2D data than for the 3D data.

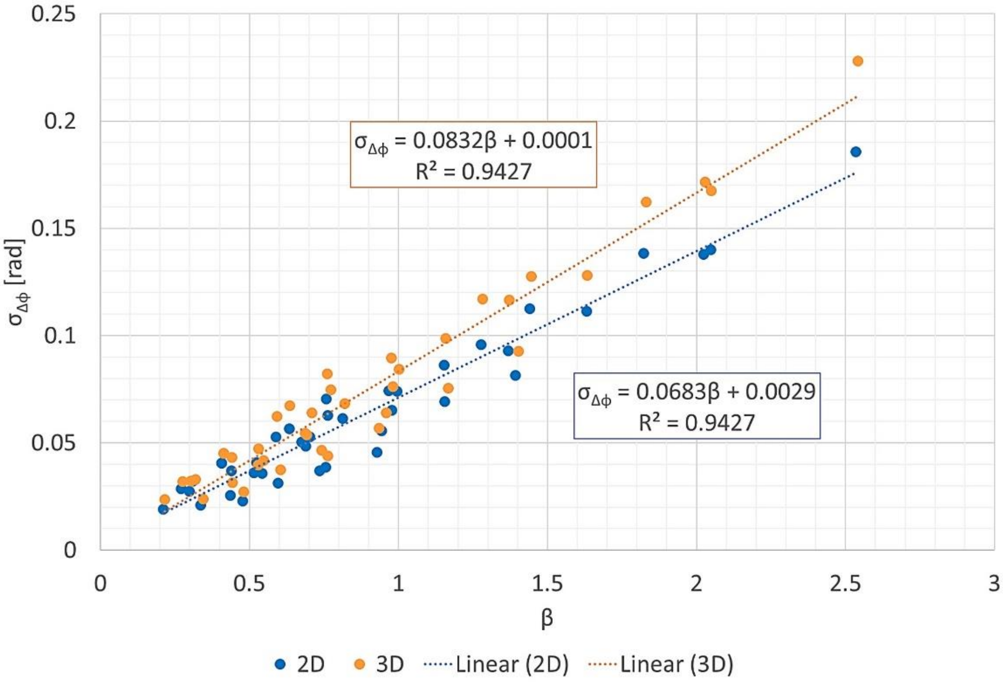


Figure 15. Phase error standard deviation $\sigma_{\Delta\phi}$ vs. β, with separate linear regressions for 2D and 3D data.

