remote sensing

Article

An Improved Back-Projection Algorithm for GNSS-R BSAR Imaging Based on CPU and GPU Platform

Shiyu Wu, Zhichao Xu, Feng Wang *, Dongkai Yang and Gongjian Guo

School of Electronics and Information Engineering, Beihang University, Beijing 100191, China;
shiyu_wu@buaa.edu.cn (S.W.); by1602112@buaa.edu.cn (Z.X.); edkyang@buaa.edu.cn (D.Y.);
ricky11inter@buaa.edu.cn (G.G.)
* Correspondence: wangfeng_buaa@buaa.edu.cn; Tel.: +86-18801493608

Abstract: Global Navigation Satellite System Reflectometry Bistatic Synthetic Aperture Radar (GNSS-R BSAR) is becoming more and more important in remote sensing because of its low power, low mass, low cost, and real-time global coverage capability. The Back Projection Algorithm (BPA) was usually selected as the GNSS-R BSAR imaging algorithm because it can process echo signals of complex geometric configurations. However, the huge computational cost is a challenge for its application in GNSS-R BSAR. Graphics Processing Units (GPU) provides an efficient computing platform for GNSS-R BSAR processing. In this paper, a solution accelerating the BPA of GNSS-R BSAR using GPU is proposed to improve imaging efficiency, and a matching pre-processing program was proposed to synchronize direct and echo signals to improve imaging quality. To process hundreds of gigabytes of data collected by a long-time synthetic aperture in fixed station mode, a stream processing structure was used to process such a large amount of data to solve the problem of limited GPU memory. In the improvement of the imaging efficiency, the imaging task is divided into pre-processing and BPA, which are performed in the Central Processing Unit (CPU) and GPU, respectively, and a pixel-oriented parallel processing method in back projection is adopted to avoid memory access conflicts caused by excessive data volume. The improved BPA with the long synthetic aperture time is verified through the simulation of and experimenting on the GPS-L5 signal. The results show that the proposed accelerating solution is capable of taking approximately 128.04 s, which is 156 times lower than pure CPU framework for producing a size of 600 m × 600 m image with 1800 s synthetic aperture time; in addition, the same imaging quality with the existing processing solution can be retained.

Keywords: Global Navigation Satellite System Reflectometry (GNSS-R); Back-Projection Algorithm (BPA); Graphics Processing Units (GPU)

check for updates

Citation: Wu, S.; Xu, Z.; Wang, F.; Yang, D.; Guo, G. An Improved Back-Projection Algorithm for GNSS-R BSAR Imaging Based on CPU and GPU Platform. *Remote Sens.* **2021**, *13*, 2107. https://doi.org/10.3390/rs13112107

Academic Editors: Joan Francesc Munoz-Martin, Mary Morris and Nereida Rodriguez-Alvarez

Received: 9 April 2021
Accepted: 26 May 2021
Published: 27 May 2021

1. Introduction

The concept of GNSS-R was firstly considered by Hall and Cordy in 1988 [1], which uses the reflected GNSS signals to obtain information about the Earth's surface; however, they consider the technology to be less promising. The concept of ocean altimetry using the GNSS-R was proposed by Martin-Neira in 1993 [2]. In recent years, GNSS-R technology has achieved rapid development, both in the measurement concepts, instruments, and applications, such as retrieving wind speed [3–6], measuring sea-surface level [7–10], estimating soil moisture and biomass [11–16], detecting oil spill [17], detecting sea ice [18], etc.

GNSS-R BSAR is one of the emerging applications of GNSS-R technology [19], which uses the GNSS satellites as non-cooperative illuminators and employs bistatic SAR technology to obtain an image of the Earth's surface. This technology has been proven to be applicable to target detection and recognition [20], deformation monitoring [21,22], etc. It is also evident that more precise phase-based analyses are required for deformation monitoring, which is likely to be the topic of our future activities. The advantage of GNSS-R BSAR is significant compared with traditional SAR. Firstly, as no additional transmitter is needed, the GNSS-R BSAR equipment can be made low mass, low power, and low cost. Secondly, the GNSS constellation has more than 140 satellites, which can provide more than 20 satellites in any certain area to observe the Earth's surface from multiple angles. Furthermore, abundant satellite resources provide the feasibility of fusing multiple satellites at different angles, not only to improve the spatial resolution and time resolution of the system but also to obtain more information on the characteristics of the observation area [23]. Finally, GNSS can provide precise timing services to ensure the synchronization performance of the technology.

GNSS signals are not designed for radar or remote sensing; therefore, this has caused some problems in the application of GNSS-R BSAR. Firstly, the power density of GNSS signals near the surface of the Earth is relatively low, but a sufficiently high signal-to-noise ratio (SNR) can be obtained by increasing the synthetic aperture time [24]. Secondly, the signal bandwidth of GNSS limits its range resolution. Thirdly, the instability of signal power on the Earth's surface causes signal power calibration problems. Finally, due to its complex bistatic geometric structure, these improved frequency domain imaging methods still have limitations, such as the Range Doppler Algorithm (RDA) [25], Chirp Scaling Algorithm (CSA) [26,27], etc. With the increase in synthetic aperture time and the deterioration of geometry, the image quality of the frequency domain imaging methods will deteriorate further [28]. The BPA is better suited to GNSS-R BSAR because of its ability to handle different geometrical configurations of echo signals, and the fact that it is not restricted by synthetic aperture time [19]. However, the computational cost of BPA is huge. Therefore, the huge computational cost is a challenge for the application of BPA to GNSS-R BSAR. To reduce the computational cost, there are two methods for solving this problem:

1. Using Fast Back Projection Algorithm (FBPA) for GNSS-R BSAR imaging has the advantage of reducing the computational cost, but it will reduce the quality of the imaging [29].
2. Using GPU to accelerate BPA in parallel; this method will increase the complexity of the system but will obtain the optimal imaging quality [30–35].

In this paper, our research focuses on the use of GPU to accelerate BPA in GNSS-R BSAR. An improved BPA was proposed for making it suitable for GPU parallel implementation. Firstly, a pre-processing program was designed to obtain parameter information matching BPA. In the pre-processing program, to ensure the synchronization of the direct signal and the echo signal, the echo signal is synchronized with the accurate pseudorange and code phase information in the direct signal. The precision ephemeris of the GPS is introduced to improve the orbit accuracy of the navigation satellites and the positioning accuracy of the receivers, which improves the imaging quality of the BPA algorithm. Secondly, the parallel acceleration strategy of BPA in CPU and GPU heterogeneous architecture is designed and implemented. In fixed station mode, the GNSS-R SAR system may work for thousands of seconds on a single mission resulting in the acquisition of data that can reach hundreds of gigabytes. A stream processing structure was used to process such a large amount of data. According to the memory size of the GPU, the original data are divided into several subdata in chronological order and transmitted to GPU. To optimize calculation efficiency, the preprocess program is allocated to the CPU for execution, after which the BPA is allocated to the CPU for execution.

To prove the feasibility of the proposed imaging algorithm, simulations and experiments were carried out using a GPS-L5 signal in the ground fixed station mode. In simulation and experiment, the time consumed by the proposed algorithm is 163.96 and 156 times less than that consumed by BPA running only on CPU, respectively. However, the image quality has not been reduced. The simulation and experimental results prove the feasibility and effectiveness of the proposed algorithm.

This paper is organized into six sections. In Section 2, the signal model of GNSS-R BSAR is reviewed briefly. In Section 3, the improved BPA is discussed in detail. A GPU parallel acceleration method compatible with the improved BPA is proposed and analyzed in Section 4. The simulation and experimental results are provided in Section 4. A further discussion of the proposed algorithm is presented in Section 5, and conclusions are drawn in Section 6.

