

Article

GPU-Based Soil Parameter Parallel Inversion for PolSAR Data

Qiang Yin , You Wu, Fan Zhang *  and Yongsheng Zhou 

College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China; yinq@mail.buct.edu.cn (Q.Y.); youwu@mail.buct.edu.cn (Y.W.); zhyosh@mail.buct.edu.cn (Y.Z.)

* Correspondence: zhangf@mail.buct.edu.cn

Received: 19 December 2019; Accepted: 23 January 2020; Published: 28 January 2020



Abstract: With the development of polarimetric synthetic aperture radar (PolSAR), quantitative parameter inversion has been seen great progress, especially in the field of soil parameter inversion, which has achieved good results for applications. However, PolSAR data is also often many terabytes large. This huge amount of data also directly affects the efficiency of the inversion. Therefore, the efficiency of soil moisture and roughness inversion has become a problem in the application of this PolSAR technique. A parallel realization based on a graphics processing unit (GPU) for multiple inversion models of PolSAR data is proposed in this paper. This method utilizes the high-performance parallel computing capability of a GPU to optimize the realization of the surface inversion models for polarimetric SAR data. Three classical forward scattering models and their corresponding inversion algorithms are analyzed. They are different in terms of polarimetric data requirements, application situation, as well as inversion performance. Specifically, the inversion process of PolSAR data is mainly improved by the use of the high concurrent threads of GPU. According to the inversion process, various optimization strategies are applied, such as the parallel task allocation, and optimizations of instruction level, data storage, data transmission between CPU and GPU. The advantages of a GPU in processing computationally-intensive data are shown in the data experiments, where the efficiency of soil roughness and moisture inversion is increased by one or two orders of magnitude.

Keywords: GPU; parallel computing; PolSAR; roughness; soil moisture

1. Introduction

Soil moisture and roughness are important parameters in the fields of agriculture, ecology, meteorology, and hydrology. They are widely used in farmland irrigation management, climate prediction, and drought monitoring [1]. For example, in the agricultural area, soil water content and roughness directly affect crop growth [2]. The correct assessment of soil moisture is also the basis of hydrological modeling [3]. In the meteorological field, soil water content is an essential component of the land–atmosphere boundary energy budget [4,5]. Therefore, the inversion of soil water content and roughness has become a research hotspot for scholars.

With the increase in the number of global satellites, the application of ground exploration has become increasingly common [6–8]. But low resolution is also a major problem for exploration [9,10]. The development of science and technology has promoted the rapid development of synthetic aperture radar (SAR) techniques, through which the quality and resolution of radar imaging has been significantly improved. However, in the process of ground detection, the anti-interference of SAR is very low, and the detection process is easily affected. Both GIS and remote sensing assistance information are used for soil moisture estimation [11]. Multi-satellite collaboration can also improve spatio-temporal resolution [12]. By means of dual/multi/full polarization, the polarized synthetic aperture radar (PolSAR) has a good detection effect and high resolution [13]. PolSAR plays an important role in geographic surveying,

geological hazard monitoring, vegetation monitoring, and other applications [14]. The applications of PolSAR data can be generally divided into two categories, which are qualitative and quantitative ones. For qualitative applications, the PolSAR technique has a great advantage in unsupervised classification due to the inherent scattering mechanisms contained in PolSAR data. This kind of physical scattering information can be directly used for land cover classification which need no training.

For quantitative applications, PolSAR has a profound impact on the study of parameter inversion. Here PolSAR provides the relations between observations of multi-polarimetric channels with system parameters and object parameters, which is more stable than single polarimetric channel observation. Soil parameter estimation is a representative quantitative application of PolSAR data, since it covers both geometrical and physical parameters. In the study of bare soil parameters, scholars at home and abroad have developed a variety of soil parameter inversion models. These models are broadly divided into theoretical scattering models and empirical scattering models. The theoretical models include physical optics (PO) models, geometrical optics (GO) models, and integral equation methods (IEM). The theoretical models are based on the assumptions that the naturally exposed surface is a uniform half-space dielectric layer, so the accuracy of these models is limited [9,14,15]. Some scholars have studied the accuracy of the inversion model [16–18]. However, its complicated form makes it very difficult to obtain the roughness and water content parameters directly from the polarization imagery. Therefore, combining theoretical model analysis and polarization data sets, establishing an empirical relationship between various echo parameters and surface parameters has become the main way to obtain surface parameters [19,20]. The empirical and semi-empirical models include small perturbation methods (SPM), Dubois, Oh, and other models. By comparing the calculated result with the actual external measurement, then with the adjustment of the parameters, the soil parameters calculated by the model are more in line with the actual situation [21]. In order to obtain more accurate results, the X-Bragg model based on full polarization is also proposed. The model is based on the eigenvalues and eigenvectors of the polarization coherence matrix. However, the high-resolution PolSAR massive imagery becomes the bottleneck of computing efficiency.

The efficient processing speed of soil moisture retrieval can help make timely decisions in the real-time application of geological exploration [4]. In recent years, the development of high-performance technologies has solved the computing-intensive problem, especially the GPU parallel method, e.g., hybrid OpenMP-CUDA based PDE source inversion [22], multiplication regular comparison source (MR-CSI) graphic processing unit (GPU) parallel optimization [23], GPU 2D and 3D multi-frequency regularization comparison source [24], parallel optimization of multi-scale MAS systems, and GPU-based accelerated TMI [25]. Parallel computing research of large-scale grounded grid PC cluster is realized [26]. In the heterogeneous soil model, OpenMP parallel optimization is used for multi-core parallelism implementation [27]. In our previous work, various parallel mechanisms have been introduced to accelerate the SAR raw data simulation, including clouding computing, GPU parallel, CPU parallel, and hybrid CPU/GPU parallel [28–35]. As far as the inversion algorithms are concerned, the time cost is only minute-level. Compared with the hybrid CPU/GPU parallel accelerating, the GPU parallel is expected to be a better choice for balancing the algorithm complexity and efficiency. Therefore, single GPU has been employed to implement the massive parallel parameter inversion for PolSAR imagery.