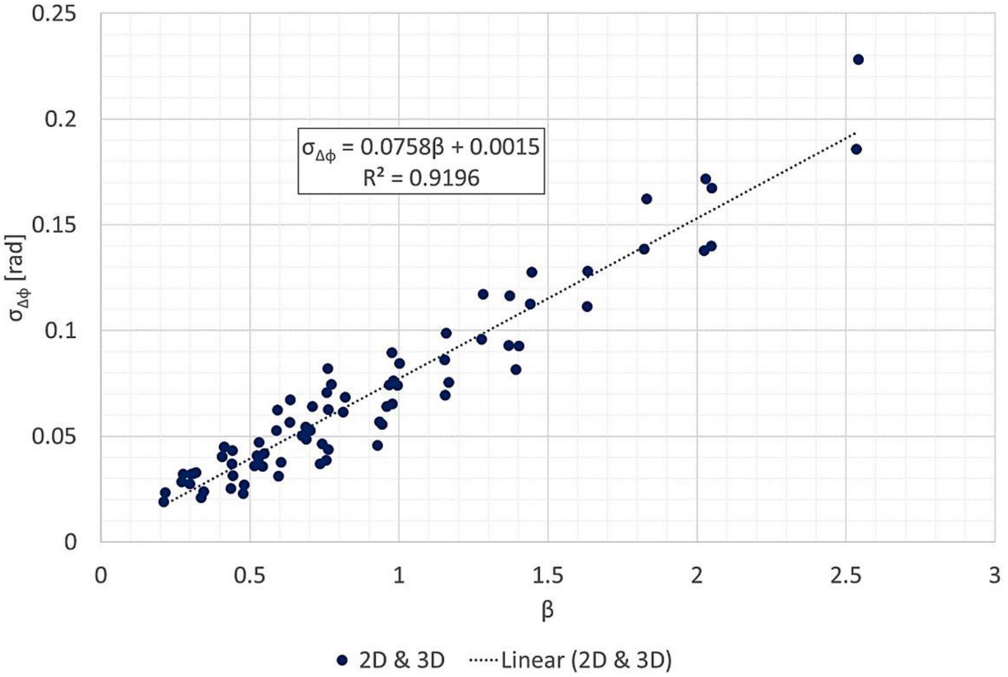


Figure 16. Phase error standard deviation $\sigma_{\Delta\phi}$ vs. β, with the same linear regression for both 2D and 3D data.

The statistics for all three linear regression models are indicated in Table 4. All intercepts had high p-values, and all of their confidence intervals contained zero. Thus, the intercepts were not statistically significant. On the other hand, all slopes had negligible p-values, and neither of their confidence intervals contained zero. Moreover, all linear regression models presented high coefficients of determination, $R^2 > 0.9$. Therefore, the hypothesis $\left(\sigma_{\Delta\phi} \propto \beta\right)$ is supported by the data. Moreover, the hypothesis is accepted even when combining 2D and 3D data (Figure 16).

Table 4. Statistics of the linear regression models.

Data	Coefficient	Estimate	Standard Error	95% Confidence Interval		p-Value	R^2
2D	Intercept Slope	0.0029 0.0683	0.0028 0.0027	−0.0029 0.0628	0.0086 0.0738	0.32 3.3×10^{-25}	0.9427
3D	Intercept Slope	0.0001 0.0832	0.0035 0.0033	−0.0069 0.0765	0.0071 0.0899	0.97 3.3×10^{-25}	0.9427
2D and 3D	Intercept Slope	0.0015 0.0758	0.0026 0.0025	−0.0038 0.0708	0.0067 0.0809	0.58 1.9×10^{-44}	0.9196

3.4. Phase Error vs. Time

Figure 17 presents the phase error standard deviation versus the processing time for the 2D images at different values of L. Figure 18 shows a similar line chart for the 3D images. Here, the phase error standard deviation was calculated for the whole image, not only for the selected image blocks of Figure 8.

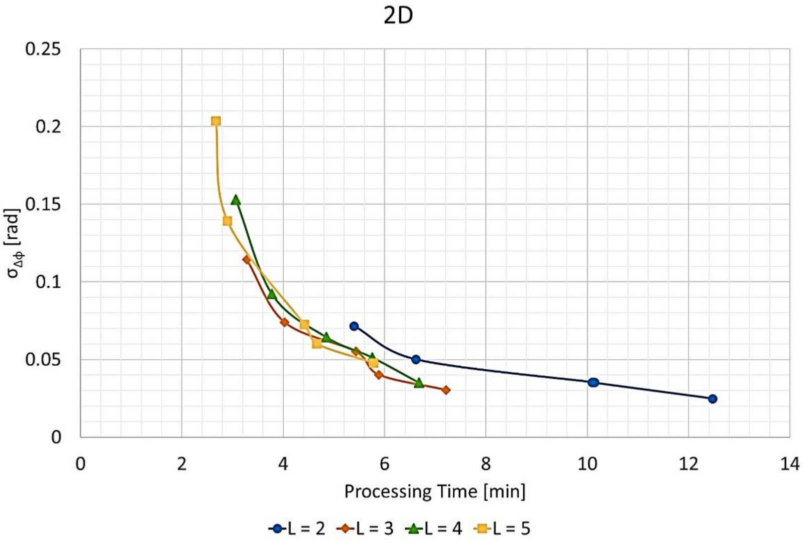


Figure 17. Phase error standard deviation $\sigma_{\Delta\phi}$ vs. processing time for the 2D output images and different values of L.