2. Materials

We consider a ground-based stationary receiving platform collecting the signals emitted from a GNSS satellite and reflected by a stationary scene. Figure 1 shows the geometric configuration of GNSS-R BSAR, where the origin O is set to center of the XYZ coordinate system, $P(x_p, y_p, 0)$ is an arbitrary point target in the imaging area (the targets are assumed to be located on the XOY plane). T and R are the transmitter (navigation satellite) and fixed receiver, respectively. The transmitter is located at $T(x_T(t), y_T(t), z_T(t))$, while the receiver at (x_R, y_R, z_R). $R_T(t)$ is the distance between the satellite T and target P, R_R is the distance between the receiver R and target P, $R_B(t)$ is the distance between the satellite T and receiver R. $R_T(t)$, R_R, and $R_B(t)$ can be expressed as

$$R_T(t) = \sqrt{(x_T(t) - x_p)^2 + (y_T(t) - y_p)^2 + (z_T(t) - z_p)^2} \tag{1}$$

$$R_R = \sqrt{(x_R - x_p)^2 + (y_R - y_p)^2 + (z_R - z_P)^2} \tag{2}$$

$$R_B(t) = \sqrt{(x_T(t) - x_R)^2 + (y_T(t) - y_R)^2 + (z_T(t) - x_R)^2} \tag{3}$$

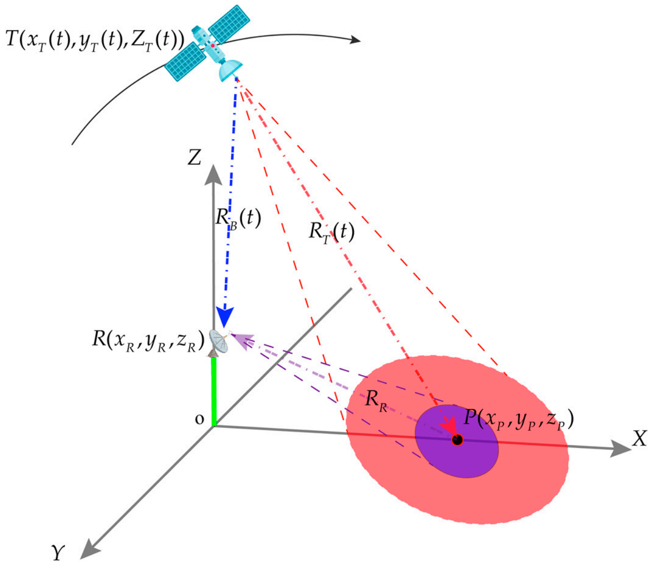


Figure 1. Geometric configuration of the GNSS-R BSAR.

The range history is $R(t) = R_T(t) + R_R$.

GNSS-R BSAR's receiver contains two channels, a direct channel, and an echo channel. The direct channel connects a Right-Handed Circularly Polarized (RHCP) omnidirectional antenna for receiving direct signals from GNSS satellites, which is used to realize system positioning and obtain accurate carrier phase and code phase to provide reference information for signal synchronization. Based on the Stop–Go model [36], the length of a ranging code (for GPS C/A code is 1 ms) is equivalent to the width of the range pulse. After quadrature demodulation and SAR data formatting [36], the direct signal can be expressed as a two-dimensional form

$$s(\eta, \tau) = W(\eta)N(\eta)C\left[\tau - \frac{R_B(\eta)}{c}\right]\exp\left[-j2\pi f_c \frac{R_B(\eta)}{c}\right] \tag{4}$$

η is the azimuth time, also called slow time, τ is the range time, also called fast time. $W(\eta)$ is the envelope of the received signal in the azimuth dimension, which is a rectangular window. $N(\eta)$ is the data code of the navigation signal, $C(\tau)$ is the ranging code sequence, f_c is the carrier frequency of the transmitted signal, and c is the speed of light. The echo channel connects a high-gain Left-Handed Circular Polarized (LHCP) antenna to receive the echo signal from the target area. After removing the data code, the echo signal can be expressed as

$$s_r(\eta,\tau) = W(\eta)C\left[\tau - \frac{R(\eta)}{c}\right]\exp\left[-j2\pi f_c\frac{R(\eta)}{c}\right] \tag{5}$$

The direct channel and echo channel use the same oscillator to mix the sampling signal, and the analog-to-digital converter chips for sampling the direct signal and echo use the same clock. After demodulation, the signals are sampled and stored by a large storage data collector for postprocessing.

3. Methods

3.1. Review on BPA

The BPA is a typical time-domain imaging algorithm [37], which comes from computer tomography. Firstly, the imaged plane is divided into grids, and the center of each grid is determined as the pixel point. Secondly, range compression used matched filtering to achieve pulse compression in the range. Thirdly, in the range–azimuth time domain, the coherent accumulation method is used to realize the azimuth focusing of each pixel point by point. Each pixel in the imaged plane is traversed to calculate the position of the target in each echo, and the phase is compensated for coherent accumulation to obtain the pixel value of the current point.

3.2. Data Pre-Processing

To optimize the imaging quality of the BPA algorithm and extract the parameters needed to accelerate BPA in parallel using the GPU, a pre-processing program is designed for a GNSS-R BSAR system. Three crucial technical issues relating to the program are considered. Firstly, it is necessary to ensure the synchronization of the direct signal and the echo signal during the entire imaging process. Secondly, BPA requires accurate time, and the location of the transmitter, receiver, and imaging area [38]. Finally, the relevant parameters (the reference point, imaging area, synthetic aperture time, the pixel interval) of the BPA algorithm are obtained. The pre-processing program was designed for the above three issues; the specific process is shown in Figure 2.

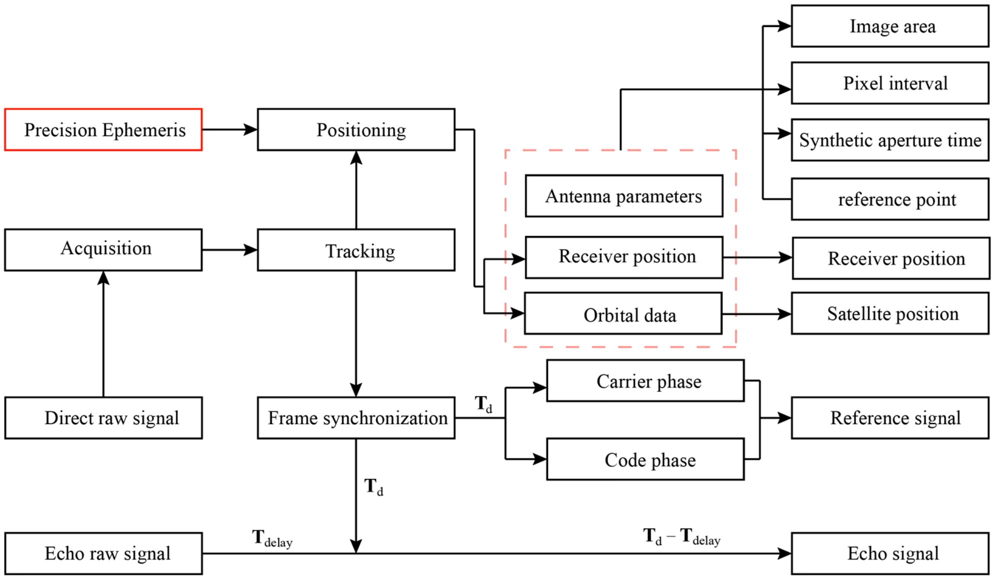


Figure 2. Pre-processing for BPA running on GPU.

The range compression in the BPA of GNSS-R SAR is achieved through correlation operations between the direct channel and the echo channel. The performance of the autocorrelation depends on the quality of the signal synchronization between the direct and echo channel. Ideally, the direct channel and the echo channel would have good channel coherence, but unfortunately, the signal can be affected by various factors. To solve the problem of time synchronization, the signals of the direct channel and the echo channel are processed, as shown in Figure 2. The GNSS direct signal is captured and tracked to obtain a subframe start timestamp position T_d of the direct signal and calculate the time delay difference T_{delay} between the echo signal and the direct signal due to the different propagation paths. To ensure the time and phase synchronization of the two signals, the direct signal of 1 ms (a ranging code length) taken by timestamp T_d, and the complete pseudorandom noise code of zero-initial phase is reconstructed as a reference signal. According to Phase-locked loop (PLL) theory [39], after compensating for the time difference T_{delay}, we can obtain the direct signal and the echo signal, which was emitted by satellite at the same time. The direct signal and the echo signal have good time synchronization performance by this operation. The expression of the reference signal is

$$h = C[\tau - \frac{R(\eta)}{c}] \tag{6}$$

The method for improving the satellite orbit accuracy and positioning accuracy is shown in Figure 2. International GNSS Service (IGS) Ultra-Rapid Precision Ephemeris was used to extract satellite obit data and calculate its position with less than 10 cm position error and satellite clock 5 ns error [40]. As IGS Ultra-Rapid data has a 15 min update rate, interpolation was necessary to meet the sampling rate of the GNSS-R BSAR system.