This paper studies the Dubois, Oh, and X-Bragg model inversion algorithms, which basically covers all widely used empirical models. Among them, the Dubois and Oh models use scattering coefficients of two or three polarimetric channels, which is only the amplitude information, while the X-Bragg model utilizes the full polarimetric scattering matrix including both amplitude and phase information. In fact, the higher data requirement of models has, the better performance it achieves. When the full polarimetric scattering matrix is available, Oh and X-Bragg models can be used. However, for the surface with vegetation and non-unneglectable slope, X-Bragg model should be chosen with priority. When we have amplitude data of polarimetric channels, Dubois and Oh models could be employed. In the case that only dual polarimetric data is available (HH,VV), then the Dubois model is left. It should be noted that with the Dubois model, good results are obtained under the condition that the incidence

angle is larger than 30 degrees. The scattering models themselves are a forward model, that is, from the input object parameters and observation parameters to the scattering matrix or coefficients. However, if we want to retrieve the soil roughness and moisture from the data, the corresponding inversion algorithms are needed. We mainly analyze the optimization of the three algorithms for soil inversion based on the GPU parallel method. Therefore, the contributions of this paper are mainly reflected in the parallel design (GPU thread allocation strategy), parallel optimization for computing-intensive issue (instruction optimization), and parallel optimization for data-intensive issue (storage optimization and data type conversion). Through the three aspects of parallel acceleration, $14 \times -169 \times$ speedups can be achieved for the three inversion models.

The rest of this paper is organized as follows. In Section 2, the three inversion algorithms are specifically introduced, including the Dubois model, Oh model, and X-Bragg model. In Section 3, the sequential algorithm analysis and the proposed parallel methods are presented. In Section 4, the experimental results and analysis are presented. The conclusion is given in the final section.

2. Inversion Algorithm

2.1. Dubois Model

In 1995, Dubois proposed an empirical model that only requires the same polarization backscatter coefficients σ_{HH}^0 and σ_{VV}^0 to extract the root mean square height and water content of the bare soil. The model was built using the datasets collected by a truck-mounted scatterometer at the University of Michigan and the RASAM scatterometer at the University of Bern. Through the measurement of the scatterometer and the data, the local incident angle and frequency, the dielectric constant and the surface roughness are mapped to the co-polarized scattering coefficient. Studies have found that this relationship is close to the tangent of the angle of incidence. The algorithm is applied to SAR data (AIRSAR and SIR-C) to prove its robustness [36].

The empirical formula is as follows:

$$\sigma_{HH}^0 = 10^{-2.75} * \frac{\cos^{1.5}\theta}{\sin^5\theta} * 10^{0.028*\epsilon_r*\tan\theta} (ks * \sin\theta)^{1.4} * \lambda^{0.7} \quad (1)$$

$$\sigma_{VV}^0 = 10^{-2.37} * \frac{\cos^3\theta}{\sin^3\theta} * 10^{0.046*\epsilon_r*\tan\theta} (ks * \sin\theta)^{1.1} * \lambda^{0.7} \quad (2)$$

where θ is the local incidence angle, ϵ_r is the real part of the dielectric constant, ks the normalized surface roughness and λ the wavelength.

The volume water content of soil m_v can be calculated from the relationship between ϵ_r and m_v :

$$m_v = 4.3 * 10^{-6} \epsilon_r^3 - 5.5 * 10^{-4} \epsilon_r^2 + 2.92 * 10^{-2} \epsilon_r - 5.3 * 10^{-2} \quad (3)$$

The effective range of the inversion model for estimating surface parameters is $m_v \leq 35\%$, $ks \leq 2.5$ and $\theta \geq 30^\circ$.

2.2. Oh Model

At the University of Michigan, based on the analysis of the classical theoretical scattering model Kirchhoff approximation (KA) and SPM, Y. Oh, K.Sarabandi, and F.T. Ulaby developed this semi-empirical model in 1992. The model uses full-polarization data (LCX POLARSCAT) measured by an on-board network analysis scatterometer at three frequencies (1.5, 4.5, and 9.5 GHz), as well as comprehensive and accurate surface measurements, with incident angles ranging from 10° to 70° [36,37].

The model proposes a clear cross-polarization and co-polarization backscatter ratio function. The empirical equation is:

$$P = \frac{\sigma_{HV}^0}{\sigma_{VV}^0} = 0.23\sqrt{\Gamma^0}(1 - e^{-ks}) \quad (4)$$

$$Q = \frac{\sigma_{HH}^0}{\sigma_{VV}^0} = \left(1 - \left(\frac{2\theta}{\pi}\right)^{\frac{1}{3\Gamma^0}} * e^{-ks}\right)^2 \quad (5)$$

where P and Q represent the cross-polarization and co-polarization backscatter ratios (i.e., $\frac{\sigma_{HV}^0}{\sigma_{VV}^0}$ and $\frac{\sigma_{HH}^0}{\sigma_{VV}^0}$), θ is the local incident angle, and ks is the root mean square height (i.e., roughness) after the wavelength is normalized, and Γ^0 is the Fresnel reflection coefficient.

$$\Gamma^0 = \left| \frac{1 - \sqrt{\epsilon_r}}{1 + \sqrt{\epsilon_r}} \right|. \quad (6)$$

By combining Equations (4)–(6) we can obtain the mathematical Equation (7):

$$x^n = \frac{a^{\frac{(x_{n-1})^2}{3}}(1 - bx_{n-1}) + c}{\left[\frac{2x_{n-1}}{3} \ln(a)(1 - bx_{n-1}) - b \right] a^{\frac{(x_{n-1})^2}{3}}}. \quad (7)$$

Among them, $x = \frac{1}{\Gamma^0}$, $b = \frac{q}{0.23}$, $a = \frac{2\theta}{\pi}$ and $c = \sqrt{p} - 1$. x is obtained by the iterative method in the program, and then the Fresnel reflectivity Γ^0 and Fresnel reflection coefficient ϵ_r can be obtained, as well as the soil roughness (ks) and soil moisture (m_v). In general, the model shows good agreement on ground measurements within a certain range, where $ks \in [0.1, 6]$, $m_v \in [9, 31]$.