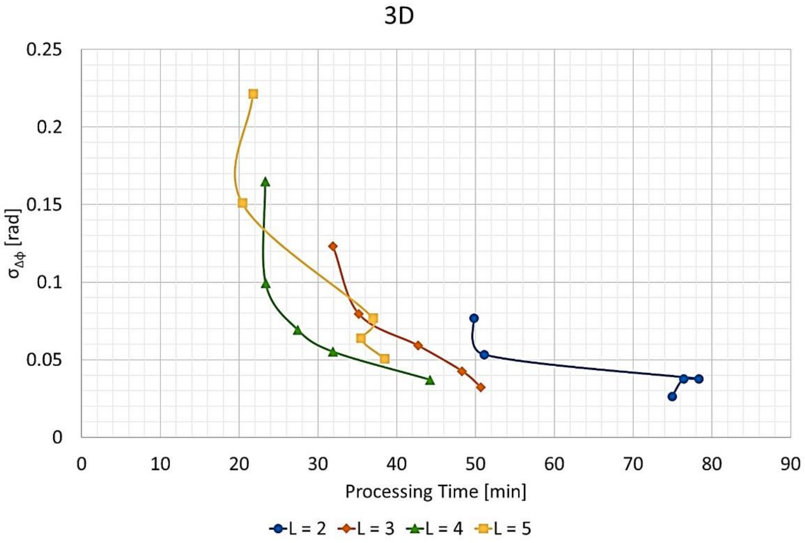


Figure 18. Phase error standard deviation $\sigma_{\Delta\phi}$ vs. processing time for the 3D output images and different values of L.

As can be seen, the curves for the 3D images (Figure 18) are not as smooth as those for the 2D images (Figure 17). The reason is that the function for defining the split scheme causes unnecessary waste and needs improvement. Beyond that, it is easy to notice that both curves for $L = 2$ are far slower than for other values of L.

Table 5 lists the slowest, fastest, and average processing times of the FFBP algorithm compared to the BP for 2D and 3D images. Table 5 also presents the corresponding speed-up factors. These results are from the same data sets of Figures 17 and 18. It is worth noticing that the speed-up factor was more pronounced for the 3D images.

Table 5. Processing time of the slowest, fastest, and average FFBP configurations compared to the BP algorithm.

Image Type	BP Processing Time	FFBP		
		Configuration	**Processing Time**	**Speed-Up Factor**
2D	35 min 33 s	Fastest	2 min 40 s	13.33
		Average	5 min 45 s	6.18
		Slowest	12 min 28 s	2.85
3D	7 h 12 min 18 s	Fastest	20 min 24 s	21.2
		Average	42 min 8 s	10.3
		Slowest	1 h 18 min 18 s	5.52

4. Discussion

The hypothesis that the geometric parameters at the first iteration can predict the phase error standard deviation at the output was validated for the P-band data. It was also validated when joining the 2D and 3D data sets (Figure 16), reinforcing the idea that what matters most for this FFBP algorithm is the diagonal of the subimages, not their width. In Section 3.3, all linear regression models produced slopes with negligible p-values, statistically irrelevant intercepts, and $R^2 > 0.9$.

This hypothesis was inspired by the range error analysis presented in [12] but disregarding the effect of any deviations from a linear flight path. The reason is that the phase compensation term (10) ensures good focusing quality for nonlinear flight patterns. This term was proposed by Zhang et al. [13] but with a different goal, namely to avoid taking range samples at each recursion in order to accelerate processing.

If (10) was removed from the FFBP algorithm, the outcome of the case study presented here would be completely unsatisfactory. Indeed, Figure 19 shows the resultant 2D image with $L = 2$ and a $(24 \times 12 \times 1)$ initial partition, i.e., the configuration with the lowest phase error standard deviation in Section 3.4. If Figure 19 is compared to the BP output image of Figure 11, the degree of coherence is a meager 0.12.