As shown in Figure 2, The method of using geometric modeling is used to determine the specific parameters of the BPA imaging process. Taking the ground fixed station mode as an example, the reference point is the position of the phase center of the echo antenna. The imaged area is the area covered by the main lobe of the echo antenna. The synthetic aperture time can be selected as a period of time between the navigation satellite antenna beam enters the imaged area to the time when the beam completely leaves the imaged area. To obtain a high-resolution image, it is necessary to determine the start and end times of the synthetic aperture according to the geometric structure of acquisition time [28]. Finally, the imaged plane is determined according to the coordinate information of the imaged area. In this paper, the height of the imaged plane is determined by introducing the Google Elevation Application Programming Interface (API), which can reduce the impact of elevation errors on the imaged quality. The pixel interval is estimated by the range and azimuth resolution at the time of acquisition. The minimum requirement is that the pixel interval sampling for the range and azimuth resolution satisfies the Nyquist sampling theorem.

3.3. Theoretical Analysis of the Improved BPA

An improved BP algorithm is proposed for GNSS-R BSAR, which can optimize the signal processing of heterogeneous architecture platform of CPU and GPU. The GNSS-R SAR system may work for thousands of seconds on a single mission resulting in the acquisition of data that can reach hundreds of gigabytes. A stream processing structure was used to process such a large amount of data.

The block diagram of the proposed algorithm is shown in Figure 3. The algorithm can be divided into three steps: range compression, back projection and phase compensation, and azimuth compression.

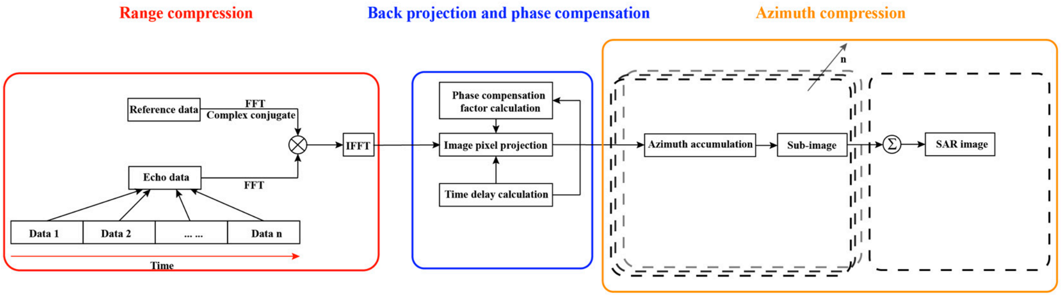


Figure 3. A block diagram of the improved BPA.

3.3.1. Range Compression

As the ranging code of the navigation satellite signal has good autocorrelation characteristics, the range compression is accomplished by autocorrelating the echo signal with the reference signal. Taking the C/A code of GPS L1 signal as an example, its autocorrelation peak is 24 dB higher than the sidelobe and cross-correlation results. Under a Gaussian white noise background, the autocorrelation gain is about 30 dB. In this paper, the calculation speed is accelerated by converting to the frequency domain, due to which the range autocorrelation is relatively computationally intensive. As shown in Figure 3, firstly, due to the actual collected echo data of GNSS-R BSAR consisting of hundreds of gigabytes, its capacity greatly exceeds the size of the video memory (tens of gigabytes) [30]. The proposed solution is to equally divide the echo data into multiple data segments and process them sequentially according to time. Secondly, the direct signal and the echo signal are converted to frequency domain for range compression, the specific implementation process is described in [25]. At a given azimuthal moment, the signal after range compression is expressed as:

$$F(\eta,\tau) = P[\tau - \frac{R(\eta)}{c}]\exp[-j2\pi\frac{R(\eta)}{\lambda}] \tag{7}$$

$P(\tau)$ represents the autocorrelation function of the ranging code.

3.3.2. Back Projection and Phase Compensation

The focus processing of BPA in the azimuth direction is carried out in the time domain and does not require Range Cell Migration Correction (RCMC). $F(\eta,\tau)$ is the correlation value at different time delays relative to the reference point, and the process of back projection is to map the $F(\eta,\tau_{mn})$ to different pixels (x_m, y_n), (see Figure 4) (m,n) is the image pixel index of the final image.

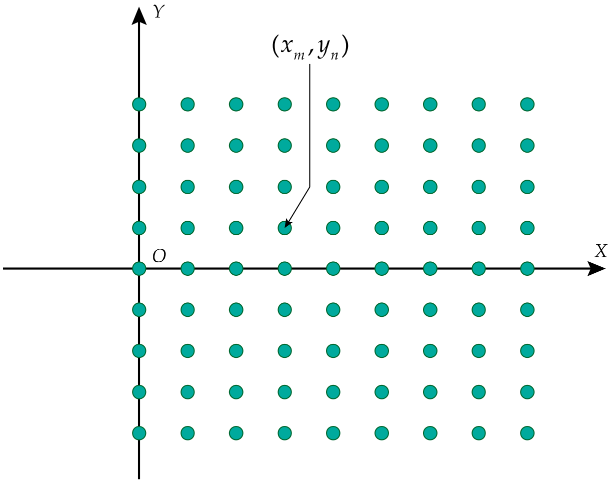


Figure 4. Schematic diagram of imaging pixel distribution.

The propagation delay can be calculated according to the signal propagation distance when the three positions of the navigation satellite, the pixels, and the receiver are uniquely determined, which can be expressed as:

$$\tau_{mn}(\eta) = \frac{R_T(\eta, m, n) + R_R(m, n)}{c} \tag{8}$$

$$R_T(\eta, m, n) = \sqrt{[x_T(\eta) - x(m, n)]^2 + [y_T(\eta) - y(m, n)]^2 + [z_T(\eta) - z(m, n)]^2} \tag{9}$$

$$R_R(m, n) = \sqrt{[x_R - x(m, n)]^2 + [y_R - y(m, n)]^2 + [z_R - z(m, n)]^2} \tag{10}$$

$$F(\eta, \tau_{mn}) = P[\tau_{mn} - \frac{R_T(\eta, m, n) + R_R(m, n)}{c}]\exp[-j2\pi\frac{R_T(\eta, m, n) + R_R(m, n)}{\lambda}] \tag{11}$$

$\tau_{mn}(\eta)$ is the time delay of the echo signal generated at the pixel (x_m, y_n). $R_T(\eta, m, n)$ and $R_R(m, n)$ are the distance from the satellite and receiver to the pixel (x_m, y_n). Before $F(\eta, \tau_{mn})$ is mapped to the pixel, the Doppler phases need be compensated. The Doppler phase compensation factor can be expressed as:

$$h(\eta, m, n) = \exp[j2\pi\frac{R_T(\eta, m, n) + R_R(m, n)}{\lambda}] \tag{12}$$

The Doppler phase compensation is realized by complex multiplication $h(\eta, m, n)$ in the time domain according to the data selected by the slant range $R_T(\eta, m, n) + R_R(m, n)$, and the result is used as the value of the pixel. An image can be generated at each azimuth time η, which can be expressed as

$$S(\eta, x_m, y_n) = F(\eta, \tau_{mn}) \cdot h(\eta, m, n) \tag{13}$$

3.3.3. Azimuth Compression

The azimuth compression of a data segment is achieved by coherently accumulating $S(\eta, x_m, y_n)$, the result obtained is defined as a subimage, which can be expressed as

$$S_{sub}(x_m, y_n) = \int_{-\frac{T}{2}}^{\frac{T}{2}} S_{sub}(\eta, x_m, y_n) d\eta \tag{14}$$

T is the time length of each data segment and the synthetic aperture time. The final SAR image is generated by accumulating each subimage, which can be expressed as

$$S(x_m, y_n) = \sum_{i=1}^{N} S_{sub}(x_m, y_n) \tag{15}$$

N is the total number of data segments.

As we all know, with the increase of scene area and synthetic aperture time, the computational cost of BPA will increase sharply. This makes BPA difficult to adopt in SAR technology. Fortunately, the two main steps (the range compression and the back-projection) of BPA can be highly parallelized. A software program taking advantage of a GPU can benefit from speed improvements thanks to parallel computing. In the next section, we will analyze these two steps.