2.3. X-Bragg Model

X-Bragg model is an SPM-based polarimetric scattering model. It utilizes the coherency matrix of full polarimetric data, including the phase information. Firstly, from the side of polarimetric coherency matrix, which contains the second order moment of scattering process shown in Equation (8), can be diagonalized by an unitary similarity transformation of the following form [38]:

$$[T] = [U_3][\Lambda][U_3]^{-1} \quad (8)$$

where

$$[\Lambda] = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}, \quad [U_3] = [\vec{e}_1, \vec{e}_2, \vec{e}_3] \quad (9)$$

$[\Lambda]$ is a diagonal matrix whose elements are $[T]$ real non-negative eigenvalues $0 \leq \lambda_1 \leq \lambda_2 \leq \lambda_3$; $[U_3]$ is an eigenvector matrix whose columns correspond to orthogonal eigenvectors \vec{e}_1 , \vec{e}_2 and \vec{e}_3 . In this way the coherency matrix T is written as

$$[T] = [U_3][\Lambda][U_3]^{-1} = \lambda_1(\vec{e}_1 \cdot \vec{e}_1^+) + \lambda_2(\vec{e}_2 \cdot \vec{e}_2^+) + \lambda_3(\vec{e}_3 \cdot \vec{e}_3^+). \quad (10)$$

The diagonalization of the coherency matrix directly produces three important physical features. Firstly with the obtained eigenvalues, the scattering probability p_i are computed by normalizing the eigenvalues.

$$p_i = \frac{\lambda_i}{\lambda_1 + \lambda_2 + \lambda_3} \quad (11)$$

Then two of the physical features are defined as follows, which are polarization scattering entropy H and scattering anisotropy A

$$H = - \sum_{i=1}^3 p_i \log_3 p_i, \quad A = \frac{p_2 - p_3}{p_2 + p_3}. \quad (12)$$

The third important parameter is obtained from the eigenvector of $[T]$. Each feature vector \vec{e}_i can be represented by five angles [31]. The β_i angle can be interpreted as the rotation of the corresponding feature vector \vec{e}_i in a plane perpendicular to the scattering plane, while φ_{1i} , φ_{2i} , and φ_{3i} explain the phase relationship between the \vec{e}_i elements. In this work, the average scattering angle α is more important, which is defined as

$$\vec{e}_i = \begin{bmatrix} \cos\alpha_i \exp(i\varphi_{1i}) & \sin\alpha_i \cos\beta_i \exp(i\varphi_{2i}) & \sin\alpha_i \sin\beta_i \exp(i\varphi_{3i}) \end{bmatrix} \quad (13)$$

$$\alpha = p_1\alpha_1 + p_2\alpha_2 + p_3\alpha_3. \quad (14)$$

To extend the Bragg scattering model to a wider range of roughness conditions, the Bragg coherency matrix $[T]$ is rotated around a plane perpendicular to the scattering plane. The rough surface is modeled as a reflective symmetry depolarizer, as shown in Equation (15). A configuration averaging is performed on a given distribution β of $P(\beta)$:

$$[T(\beta)] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 2\beta & -\sin 2\beta \\ 0 & \sin 2\beta & \cos 2\beta \end{bmatrix} \begin{bmatrix} \langle |\sigma_{HH}^0 + \sigma_{VV}^0|^2 \rangle & \langle (\sigma_{HH}^0 - \sigma_{VV}^0)(\sigma_{HH}^0 + \sigma_{VV}^0)^* \rangle & 0 \\ \langle (\sigma_{HH}^0 + \sigma_{VV}^0)(\sigma_{HH}^0 - \sigma_{VV}^0)^* \rangle & \langle |\sigma_{HH}^0 - \sigma_{VV}^0|^2 \rangle & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 2\beta & \sin 2\beta \\ 0 & -\sin 2\beta & \cos 2\beta \end{bmatrix} \quad (15)$$

$$[T] = \int_0^{2\pi} [T(\beta)]P(\beta)d\beta. \quad (16)$$

Indeed, Figure 1 shows the corresponding spatial relationship of the surface slope in detail.

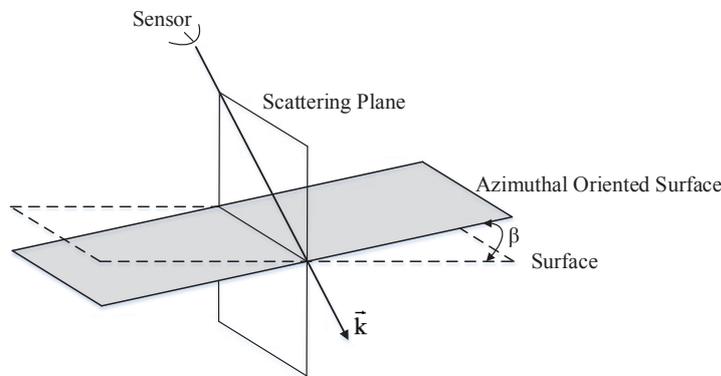


Figure 1. Surface slope diagram.