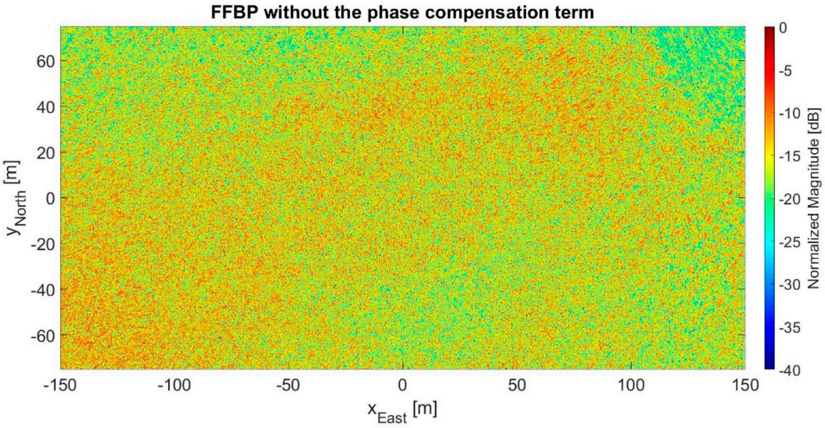


Figure 19. 2D output image processed by the FFBP algorithm without the phase compensation term (10) for the setup with L 2 and a $(24 \times 12 \times 1)$ initial partition.

According to the method for controlling the phase error proposed in [12] (and briefly described in Section 2.2), the partition scheme should attempt to keep the product of the subimage diagonal by the subaperture length constant across all iterations. This was possible in the processing of the 2D images but not for the 3D images. The reason is that the number of voxels in the x- and y-directions are significantly larger than in the z-direction. Therefore, in some setups, the volumetric images were only split across the x- and y-directions for the last iterations. Consequently, the linear regression of the 3D image data set had a slightly steeper slope than that of the 2D data set.

In the future, this methodology should be repeated for other frequencies as the phase error also depends on the radar wavenumber. The linear regression models can be used to determine processing parameters from a requirement in phase error, which would be more accessible for other users to benefit from the FFBP algorithm.

As expected, the configuration with the lowest image quality (see Figures 10 and 12) had the longest subaperture length and subimage diagonal, i.e., $L = 5$ with an $(8 \times 4 \times 1)$ initial partition. Likewise, the configuration with the highest image quality had the shortest subaperture length and subimage diagonal, i.e., $L = 2$ with a $(24 \times 12 \times 1)$ initial partition. Table 6 lists some figures of merit at these extremes for the 2D and 3D data sets, namely the phase error standard deviation, the degree of coherence, and an SNR of equivalent thermal noise, calculated according to [26]. SNR of equivalent thermal noise can be understood as the signal-to-thermal noise ratio that would result in an interferometric image with the same degree of coherence. Table 6 also shows the values for an average image quality, which corresponds to the following configurations:

- $L = 5$ with a $(16 \times 8 \times 1)$ initial partition for 2D;
- $L = 2$ with an $(8 \times 4 \times 1)$ initial partition for 3D.

It is important to note that the term "lowest quality" refers to a relative comparison within the data set, not to poor quality in absolute terms. Qualitatively, Figures 10 and 12 appear to be almost identical to Figures 9 and 11, which may well indicate that this level of image quality is suitable for SAR processing. Indeed, in [23], the same drone-borne SAR system produced a high-accuracy forest inventory with SAR interferometry in the P-band. A 5% accuracy was possible thanks to the forest SNR being higher than 17 dB. Because the SNR of equivalent noise was more than 20 dB, the configurations with the lowest image quality were already satisfactory. Moreover, they were also associated with the fastest processing times (see Table 5), with speed-up factors of 13 and 21 for 2D and 3D images, respectively.