3.4. Parallelized BPA on GPU

To optimize the processing process, a heterogeneous processing architecture was designed. Figure 5 shows the system block diagram of the proposed GNSS-R BSAR signal processing. In the heterogeneous framework, the GPU and the CPU process the data serially. The pre-processing part is mainly executed in CPU, the BPA is executed in GPU, finally, the image is sent back to CPU for output and storage.

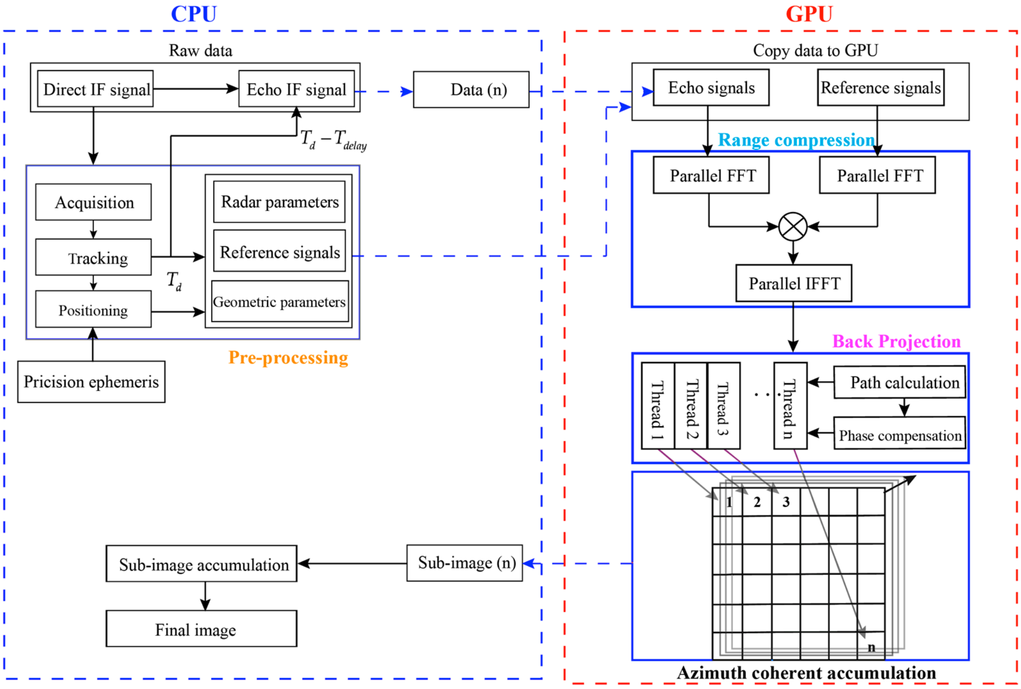


Figure 5. GNSS-R BSAR signal processing flowchart.

3.4.1. Parallelized Range Compression on GPU

It can be seen from Section 3 that the range compression is realized by converting to the frequency domain. As shown in Figure 5, the range compression module is all processed on the GPU, which includes main operations such as FFT and IFFT operation, complex multiplication, and complex addition. FFT and IFFT operations can be completed through the cuFFT library provided by Compute Unified Device Architecture (CUDA). The parallel complex multiplication and complex addition in the frequency domain are completed by constructing a kernel function. It is worth noting that the operation of FFT and IFFT will take up a lot of video memory resources, and the data size needs to be reasonably planned according to the size of the video memory to prevent memory overflow.

In the CUDA architecture, a thread is the smallest execution unit of a GPU that corresponds to the streaming processor (SP) of hardware. The threads are allocated to the number of range gates sampled according to the range, and each thread is solely responsible for the complex multiplication of the echo signal and the reference signal by a range gate data. The main steps of range compression on GPU include:

(1) N_a and N_r are determined according to the size of one segment data, where N_a and N_r are the number of range and azimuth samples, respectively. The memory of CPU and GPU are requested for the host for the corresponding data, respectively, and the data are transferred from the memory of the CPU to the memory of the GPU.

(2) The cuFFT library function of CUDA is used to complete the FFT of the reference signal and echo signal, respectively.

(3) The kernel function of parallel complex multiplication of reference signal complex conjugate matrix and echo signal matrix in the frequency domain is designed. Threads are allocated to compressed data based on range gate, and kernel function is called to complete parallel complex multiplication.

(4) Use the cuFFT library function to develop the IFFT solution to convert the range compressed data into the time domain, generate correlation values $F(\eta, \tau)$ that match the slant range.

(5) Repeat the steps (1)–(4) until all echo data are processed.

3.4.2. Parallelized Back Projection on GPU

In each imaged area, the corresponding time delay is calculated according to the slant range of each pixel, and a compensation factor is constructed to compensate for the phase of the pixel. Furthermore, both the sampling points processing at each portion and the pixel processing in the image are the same and independent. Therefore, all the pixels in the imaging area are implemented parallelly by an assigned thread, which was also used to calculate and process azimuth time images. In the viewpoint of thread mapping, there are two kinds of implementations for the parallel Back Projection step: pixel-oriented parallel and range code-oriented parallel. The two methods described above have good parallelism, the pixel-oriented parallelism is more suitable for the engineering realization of SAR image [30]. Thus, the pixel-oriented parallel BPA is adopted in this paper. The parallel implementation steps of pixel-oriented back-projection are as follows:

(1) Calculate the delay $\tau_{mn}(\eta)$ by constructing a kernel function to allocate threads to each pixel at a range code.
(2) According to the delay of each pixel $\tau_{mn}(\eta)$, the information in the corresponding range gate in the range compression is selected, and the compensation factor $h(\eta, m, n)$ is constructed according to the slant range information $R(\eta, m, n)$ and the phase difference value.
(3) Phase compensation is performed by complex multiplication of the range compression data $F(\eta, \tau_{mn})$ for that pixel point with the compensation factor $h(\eta, m, n)$ in the time domain. The kernel function will be invoked tens of thousands of times to complete the complex multiplication and complex addition processing.
(4) The data $S(\eta, x_m, y_n)$ from each azimuthal moment is coherently summed to generate the subimage $S_{sub}(\eta, x_m, y_n)$.
(5) Repeat steps (1)–(4) until all echo data are processed.

4. Results

To verify the acceleration of the proposed algorithm, running times and image quality were compared using a unified hardware development platform. In simulation and experiments, NVIDIA GeForce GTX 3090 is used, which is mounted in a computer equipped with AMD-3800X (Core frequency is 3.80~4.5 GHz), 64-bit CPU serving as the host, 64 GB RAM, and Windows 10 64-bit OS, whereas the C/C++ Visual Studio 2019 development environment is used to implement the algorithms. The specific operation process is implemented in the following environment:

(1) C++ as a reference compiled procedural language using the Faster Fourier Transform in the West (FFTW) library version 3.3.5.
(2) CUDA 10 with the cuFFT and NPP included libraries.

4.1. Simulation

Simulations were conducted to demonstrate the performance of the proposed algorithm, which were carried out using the CPU and GPU of the hardware platform described above, respectively. The geometric configuration of the satellite and receiver is shown in Figure 6a. Some parameters used in the simulation are listed in Table 1.

Figure 6b shows the multitarget simulation results of the proposed algorithm; all point targets are focused on the correct position in the image. Table 2 shows the elapsed time involved in the whole processing using CPU and GPU. The total running time of using the GPU and CPU to generate images of the size 4 km × 4 km is 106.468 s, while the running time of the improved BPA on the CPU is about 17,456.88 s, and the total speedup is about 163.96 times. FFT and IFFT process are accelerated by 117 times, the complex multiplication is accelerated by 261 times, and the back projection is accelerated by 164.1 times. Each step of the entire process of the BPA can be accelerated by GPU in parallel, and since most of the time of the algorithm is consumed in the back projection process, the overall acceleration ratio mainly depends on the back projection acceleration ratio.