The width of the assumed distribution corresponds to the amount of roughness disturbance of the modeled surface [38]. Assuming $P(\beta)$ to be a uniform distribution about zero with width β_1 :

$$P(\beta) = \begin{cases} \frac{1}{2\beta_1} & |\beta| \leq \beta_1 \\ 0 & 0 \leq \beta_1 \leq \frac{\pi}{2} \end{cases} \quad (17)$$

The coherency matrix for the rough surface becomes:

$$[T] = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix} = \begin{bmatrix} C_1 & C_2 \text{sinc}(2\beta_1) & 0 \\ C_2 \text{sinc}(2\beta_1) & C_3(1 + \text{sinc}(4\beta_1)) & 0 \\ 0 & 0 & C_3(1 - \text{sinc}(4\beta_1)) \end{bmatrix} \quad (18)$$

the coefficients C_1 , C_2 and C_3 describing the Bragg components of the surface are given by

$$C_1 = \sigma_{HH}^0 + (\sigma_{VV}^0)^2, \quad C_2 = (\sigma_{HH}^0 + \sigma_{VV}^0)((\sigma_{HH}^0)^* - (\sigma_{VV}^0)^*), \quad C_3 = \left| \sigma_{HH}^0 - \sigma_{VV}^0 \right|^2 / 2. \quad (19)$$

For the soil roughness estimation, ks can be calculated by Equation (20)

$$ks = 1 - A. \quad (20)$$

With the obtained roughness ks , the corresponding entropy H and α angle values are stored in the look-up-table (LUT) by Equations (12) and (14). Using this LUT, the dielectric constant value can be obtained directly from the estimated entropy H and α angle values. Thus, the corresponding moisture mv is obtained.

3. Proposed Parallel Inversion Methods

3.1. Inversion Algorithms Analysis

These three inversion algorithms based on Dubois, Oh, and X-Bragg scattering algorithms differ in the aspects of input data, valid ranges, features, and computation complexity, as do the parallel processing methods applied to them, as shown in Figures 2 and 3.

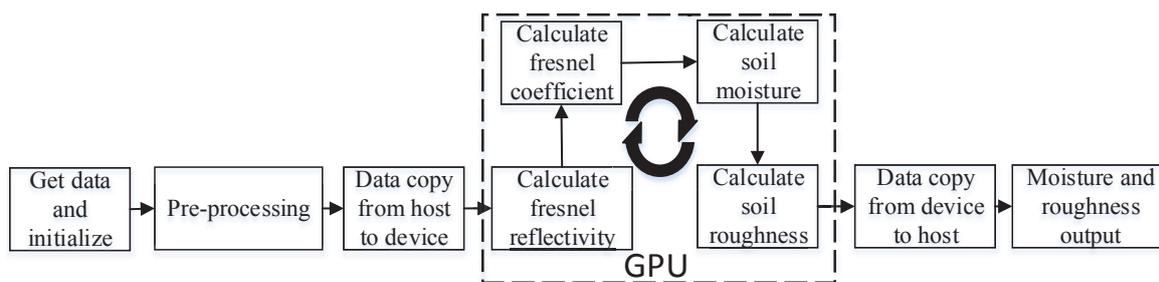


Figure 2. Dubois and Oh inversion algorithm optimization framework.

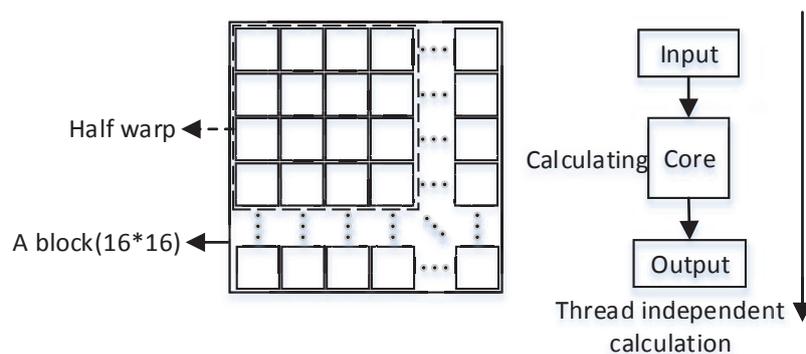


Figure 3. Thread allocation strategy.

For the inversion of the Dubois algorithm, it is straight and simple from the algorithm equations. At first, the dielectric constant is computed then the surface roughness is calculated. It requires only the scattering coefficients of HH and VV channels, hence they could be applied widely in the presence of the dual pol data availability of many airborne and spaceborne platform. However, it should be noticed that only when the incidence angle is larger than 30 degrees, the algorithm has reliable inversion results. According to Equations (1)–(3), the algorithm complexity is calculated as $O(n)$, where n indicates the number of PolSAR image pixels. Although the algorithm complexity is ordinary, there are many

time-consuming functions including trigonometric and exponential functions, which may reduce the acceleration efficiency.

The Oh algorithm utilizes the full polarimetric scattering coefficients. While for inversion, the Fresnel coefficient is first obtained by an iterative process, following that, the dielectric constant and roughness are computed consequently. Oh has a large valid range of roughness and moisture among the empirical inversion models. When the amplitudes of full polarimetric SAR data are available, it can be applied. According to Equation (4)–(7), its algorithm complexity can be approximated as $O(m \cdot n)$, where m is the number of iterative calculation, and is set to 100 in the experiments. Compared to the Dubois algorithm in computing efficiency, the advantage is that the trigonometric function calculations are avoided, and the disadvantage is that the iterative calculation should be performed.

The X-Bragg algorithm is considered to extend the Bragg scattering algorithm for a slight roughness in the soil surface. It has a wider valid range for the roughness parameter, and is also not sensitive to the existence of slope. The X-Bragg algorithm is the real full polarimetric algorithm for soil surface, which utilizes both the amplitude and the phase information of full polarimetric channels. However, the inversion of this algorithm is not straightforward. The main steps are to compute the roughness from anisotropy, to construct the two-dimensional space of entropy and mean alpha, then to find out the dielectric constant by use of look-up-table (LUT) under certain conditions of incident angle and roughness. According to Equations (8)–(20), the algorithm complexity can be simplified as $d \cdot O(l \cdot n)$, where d indicates the algorithm complexity of matrix diagonalization, l represents the dimension of lookup table. Based on the above complexity analysis, it can be seen that the X-Bragg algorithm is the most complicated calculation, and is worthy of deep optimization.

According to the differences of the three inversion schemes, the key points of parallel computing are thread allocation, data storage, and instruction optimization. For the Dubois and Oh algorithms, two optimization methods were used in our experiment: thread allocation and instruction optimization. For the X-Bragg algorithm, we used a variety of optimization methods such as thread allocation, storage optimization, and instruction optimization.