Table 6. Performance of the configurations with highest, average, and lowest image quality.

Image Type	Figure of Merit	Image Quality		
		Highest	Average	Lowest
2D	Phase Error Standard deviation	0.025 rad (1.4°)	0.073 rad (4.2°)	0.20 (11.7°)
	Degree of coherence	0.9999	0.9993	0.9945
	SNR of equivalent Thermal noise	40 dB	31 dB	23 dB
3D	Phase error Standard deviation	0.026 (1.5°)	0.077 rad (4.4°)	0.22 rad (12.7°)
	Degree of coherence	0.9999	0.9988	0.9921
	SNR of equivalent Thermal noise	38 dB	29 dB	21 dB

On the other hand, the configurations with the highest image quality had unnecessarily slow processing times. If a specific application would require an SNR higher than 20 dB, then a configuration with average image quality could be employed. The average phase error standard deviation points were close to those with average processing time in Figures 17 and 18. Therefore, more demanding applications could benefit from a speed-up factor of about 6 for 2D images and 10 for 3D images.

Figures 9–12 show processed images from data acquired with a spiral flight path. As can be seen, the trees on the eucalyptus plantation are easily recognized. If the same area was surveyed with a linear flight pattern, the resultant image would have a slant range resolution of 3 m and an azimuth resolution of 50 cm [23]. However, thanks to the 360° acquisition, the resolution across all directions in the (x, y) plane was at least 50 cm. The maximum attainable resolution in the (x, y) plane would be $\sim \lambda_0/4$ [1–3].

Unsurprisingly, the speed-up factor was higher for 3D images than for 2D images. Section 2.1 pointed out that FFBP algorithms can reduce the computational cost from $\mathcal{O}(N^3)$ to $\mathcal{O}(N^2 \log N)$. Therefore, the expected speed-up factor $N/\log N$ would increase with the size of the output image.

It was noted in Section 3.4 that the function for creating the partition scheme needs improvement. Moreover, the current version of the algorithm assumes that the radar is constantly illuminating the imaged area. In future works, this assumption should no longer be required. Finally, a bistatic version of the algorithm should be implemented as well.

5. Conclusions

Spiral and multicircular SAR acquisition techniques can produce high-resolution 3D SAR images. In [20], the authors presented an FFBP algorithm capable of processing simulated SAR data replicating a spiral flight path. In the present work, an improved version of the FFBP algorithm [21] could successfully process real P-band SAR data acquired by a drone-borne SAR system that performed a spiral flight pattern.

This paper proposes a statistical phase error analysis to determine how the FFBP setup affects the quality of the output images. In the case study, the same raw radar data were processed with the FFBP algorithm with different parameters to produce several 2D and 3D SAR images. The analysis validates the hypothesis that geometric parameters defined at the beginning of processing can predict the phase error standard deviation at the output. In future works, the linear regression models generated in the analysis could be applied to determine the processing setup from a requirement in phase error.

The FFBP algorithm produces nearly identical images to those processed with a direct BP algorithm, only faster. The speed-up factor is up to 21 times for the 3D images and 13 times for 2D images, with a phase error standard deviation of ~12°, corresponding to an SNR of equivalent thermal noise of 20 dB. For higher image quality, with a phase error standard deviation of ~4° and 30 dB SNR of equivalent thermal noise, the speed-up factor is 10 and 6 times for the 3D and 2D images, respectively.

Author Contributions: Conceptualization, formal analysis, investigation, methodology, and software, J.A.G.; writing and visualization, J.A.G. and V.C.; supervision, L.S.B. and H.E.H.-F. All authors have read and agreed to the published version of the manuscript.

Funding: This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Brasil (CAPES), Finance Code 001.

Data Availability Statement: The data presented in this study are openly available in Zenodo at doi:10.5281/zenodo.4883258, reference number [21].