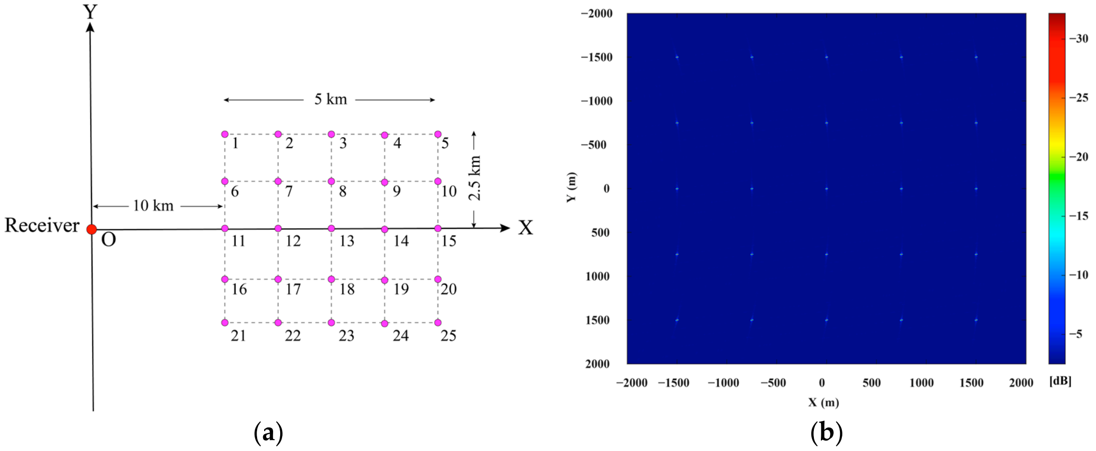


Figure 6. Layout of the point targets in simulation and the imaging results. (**a**) is the distribution of the simulation scene, and (**b**) is the imaging results with improved BPA on CPU and GPU.

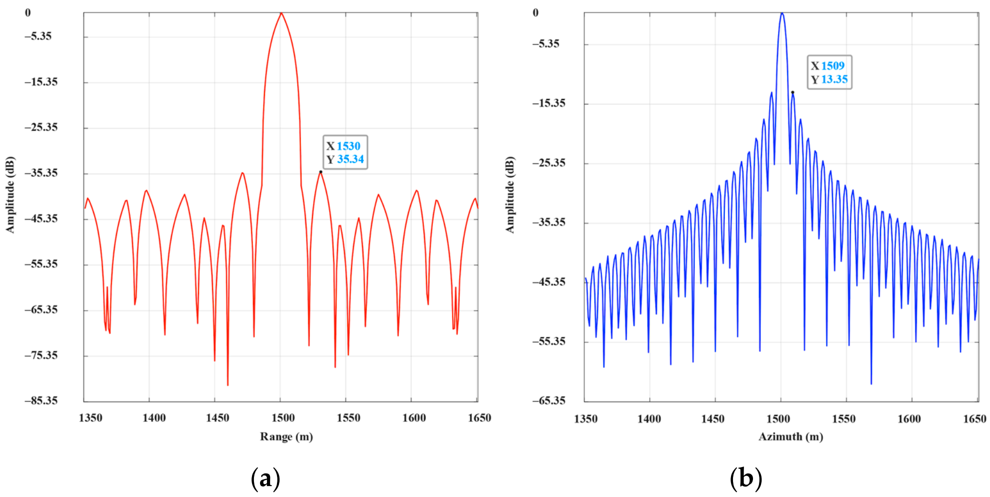
Table 1. Simulation parameters.

Parameters	Value
Satellite PRN	3
Carrier frequency	1176.45 MHz
Synthetic aperture time	300 s
Signal band width (GPS L5)	20.46 MHz
Sampling rate	62 MHz
Equivalent PRF	1000 Hz
Receiver position	(0, 0, 100) m
Satellite position at center time	(20,133.7258, 10,697.3032, 728.0291) km
Satellite speed at center time	(1392.7068, −2766.6856, 138.3063) m/s

Table 2. Comparison of processing times.

Parameters	Fft&Ifft	Complex Multiplication	Back Projection	Total Time
CPU(s)	74.4	20.88	17,361.6	17,456.88
GPU(s)	0.636	0.08	105.752	106.468
Real speed-up	117	261	164.17	163.96

To verify the imaging quality of the proposed algorithm, cross-sectional measurements were taken along the range and azimuth in the point targets. The imaging results of chosen target 25 are presented in Figure 7. As can be seen, Target 25 is well focused. The range profile and the azimuth profile were well consistent with theoretical ones [23]. Table 3 shows the comparison of the proposed algorithm imaging and CPU imaging of target 25, the range and azimuth imaging resolution, Peak Side-Lobe Ratio (PSLR), and Integral Side-Lobe Ratio (ISLR) are the same. This effectively shows that GPU-accelerated BPA can effectively reduce the imaging time without reducing the image quality at all. In the simulation, we ignored the cross-correlation interference caused by other navigation satellite signals. Firstly, the GPS-L5 signal's autocorrelation peak is about 35 dB higher than the sidelobe and cross-correlation results. Secondly, the strength of the autocorrelation signal will increase by $10\log_{10}{}^{N_r}$ times after azimuth focus, N^r is the number of azimuth samples, while the strength of the cross-correlation signal will not increase due to the different slant range of different satellites.



(**a**) (**b**)

Figure 7. Evaluated results for point target 25, (**a**) is range profile and (**b**) is azimuth profile.

Table 3. Comparison of imaging quality on the CPU and CPU and GPU.

	Range			Azimuth		
	Resolution/m	PSLR/dB	ISLR/dB	Resolution/m	PSLR/dB	ISLR/dB
CPU	15.8	−35.34	−13.425	8.65	−13.35	−10.551
GPU and CPU	15.8	−35.34	−13.425	8.65	−13.35	−10.551

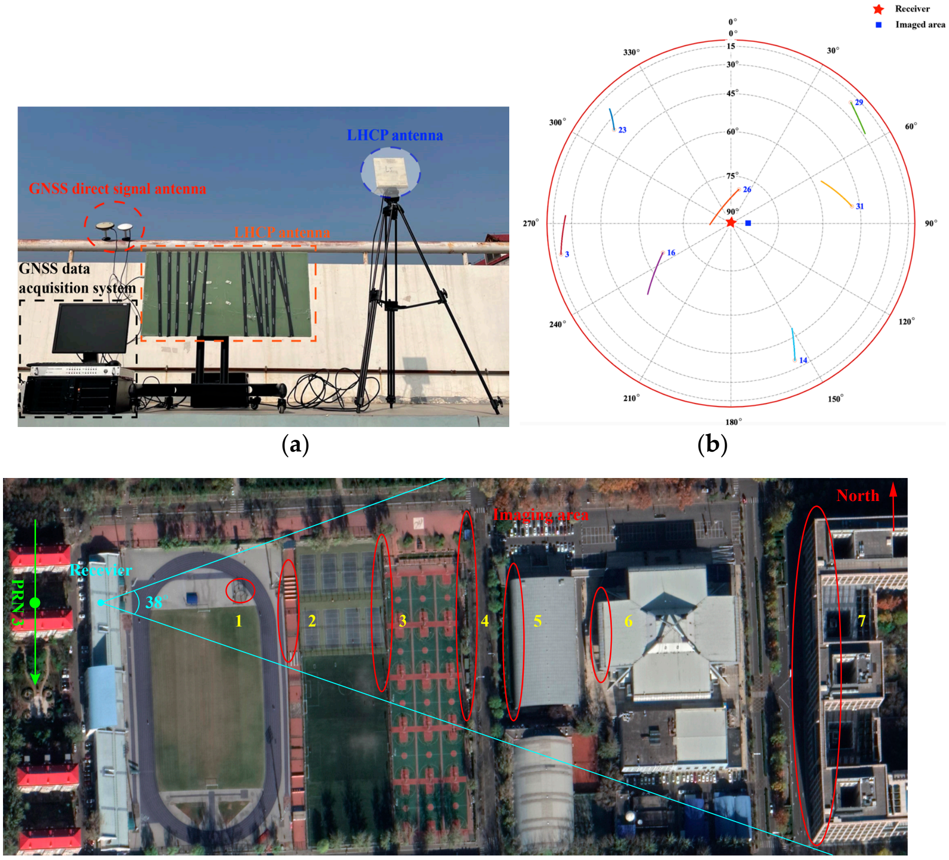
The simulation of scenes of different sizes is performed, and the results are shown in Table 4. In terms of computing efficiency, GPU and CPU-based methods are superior to CPU-based methods. The time consumed by the simulation increases as the size of the scene increases. When the scene size increases to a certain extent, the GPU stream processors are all in a working state, and the speedup ratio no longer changes. This illustrates that the acceleration capability of GPU-accelerated BPAs is largely dependent on the number of stream processors, with the higher the number of stream processors, the better the acceleration performance. To obtain better acceleration performance, developers can choose GPUs with more stream processors or use multiple GPU parallel methods to improve performance.