3.2. GPU-Based Dubois and Oh Parallel Inversion

In principle, the Dubois algorithm can be seen as a simplification of the Oh inversion algorithm, so the optimization methods of the two inversion algorithms are roughly the same. The implementation of these two inversion algorithms includes the following parts: data acquisition, data preprocessing, inversion algorithm implementation, and data output. The calculation process of the inversion model is optimized in parallel, which can efficiently achieve the inversion of soil water content and roughness. The overall framework of the inversion algorithm is as follows:

In Figure 2, the black dotted frame is the part that needs to be optimized. The number of cycles of the calculation process is determined by the amount of data. This article uses two ways to optimize:

- (1) Thread allocation: In the thread allocation process, the computing power of the hardware needs to be considered. In this experimental environment, each block can be allocated up to 1024 threads, which does not mean that the number of threads per block is as high as possible. The amount of data used in the experiment is much larger than 1024, and the pixels remain independent during the calculation, so all threads are independent. Warp is the basic transmission unit of SM (streaming multiprocessor), and a warp has 32 threads. Therefore, the size of each thread block in this experiment is 16×16 . And the problem of limited storage space for threads is solved. This size ensures the full utilization of each scheduling unit and the threads have sufficient memory. It can make computing more efficient. Figure 3 shows the detailed thread allocation.
- (2) Instruction optimization: In Equations (1), (2), and (7), there are a large number of trigonometric and power functions. When parallel optimization is used, these functions are not applicable. In the CUDA runtime, there are some corresponding mathematical functions, and the calculation efficiency is higher under the condition of partial precision loss. For example, to replace the

function $\sin(\cdot)$ with the function $_sinf(\cdot)$. The calculation time of the inversion can be reduced by using the $_sinf(\cdot)$ function, which is an internal function of GPU.

3.3. GPU-Based X-Bragg Parallel Inversion

According to the principle of the analytic algorithm, X-Bragg is different from the other two inversion algorithms, and the lookup table is calculated before all data preprocessing. This table is used to find out the corresponding soil moisture under certain conditions of incidence angle and roughness. Figure 4 below shows the overall framework of the inversion algorithm based on X-Bragg.

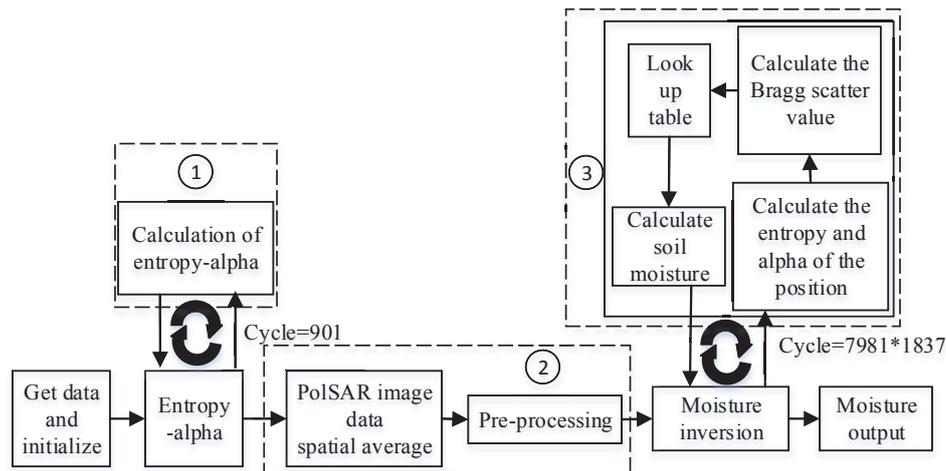


Figure 4. X-Bragg algorithm framework.

There are three main parts in the graph. The first one is to calculate the H and α according to X-Bragg algorithm, then H and α are stored in the lookup table. The second part is that the raw data needs to be spatially averaged for preprocessing etc. The third part is the inversion process of X-Bragg algorithm. This part solves the $[T]$ matrix for each corresponding pixel, and calculates real non-negative eigenvalues $\lambda_1, \lambda_2, \lambda_3$, and orthogonal eigenvectors \vec{e}_1, \vec{e}_2 , and \vec{e}_3 . Then the entropy H and α angle are computed corresponding to the eigenvalues and eigenvectors. Following that, the soil roughness can be calculated by scattering anisotropy A (ks), and finally with the obtained entropy H and α angle corresponding to the lookup table, soil moisture (mv) is inverted.

The size of data used in this experiment was 7981×1837 . After testing and averaging six times, the calculation time for each process is obtained. The calculation time of the first part is about 15 ms, and the calculation time of the second and third parts are about 7247 ms and 330,582 ms, respectively. The total computation time is 337,844 ms, in which the third part accounts for 97.85% of the total time. In the local environment, it takes more than five minutes to proceed with data of size 7981×1837 , which indicates that X-Bragg algorithm inversion is inefficient and parallelism. Since the third part of the time affects the real-time processing of the inversion, it is considered as the main part for optimization.

Figure 5 describes in detail the flowchart of CPU/GPU collaborative processing based on inversion algorithms.

In Figure 5, N_{diel} is the step size of the inverted dielectric constant. N_{beta} is the step size of the roughness angle in the inversion. Through the preliminary test, the display driver stopped responding during the calculation because of the execution time of the kernel function is too long. So the kernel function is divided into three parts: (1) Entropy H and α angle are calculated by high concurrent multithreading; (2) the position of the pixel corresponding to the scatter table is determined, and the entropy H and α angle are calculated using the zero-start consumption of the kernel function; (3) correspond to the lookup table, the moisture (mv) and roughness (ks) efficiently are calculated.