Acknowledgments: The authors would like to thank Radaz Ltda. for providing the raw radar data. We would also like to thank João Roberto Moreira Neto for his support and assistance.

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

References

1. Ishimaru, A.; Chan, T.-K.; Kuga, Y. An Imaging Technique Using Confocal Circular Synthetic Aperture Radar. *IEEE Trans. Geosci. Remote Sens.* **1998**, *36*, 1524–1530. [CrossRef]
2. Ponce, O.; Prats-Iraola, P.; Pinheiro, M.; Rodriguez-Cassola, M.; Scheiber, R.; Reigber, A.; Moreira, A. Fully Polarimetric High-Resolution 3-D Imaging with Circular SAR at L-Band. *IEEE Trans. Geosci. Remote Sens.* **2013**, *52*, 3074–3090. [CrossRef]
3. Moore, L.; Potter, L.; Ash, J. Three-Dimensional Position Accuracy in Circular Synthetic Aperture Radar. *IEEE Aerosp. Electron. Syst. Mag.* **2014**, *29*, 29–40. [CrossRef]
4. Ponce, O.; Prats-Iraola, P.; Scheiber, R.; Reigber, A.; Moreira, A. First Airborne Demonstration of Holographic SAR Tomography with Fully Polarimetric Multicircular Acquisitions at L-Band. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6170–6196. [CrossRef]