Table 4. Simulation time (s) comparison in generating different pixels.

Pixel	**128 × 128**	**256 × 256**	**512 × 512**	**1024 × 1024**	**2048 × 2048**	**4096 × 4096**	**8192 × 8192**
CPU(s)	123.1	206.4	507	2087.4	7611.9	30,543.3	68,871.9
GPU(s)	1.1	1.73	3.43	13.38	45.47	180	407.26
Speed-up	111.6	119.07	147.6	156	167.4	168.9	169.11

4.2. Experiments

To demonstrate the validity and feasibility of our proposed GPU accelerated improved BPA, real scene experiments were carried out. Figure 8 shows the real scene of the experiment. As shown in Figure 8a, the experimental hardware contains four navigation signal receiving channels. The RHCP omnidirectional antenna is used to receive direct signals from GNSS. The echo channel uses a high-gain LHCP antenna to receive the echo signal from the target area. The receiver location is the stadium stand of Beihang University, and the LHCP antenna was pointed toward the east, see Figure 8c. The experiment was conducted at 9:50 on 25 March 2021. According to the strategy of optimal resolution [28], GPS satellite PRN3 was selected as the optimal signal source, and a long-time synthetic aperture was carried out. The geometric configuration at the time of collection is shown in Figure 8b,c, and Table 5 lists the detailed parameters.



(**a**)

(**b**)

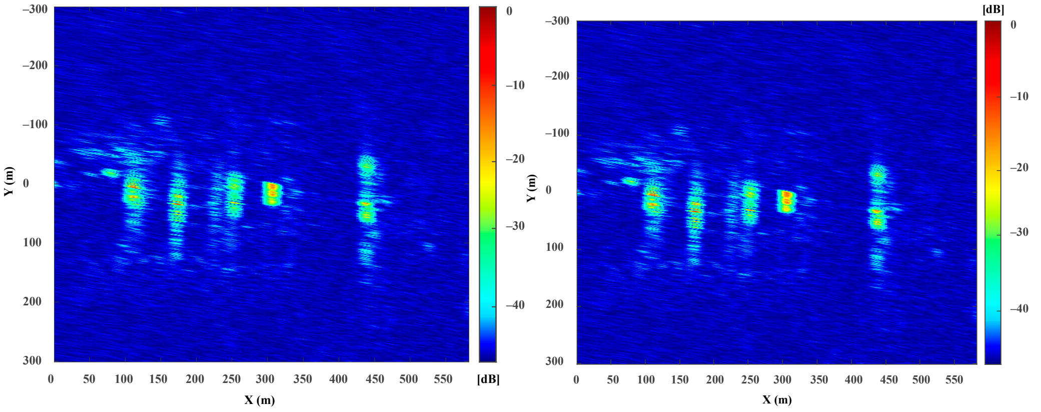
(**c**)

Figure 8. Experiment specific configuration. (**a**) Experimental equipment hardware, (**b**) geometric configuration at the time of collection, (**c**) optical image of experimental area (Google Earth map). Target No. 1 is a circular metal fence, Target No. 2~4 is metal fence, Target No. 5 is the swimming pool, Target No. 6 is the gymnasium, Target No. 7 is the E, F, G blocks of the new main building.

Table 5. Parameters used in the experiment.

Parameters	Value
GPS satellite	PRN3
Satellite angle—middle time (Elevation, Azimuth)	(28.59°, 270°)
Select signal	L5 (1176.45 MHz)
Select signal bandwidth	20.46 MHz
RHCP antenna gain	3 dBi
LHCP antenna gain/beamwidth	13 dBi/±19°
System sampling rate	62 MHz
Quantization bit	14 bit
Synthetic aperture time	1800 s

The imaging results are shown in Figure 9. It can be seen that Figure 9a,b were very similar to each other—this shows that the image quality obtained by using GPU and CPU to run BPA is the same, which is consistent with the theoretical results. Based on resolution analysis [23], the theoretically predicted resolution is approximately 16.8 m in the range and around 0.95 m in the azimuth. The time taken was 128.04 s to run BPA imaging on the GPU, while that of the CPU was 19,974.24 s. The ratio of time consumed by GPU slightly higher than the simulation result; this is caused by I/O data transmission. As the scene becomes larger or the synthetic aperture time increases, the advantages of the proposed algorithm will become more obvious.



(**a**) (**b**)

Figure 9. Comparison between imaging results, (**a**) is the imaging results of CPU, (**b**) is the imaging results of GPU.

It can be seen from Figure 9 that there are multiple strong scattering targets in the image, but we cannot interpret them. The low range resolution of the system causes this result. The interpretation is made easier by matching with a very high-resolution optical image. It can be seen from Figure 10 that the buildings in the optical image and the strong scattering area in the radar image have a good match. It can be seen from Figure 10 that the strong scattering area of the radar image is mainly concentrated in the west of the building, which is caused by the geometry of the acquisition time. As shown in Figure 8b, this geometry leads to the majority of the echo signal coming from the west side of the building. To obtain the top view image of the target area, the receiver should be installed at a higher position or on an airplane for data collection. The resulting image will contain more information about the target scene.

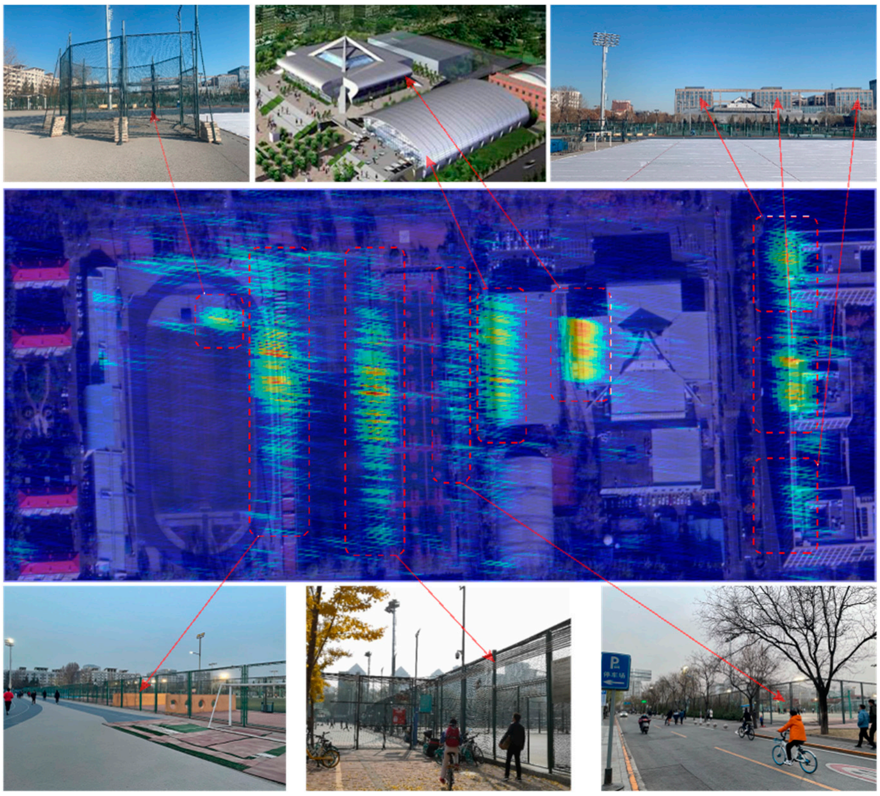
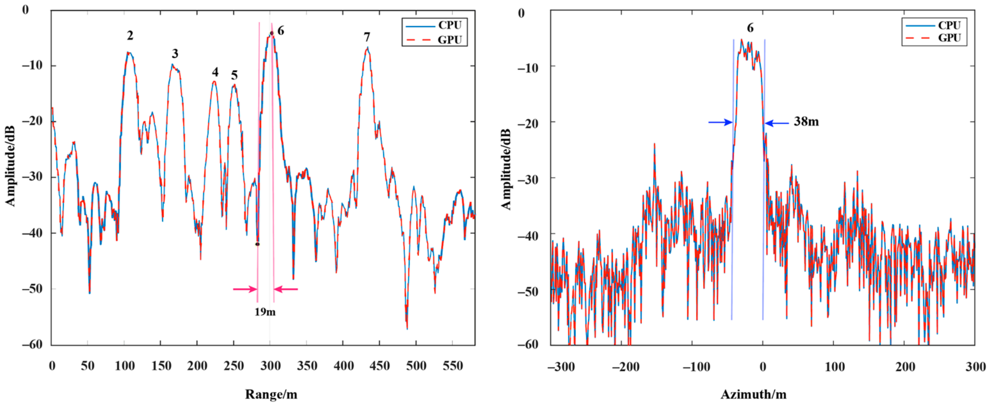


Figure 10. Comparison between the optical image and the resultant radar image.