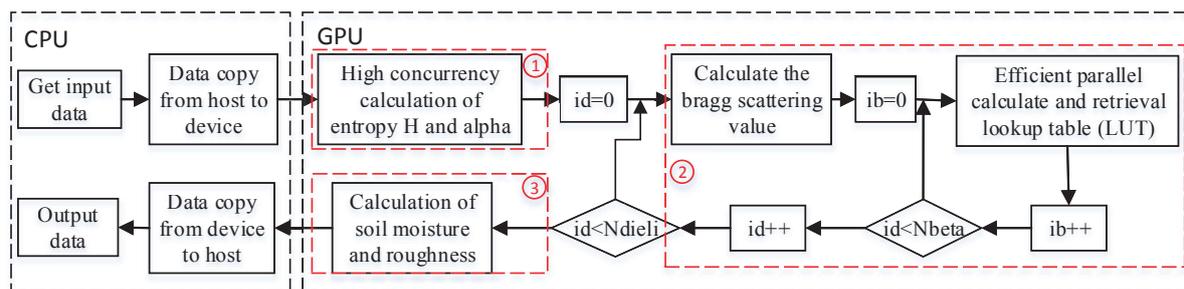


Figure 5. Optimization flowchart of the X-Bragg algorithm.

The pseudo code shows the details of the optimized X-Bragg inversion algorithm.

Parallel pseudo code of X-Bargg algorithm:

```

Step1:  Input polarization matrix S parameters
        for each GPU thread i [0 : Nlig*Ncol] do
Step2:  Each thread inverts an element target
        if (flag=1) Need to perform inversion
Step3:  T[18] ← Intermediate matrix calculated from the input matrix
Step4:  V[18] and lambda[3] ← Calculation of V complex eigenvector matrix,
        lambda real vector by diagonalisation(3, T, V,
        lambda)
Step5:  al[i], se[i] and valid[i] ← Calculation of scattering mechanism
        probability of occurrence and mean scattering
        mechanism
        end for
        for id [0 : Ndeili]
Step6:  Braggs and Braggp ← Calculation of the Bragg scattering value
        for ib [0 : Nbeta]
Step7:  T[18] ← Calculation of the intermediate variable T[18] based on the
        Bragg scattering value
Step8:  V[18] and lambda[3] ← Calculation of V complex eigenvector
        matrix, lambda real vector by
        diagonalisation(3, T, V, lambda)
Step9:  pos[i] ← Retrieval of minimum between entropy, alpha from (LUT)
        and entropy, alpha (data)
        end for
        end for
        for each GPU thread i [0 : Nlig*Ncol] do
Step10: Mmv_out[i] ← Calculation of soil moisture according to model formula and
        pos[i]
        end for
Step11: Output inversion result

```

In pseudo code, the T matrix is calculated. V is the eigenvector and $lambda$ is the corresponding eigenvalue. al and se are calculated as lookup tables. pos is the position in the LUT. $Nlig * Ncol$ is the

size of the data. *Mmv_out* is the output data. In the following part, a detailed optimization analysis of the X-Bragg algorithm is performed through four points.

V1: Thread allocation optimization

The thread allocation optimization is basically consistent with the analysis of Figure 2. In Figure 4, the experiment is divided into three kernel functions. The first reason is to calculate the time limit. In addition, if the single thread independently calculates the entire inversion process, it will lead to parallel branches. Considering the fact that when the entropy H and α angle are calculated by Equations (17) and (19), there is a threshold judgment $H \leq \max_H$ and $\alpha \leq \max_a$, while the inversion is only performed in range, so those threads do not perform inversion will be idle. The computation resources is wasted in this way. Therefore, in our experiment the kernel function is split before inversion, which can greatly increase the utilization rate of computing resources and avoid the waste of resources caused by parallel branches.

V2: Storage optimization

In the pseudo code, steps 5 and 9 use three constant arrays *lia_blockrange*, *max_en* and *max_al* of size 901. In the process of calculating and searching lookup tables, these three arrays are read multiple times. In general, data is transported from CPU to GPU global memory. Each thread needs to acquire data from the global memory for multiple times, hence the slow transmission time leads to the bottleneck of data processing. This problem can be solved well in hardware storage configuration. Constant memory has 64 kb of storage, and it is much larger than the size of the three arrays. So it is possible to pre-calculate the indices and addresses of the three arrays on CPU and uploaded them to the cached GPU constant memory, where they can be retrieved by the thread blocks at both high bandwidth and low latency. Besides, constant memory is a good fit for these three arrays in read-only operations. In this way, the transmission objective of the data is changed from 1 to 2, as shown in Figure 6.

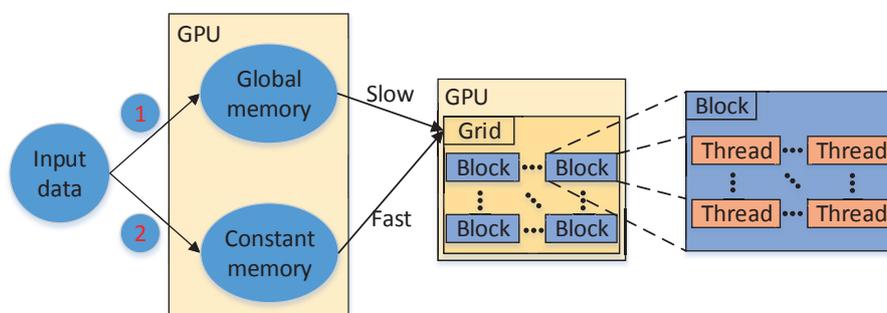


Figure 6. Storage optimization strategy.

V3: Data type conversion optimization

In steps 4 and 8, the diagonalization function is used for multiple times. By implementing Equation (13) $[T]$ matrix is diagonalized by an unitary similarity transformation, and the non-negative eigenvalue matrix $[\Lambda]$ and the eigenvector matrix $[U_3]$ are obtained. This process uses intermediate variables of double-type to make the diagonalization process more precise. But double-type data also limits the speed of GPU operations, while float-type is more suitable than double. Under the AIDA64 software test, the GTX 1080Ti provides a peak throughput of nearly 12.637 teraflop/s in float precision, but is limited to 423 gigaflop/s in double precision. In the GPU, using float data not only reduces memory consumption, but also improves data operation efficiency. Therefore, under the condition of partial accuracy loss, the usage of float-type increases the computational efficiency as well as saving storage resources for the hardware.