5. Zhu, S.; Zhang, Z.; Liu, B.; Yu, W. Three-Dimensional High Resolution Imaging Method of Multi-Pass Circular SAR Based on Joint Sparse Model. In Proceedings of the EUSAR 2016: 11th European Conference on Synthetic Aperture Radar, Hamburg, Germany, 6–9 June 2016; pp. 1–5.
6. Bao, Q.; Lin, Y.; Hong, W.; Shen, W.; Zhao, Y.; Peng, X. Holographic SAR Tomography Image Reconstruction by Combination of Adaptive Imaging and Sparse Bayesian Inference. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1248–1252. [CrossRef]
7. Farhadi, M.; Jie, C. Distributed Compressive Sensing for Multi-Baseline Circular SAR Image Formation. In Proceedings of the 2017 IEEE International Conference on Imaging Systems and Techniques (IST), Beijing, China, 18–20 October 2017; pp. 1–6.
8. Feng, D.; An, D.; Huang, X.; Zhou, Z. Holographic SAR Tomographic Processing of the Multicircular Data. In Proceedings of the 2018 Asia-Pacific Microwave Conference (APMC), Kyoto, Japan, 6–9 November 2018; pp. 830–832.
9. Feng, D.; An, D.; Chen, L.; Huang, X.; Zhou, Z. Multicircular SAR 3-D Imaging Based on Iterative Adaptive Approach. In Proceedings of the 2019 6th Asia-Pacific Conference on Synthetic Aperture Radar (APSAR), Xiamen, China, 26–29 November 2019; pp. 1–5.
10. Lv, C.; Deng, B.; Zhang, Y.; Yang, Q.; Wang, H. Terahertz Spiral SAR Imaging Algorithms and Simulations. In Proceedings of the 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 12–14 June 2020; Volume 1, pp. 1746–1750.
11. McCorkle, J.W.; Rofheart, M. Order N^2 Log(N) Backprojector Algorithm for Focusing Wide-Angle Wide-Bandwidth Arbitrary-Motion Synthetic Aperture Radar. In Proceedings of the Radar Sensor Technology; International Society for Optics and Photonics, Orlando, FL, USA, 17 June 1996; Volume 2747, pp. 25–36.
12. Ulander, L.M.H.; Hellsten, H.; Stenstrom, G. Synthetic-Aperture Radar Processing Using Fast Factorized Back-Projection. *IEEE Trans. Aerosp. Electron. Syst.* **2003**, *39*, 760–776. [CrossRef]
13. Zhang, H.; Tang, J.; Wang, R.; Deng, Y.; Wang, W.; Li, N. An Accelerated Backprojection Algorithm for Monostatic and Bistatic SAR Processing. *Remote Sens.* **2018**, *10*, 140. [CrossRef]
14. Liang, D.; Zhang, H.; Fang, T.; Lin, H.; Liu, D.; Jia, X. A Modified Cartesian Factorized Backprojection Algorithm Integrating with Non-Start-Stop Model for High Resolution SAR Imaging. *Remote Sens.* **2020**, *12*, 3807. [CrossRef]
15. Zhang, M.; Li, H.; Wang, G. High-Precision Spotlight SAR Imaging with Joint Modified Fast Factorized Backprojection and Autofocus. *Int. J. Antennas Propag.* **2020**, *2020*, e2495050. [CrossRef]
16. Li, X.; Zhou, S.; Yang, L. A New Fast Factorized Back-Projection Algorithm with Reduced Topography Sensibility for Missile-Borne SAR Focusing with Diving Movement. *Remote Sens.* **2020**, *12*, 2616. [CrossRef]
17. Song, X.; Yu, W. Processing Video-SAR Data with the Fast Backprojection Method. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 2838–2848. [CrossRef]
18. Xie, H.; Shi, S.; An, D.; Wang, G.; Wang, G.; Xiao, H.; Huang, X.; Zhou, Z.; Xie, C.; Wang, F.; et al. Fast Factorized Backprojection Algorithm for One-Stationary Bistatic Spotlight Circular SAR Image Formation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 1494–1510. [CrossRef]
19. Guo, Z.; Zhang, H.; Ye, S. Cartesian Based FFBP Algorithm for Circular SAR Using NUFFT Interpolation. In Proceedings of the 2019 6th Asia-Pacific Conference on Synthetic Aperture Radar (APSAR), Xiamen, China, 26–29 November 2019; pp. 1–5.
20. Góes, J.A.; Castro, V.; Sant'Anna Bins, L.; Hernandez-Figueroa, H.E. 3D Fast Factorized Back-Projection in Cartesian Coordinates. In Proceedings of the 2020 IEEE Radar Conference (RadarConf20), Florence, Italy, 21–25 September 2020; pp. 1–6.
21. Góes, J.A. *3D FFBP*; Zenodo: Geneva, Switzerland, 2021. [CrossRef]
22. Moreira, L.; Castro, F.; Góes, J.A.; Bins, L.; Teruel, B.; Fracarolli, J.; Castro, V.; Alcântara, M.; Oré, G.; Luebeck, D.; et al. A Drone-Borne Multiband DInSAR: Results and Applications. In Proceedings of the 2019 IEEE Radar Conference (RadarConf), Boston, MA, USA, 22–26 April 2019; pp. 1–6.
23. Moreira, L.; Lübeck, D.; Wimmer, C.; Castro, F.; Góes, J.A.; Castro, V.; Alcântara, M.; Oré, G.; Oliveira, L.P.; Bins, L.; et al. Drone-Borne P-Band Single-Pass InSAR. In Proceedings of the 2020 IEEE Radar Conference (RadarConf20), Florence, Italy, 21–25 September 2020; pp. 1–6.
24. Morton, G.M. *A Computer Oriented Geodetic Data Base and a New Technique in File Sequencing*; IBM: Otawa, ON, Canada, 1966. Available online: https://dominoweb.draco.res.ibm.com/reports/Morton1966.pdf (accessed on 26 July 2021).
25. Connor, M.; Kumar, P. Fast Construction of K-Nearest Neighbor Graphs for Point Clouds. *IEEE Trans. Vis. Comput. Graph.* **2010**, *16*, 599–608. [CrossRef] [PubMed]
26. Just, D.; Bamler, R. Phase Statistics of Interferograms with Applications to Synthetic Aperture Radar. *Appl. Opt.* **1994**, *33*, 4361–4368. [CrossRef] [PubMed]