To verify the image quality obtained, cross-sectional measurements were taken along the range and azimuth in the area around target No. 6 (the edge of the gymnasium), where the echoes have good signal strength and continuity. As shown in Figure 11a, the actual range resolution of target No. 6 is approximately 19 m, which is close to the theoretical value. The theoretically predicted azimuth resolution is approximately 0.95 m, which is much smaller than the length of the building. However, we can compare the length of the target's measured values to its physical length. As shown in Figure 11b, the total length of the target's measured values is about 38 m, which is consistent with the optical measurement result. In addition, Figure 11a,b show the cross-sectional measurement results of the images generated using GPU and CPU to run BPA, respectively. The contours generated by these two methods perfectly match each other, which shows that BPA can be executed in GPU without loss, and the calculation time is greatly reduced compared with the operation in CPU. The proposed method effectively applies BPA to GNSS-R BSAR, and its feasibility has been verified.



(**a**) (**b**)

Figure 11. Cross-sections for Target No. 6, (**a**) is range profile, and (**b**) is azimuth profile.

5. Discussion

In this section, the necessity of GPU accelerating BPA of GNSS-R BSAR is analysed. The computational cost of the BPA range compression step can be expressed as $N_aN_r\log_2 N_r + N_aN_r$, where N_a and N_r are the number of range and azimuth samples, respectively. The computational cost of the back-projection step can be expressed as $N_mN_nN_a$, where N_m and N_n are the numbers of samples of the imaging scene. Thus, the total computational cost for the improved BPA can be represented by $N_aN_r\log_2 N_r + N_aN_r + N_mN_nN_a$. The computational cost of an improved RDA is $3N_aN_r\log_2 N_r + 2N_aN_r\log_2 N_a + 7N_aN_r$ [27]. It can be seen that the calculation amount of BPA is much greater than RDA. However, BPA is more suitable for GNSS-R BSAR. The reason is as follows:

(1) The geometry of GNSS-R BSAR is complex. The BPA can focus the echo data without the influence of geometry, while the frequency domain algorithms are greatly affected by the geometric structure [28]. Especially in the multisatellite fusion or multistation fusion mode [19,23], more complex geometry will lead to most of the frequency domain algorithm, which is no longer applicable.

(2) The frequency domain algorithms rely on the accuracy of the equivalent squint model and cannot achieve a long-time synthetic aperture [41]. However, increasing the synthetic aperture time can not only improve the azimuth resolution and signal-to-noise ratio but also improve the characteristic information of the target in the imaged area [24]. Especially in one station fixed mode, the synthetic aperture time can reach thousands of seconds. The BPA algorithm is not affected by the synthetic aperture time.

In practical applications, whether airborne large scene imaging or one-station fixed mode long synthetic aperture time imaging, the BPA algorithm will achieve a huge amount of calculations, which will greatly affect the efficiency of BPA applications. Especially in the application of deformation monitoring, there is a demand for response time. Therefore, it is necessary to use GPU to accelerate BPA for reducing the imaged time.

In this paper, we proposed GPU accelerated BPA, which has a good acceleration ability in GNSS-R BSAR 's one-stop fixed mode. The simulation and experimental verification results verify the feasibility and effectiveness of the proposed algorithm. The simulated imaging is performed on an area of 4 km × 4 km, and the synthetic aperture time is 300 s. The point targets in the image generated by the proposed algorithm are well focused. Comparing the proposed algorithm with the BPA run only on CPU, the image quality is the same, while the running speed is increased by 163.96 times. In the experiment, the synthetic aperture time was increased to the 1800 s for an image area of 600 m × 600 m. The collected raw data were processed by the proposed algorithm and the BPA run only on the CPU, respectively. The imaging results are similar for the two algorithms, the running time of the proposed algorithm is 128.04 s, while the running time on the CPU is 19,974.24 s; the time taken by the proposed algorithm is greatly reduced.

6. Conclusions

With the improvement of GPU computing power, the BPA has become a feasible imaging method for processing GNSS-R BSAR data. This paper proposes an improved BPA for processing GNSS-R BSAR data using a heterogeneous architecture platform of CPU and GPU. The improved BPA can accurately compensate for the nonlinear motion error of the satellite and improve the synchronization performance of the direct signal and echo signal. The improved BPA on the CPU and GPU platform greatly reduces the total running time and can meet the needs of many offline GNSS-R BSAR imaging applications. As shown by the simulation and the experimental results, the algorithm performs well in imaging quality and efficiency. Using GPS L5 signal for a long-time synthetic aperture, the range resolution of the generated image is about 19 m, and the azimuth resolution is about submeter level. Images of this quality can meet the needs of many applications; an important application of GNSS-R BSAR imaging in one-station fixed mode is deformation monitoring. The next step is to apply this algorithm to the research of deformation monitoring, which will greatly reduce time consumption and improve emergency response time.

Author Contributions: Conceptualization, D.Y. and F.W.; methodology, S.W.; software, S.W.; validation, S.W., F.W.; formal analysis, S.W.; investigation, S.W., G.G. and Z.X.; resources, S.W.; data curation, S.W.; writing—original draft preparation, S.W.; writing—review and editing, S.W. and F.W.; visualization, S.W., G.G. and Z.X.; supervision, D.Y.; project administration, D.Y.; funding acquisition, D.Y and F.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (NNSFC) under grant no. 41774028, China Postdoctoral Innovation Talent Support Program (BX20200039), Beihang University Beidou technology achievement transformation and industrialization fund support (BARI2003).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The processed data cannot be shared at this time because they are also part of an ongoing research project.

Acknowledgments: The authors acknowledge no more support.