V4: Instruction optimization

In addition to the CUDA fast math optimization mentioned in the other two algorithms, loop expansion is also used for the optimization of the X-Bragg algorithm inversion. In general, a GPU is suitable for processing computationally intensive data, however its ability to do logical judgment is weak. In step 4, step 5, and step 8, the diagonalization process is used multiple times. There are a large number of loops in the process for calculating the real non-negative eigenvalue matrix $[\Lambda]$ and the eigenvector matrix $[U_3]$. Logical judgment has also become a huge bottleneck during GPU computing. By artificially expanding the loop within the kernel function, the instruction consumption is reduced as much as possible. Kernel performance is improved to get efficient calculations. In Figure 7, expanding the loop is vividly displayed. GPU is more suitable for computing than CPU. In terms of logical judgment, CPU has higher efficiency.

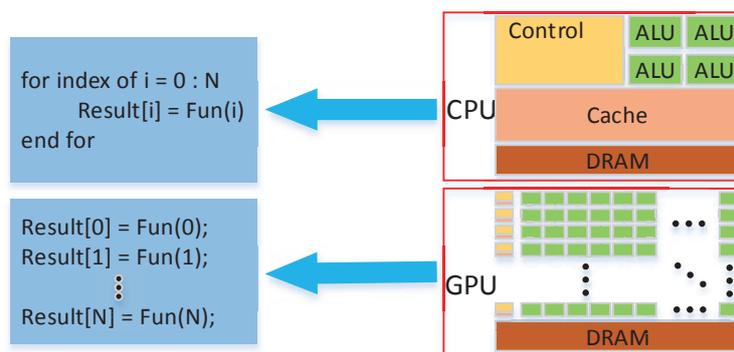


Figure 7. Loop unfolding optimization strategy.

4. Results

4.1. Experiment Environment

The hardware environment of experimentation includes Intel(R) core(TM) i5-3470 (CPU) and NVIDIA GeForce GTX 1080 Ti (GPU), and the library of visual studio 2013+ CUDA 9.0 is the software environment of the program.

4.2. Accuracy Analysis

Technically, there is no accuracy loss after these algorithms are implemented on GPU. However, there exists a small difference in the accuracy range of CPU and GPU math functions. As for GPU, the computing performance of single-precision floating-point data outperforms double-precision floating-point data by a lot. Therefore, the single-precision floating-point data type is applied for the GPU parallel design. Due to the above two reasons, the proposed methods may bring certain calculation errors, which should be analyzed to guarantee the algorithm accuracy.

Two indicators are employed to validate the parallel methods, mean absolute error (MAE) and root mean square error (RMSE), respectively. The pixel-wise comparisons of m_v are carried out among the three inversion models, as shown in Table 1. From the MAE and RMSE results, it can be seen that the errors from GPU parallel are very small and can be ignored. Meanwhile, the visual result comparisons on m_v and k_s are shown in Figure 8.

Table 1. Calculation error of m_v with CPU and GPU methods.

Model	MAE	RMSE
Dubois	1.53×10^{-4}	0.074
Oh	1.81×10^{-5}	0.016
X-Bragg	2.12×10^{-5}	0.03

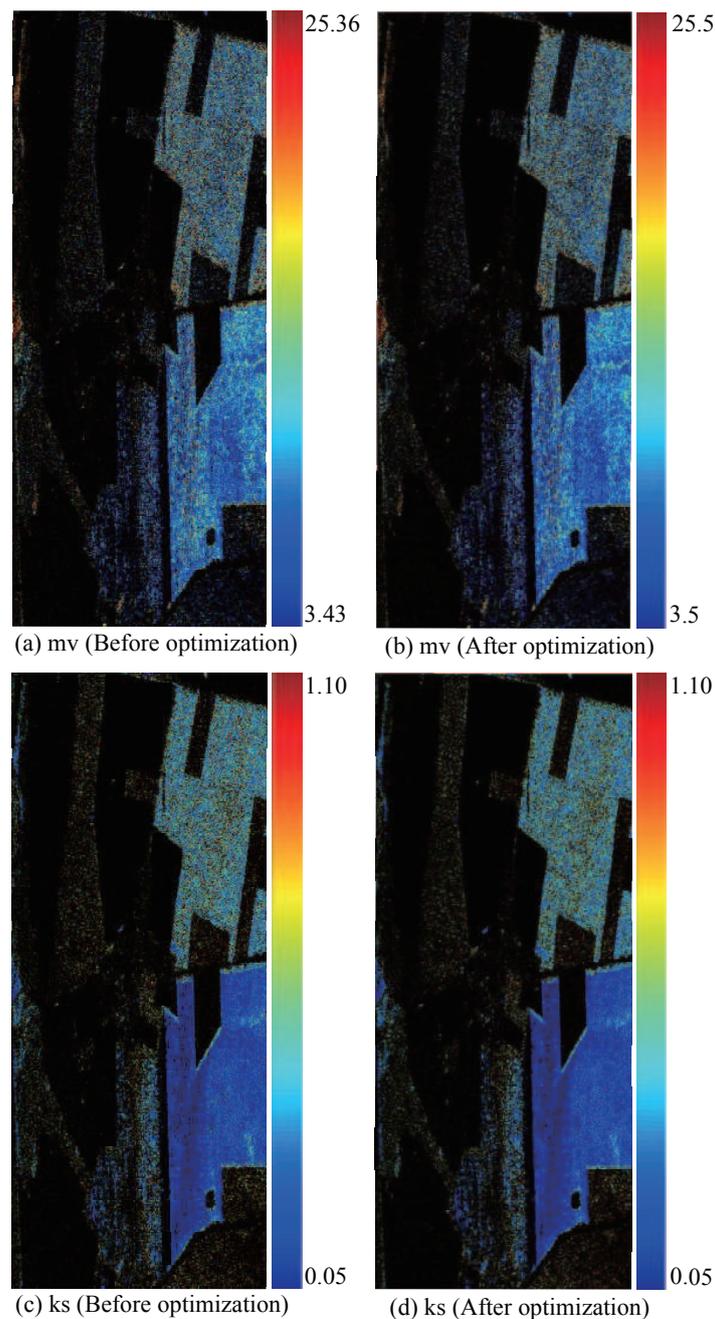


Figure 8. The soil parameter inversion comparison of sequential and parallel optimization Oh methods.