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

References

1. Hall, C.D.; Cordey, R.A. Multistatic Scatterometry. In Proceedings of the International Geoscience and Remote Sensing Symposium, Edinburgh, UK, 12–16 September 1988; pp. 561–562.
2. Martin-neira, M. A Passive Reflectometry and Interferometry System (PARIS): Application to ocean altimetry. *ESA J.* **1993**, *17*, 331–355.
3. Foti, G.; Gommenginger, C.; Jales, P.; Unwin, M.; Shaw, A.; Robertson, C.; Rosello, J. Spaceborne GNSS-Reflectometry for ocean winds: First results from the UK TechDemoSat-1 mission: Spaceborne GNSS-R: First TDS-1 results. *Geophys. Res. Lett.* **2015**, *42*. [CrossRef]
4. Clarizia, M.P.; Ruf, C.S.; Jales, P.; Gommenginger, C. Spaceborne GNSS-R Minimum Variance Wind Speed Estimator. *IEEE Geosci. Remote Sens.* **2014**, *52*, 6829–6843. [CrossRef]
5. Wang, F.; Yang, D.; Yang, L. Feasibility of Wind Direction Observation Using Low-Altitude Global Navigation Satellite System-Reflectometry. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 5063–5075. [CrossRef]
6. Li, W.; Cardellach, E.; Fabra, F.; Ribó, S.; Rius, A. Effects of PRN-Dependent ACF Deviations on GNSS-R Wind Speed Retrieval. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 327–331. [CrossRef]
7. Rius, A.; Nogués-Correig, O.; Ribó, S.; Cardellach, E.; Oliveras, S.; Valencia, E.; Park, H.; Tarongí, J.M.; Camps, A.; van der Marel, H.; et al. Altimetry with GNSS-R interferometry: First proof of concept experiment. *GPS Solut.* **2012**, *16*, 231–241. [CrossRef]
8. Cardellach, E.; Rius, A.; Martín-Neira, M.; Fabra, F.; Nogués-Correig, O.; Ribó, S.; Kainulainen, J.; Camps, A.; Addio, S.D. Consolidating the Precision of Interferometric GNSS-R Ocean Altimetry Using Airborne Experimental Data. *IEEE Geosci. Remote Sens.* **2014**, *52*, 4992–5004. [CrossRef]
9. Li, W.; Yang, D.; Addio, S.D.; Martín-Neira, M. Partial Interferometric Processing of Reflected GNSS Signals for Ocean Altimetry. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 1509–1513. [CrossRef]
10. Lowe, S.T.; LaBrecque, J.L.; Zuffada, C.; Romans, L.J.; Young, L.E.; Hajj, G.A. First spaceborne observation of an Earth-reflected GPS signal. *Radio Sci.* **2002**, *37*, 1–28. [CrossRef]
11. Rodriguez-Alvarez, N.; Camps, A.; Vall-llossera, M.; Bosch-Lluis, X.; Monerris, A.; Ramos-Perez, I.; Valencia, E.; Marchan-Hernandez, J.F.; Martinez-Fernandez, J.; Baroncini-Turricchia, G.; et al. Land Geophysical Parameters Retrieval Using the Interference Pattern GNSS-R Technique. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 71–84. [CrossRef]
12. Rodriguez Alvarez, N.; Bosch, X.; Camps, A.; Vall-llossera, M.; Valencia, E.; Marchan, J.; Ramos-Perez, I. Soil Moisture Retrieval Using GNSS-R Techniques: Experimental Results over a Bare Soil Field. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 3616–3624. [CrossRef]
13. Egido, A.; Paloscia, S.; Motte, E.; Guerriero, L.; Pierdicca, N.; Caparrini, M.; Santi, E.; Fontanelli, G.; Floury, N. Airborne GNSS-R Polarimetric Measurements for Soil Moisture and Above-Ground Biomass Estimation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 1522–1532. [CrossRef]
14. Ferrazzoli, P.; Guerriero, L.; Pierdicca, N.; Rahmoune, R. Forest biomass monitoring with GNSS-R: Theoretical simulations. *Adv. Space Res.* **2011**, *47*, 1823–1832. [CrossRef]
15. Wu, X.; Ma, W.; Xia, J.; Bai, W.; Jin, S.; Calabia, A. Spaceborne GNSS-R Soil Moisture Retrieval: Status, Development Opportunities, and Challenges. *Remote Sens.* **2021**, *13*, 45. [CrossRef]
16. Carreno-Luengo, H.; Luzi, G.; Crosetto, M. Above-Ground Biomass Retrieval over Tropical Forests: A Novel GNSS-R Approach with CyGNSS. *Remote Sens.* **2020**, *12*, 1368. [CrossRef]
17. Li, C.; Huang, W. Sea surface oil slick detection from GNSS-R Delay-Doppler Maps using the spatial integration approach. In Proceedings of the 2013 IEEE Radar Conference (RadarCon13), Ottawa, ON, Canada, 29 April–3 May 2013; pp. 1–6.
18. Gao, H.; Yang, D.; Wang, F.; Wang, Q.; Li, X. Retrieval of Ocean Wind Speed Using Airborne Reflected GNSS Signals. *IEEE Access* **2019**, *7*, 71986–71998. [CrossRef]
19. Antoniou, M.; Cherniakov, M. GNSS-based bistatic SAR: A signal processing view. *EURASIP J. Adv. Signal Process.* **2013**, *2013*. [CrossRef]
20. Zhou, X.; Wang, P.; Chen, J.; Men, Z.; Liu, W.; Zeng, H. A Modified Radon Fourier Transform for GNSS-Based Bistatic Radar Target Detection. *IEEE Geosci. Remote Sens. Lett.* **2020**, 1–5. [CrossRef]
21. Liu, F.; Fan, X.; Zhang, T.; Liu, Q. GNSS-Based SAR Interferometry for 3-D Deformation Retrieval: Algorithms and Feasibility Study. *IEEE Geosci. Remote Sens.* **2018**, *56*, 5736–5748. [CrossRef]
22. Liu, F.; Antoniou, M.; Zeng, Z.; Cherniakov, M. Coherent Change Detection Using Passive GNSS-Based BSAR: Experimental Proof of Concept. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 4544–4555. [CrossRef]
23. Wu, S.; Yang, D.; Zhu, Y.; Wang, F. Improved GNSS-Based Bistatic SAR Using Multi-Satellites Fusion: Analysis and Experimental Demonstration. *Sensors* **2020**, *20*, 7119. [CrossRef]
24. Liu, F.; Antoniou, M.; Zeng, Z.; Cherniakov, M. Point Spread Function Analysis for BSAR With GNSS Transmitters and Long Dwell Times: Theory and Experimental Confirmation. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 781–785. [CrossRef]
25. Antoniou, M.; Saini, R.; Cherniakov, M. Results of a Space-Surface Bistatic SAR Image Formation Algorithm. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 3359–3371. [CrossRef]
26. Zeng, T.; Wang, R.; Li, F.; Long, T. A Modified Nonlinear Chirp Scaling Algorithm for Spaceborne/Stationary Bistatic SAR Based on Series Reversion. *IEEE Geosci. Remote Sens.* **2013**, *51*, 3108–3118. [CrossRef]

27. Zhou, X.-K.; Chen, J.; Wang, P.-b.; Zeng, H.-C.; Fang, Y.; Men, Z.-R.; Liu, W. An Efficient Imaging Algorithm for GNSS-R Bi-Static SAR. *Remote Sens.* **2019**, *11*, 2945. [CrossRef]
28. Moccia, A.; Renga, A. Spatial Resolution of Bistatic Synthetic Aperture Radar: Impact of Acquisition Geometry on Imaging Performance. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 3487–3503. [CrossRef]
29. Shao, Y.F.; Wang, R.; Deng, Y.K.; Liu, Y.; Chen, R.; Liu, G.; Loffeld, O. Fast Backprojection Algorithm for Bistatic SAR Imaging. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 1080–1084. [CrossRef]
30. Jun, S.; Long, M.; Xiaoling, Z. Streaming BP for Non-Linear Motion Compensation SAR Imaging Based on GPU. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *6*, 2035–2050. [CrossRef]
31. Zhang, F.; Yao, X.; Tang, H.; Yin, Q.; Hu, Y.; Lei, B. Multiple Mode SAR Raw Data Simulation and Parallel Acceleration for Gaofen-3 Mission. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 2115–2126. [CrossRef]
32. Zhang, F.; Hu, C.; Li, W.; Hu, W.; Li, H. Accelerating Time-Domain SAR Raw Data Simulation for Large Areas Using Multi-GPUs. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 3956–3966. [CrossRef]
33. Frey, O.; Werner, C.L.; Wegmuller, U. GPU-based parallelized time-domain back-projection processing for Agile SAR platforms. In Proceedings of the 2014 IEEE Geoscience and Remote Sensing Symposium, Quebec, QC, Canada, 13–18 July 2014; pp. 1132–1135.
34. Fasih, A.; Hartley, T. GPU-accelerated synthetic aperture radar backprojection in CUDA. In Proceedings of the 2010 IEEE Radar Conference, Arlington, VA, USA, 10–14 May 2010; pp. 1408–1413.
35. Que, R.; Ponce, O.; Scheiber, R.; Reigber, A. Real-time processing of SAR images for linear and non-linear tracks. In Proceedings of the 2016 17th International Radar Symposium (IRS), Krakow, Poland, 10–12 May 2016; pp. 1–4.
36. Zeng, H.-C.; Wang, P.-B.; Chen, J.; Liu, W.; Ge, L.; Yang, W. A Novel General Imaging Formation Algorithm for GNSS-Based Bistatic SAR. *Sensors* **2016**, *16*, 294. [CrossRef] [PubMed]
37. Yegulalp, A.F. Fast backprojection algorithm for synthetic aperture radar. In Proceedings of the 1999 IEEE Radar Conference, Waltham, MA, USA, 22–22 April 1999; pp. 60–65.
38. Antoniou, M.; Cherniakov, M. Pre-processing for time domain image formation in SS-BSAR system. *J. Syst. Eng. Electron.* **2012**, *23*, 875–880.
39. Roland, E.B. *Phase-Locked Loops: Design, Simulation, and Applications*, 6th ed.; McGraw-Hill Education: New York, NY, USA, 2007.
40. National Aeronautics and Space Administration. Available online: https://cddis.nasa.gov (accessed on 25 March 2021).
41. Wang, P.; Liu, W.; Chen, J.; Niu, M.; Yang, W. A High-Order Imaging Algorithm for High-Resolution Spaceborne SAR Based on a Modified Equivalent Squint Range Model. *IEEE Geosci. Remote Sens.* **2015**, *53*, 1225–1235. [CrossRef]