4.3. Optimization Results of the Dubois and Oh Parallel Inversion

The test site is the Demmin area in northern Germany, and the real PolSAR data is acquired by the ESAR airborne system of the German Aerospace Center. The original size is 7981×1837 . For comparison, different sizes of data are constructed by the upsampling and downsampling methods. The principle of data construction is to set a size more suitable for each inversion algorithm, which makes the computation more stable. In the above experimental environment, the Dubois and Oh algorithms were tested and compared using the same set of size data. Here, four sizes of data are tested, which are 13600×1837 , 6800×1837 , 3400×1837 , and 1700×1837 . After six tests, the average calculation time is used as the final results. Tables 1 and 2 below show the calculation time for Dubois and Oh algorithms at different data sizes.

In Tables 2 and 3, the row data represents the calculation time for different sizes. The first column indicates CPU computation time. The second column shows the transmission time of all data from CPU to GPU. The third column is the computation time of the kernel function. The fourth column shows the overall speedup results of the inversion based on the scattering algorithm.

Table 2. Acceleration results of Dubois inversion.

Size	CPU	Running Time (ms)		Speedup	
		Calculation	Data Transmission	Calculation	Overall
$13,600 \times 1837$	3814	25	204	$152\times$	$17\times$
6800×1837	1880	13	101	$144\times$	$16\times$
3400×1837	839	6	51	$139\times$	$15\times$
1700×1837	424	2	27	$212\times$	$14\times$

Table 3. Acceleration results of Oh inversion.

Size	CPU	Running Time (ms)		Speedup	
		Calculation	Data Transmission	Calculation	Overall
$13,600 \times 1837$	32,698	27	298	$1211\times$	$100\times$
6800×1837	15,453	13	149	$1188\times$	$95\times$
3400×1837	6633	7	74	$947\times$	$82\times$
1700×1837	3361	2	35	$1680\times$	$90\times$

Table 2 shows the optimization results of the Dubois model inversion. The performance of the GPU has only increased by about 15 times. As the amount of data decreases, the speedup effect is gradually weakened. This further illustrates pertinence of GPU for computationally intensive data. The acceleration can reach hundreds of times without considering the data transmission time. In Table 3, GPU is 100 times faster than CPU. Here, the computation time can be accelerated by thousands of times without considering the data transmission.

In terms of algorithm, Oh is more complicated than Dubois. In CPU calculation process, Oh is much slower than Dubois. This shows that the complexity of the calculation process has a profound impact on GPU usage. From the two algorithms optimization results, Oh is more suitable for GPU than Dubois. The Figure 9 is a more intuitive description of the optimization of these two algorithms.

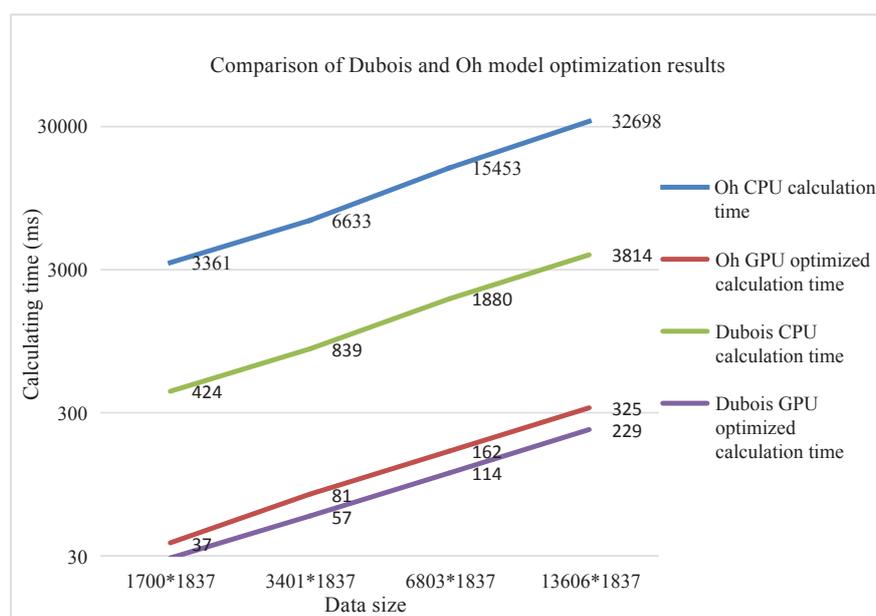


Figure 9. Comparison of the Dubois and Oh algorithm optimization results.

4.4. Optimization Result of the X-Bragg Parallel Inversion

In the above experimental environment, The 15862×1837 , 7981×1837 , 3990×1837 , and 1995×1837 size data were used in the X-Bragg algorithm. Table 4 details the calculation time of the algorithm in different situations.

Table 4. Acceleration results of the X-Bragg inversion.

Size	CPU	Running Time (ms)		Speedup	
		Final Calculation	Data Transmission	Calculation	Overall
$15,862 \times 1837$	66,0481	3808	368	173×	158×
7981×1837	330,508	1911	194	172×	157×
3990×1837	165,441	1250	97	132×	122×
1995×1837	82,813	441	48	187×	169×

In Table 4, CPU time is the time before optimization. The final computation time is the one after the final optimization. As can be seen from the table, the final result is about 150 times faster than CPU. In Figure 10, the acceleration effects of different data sizes can be clearly expressed.

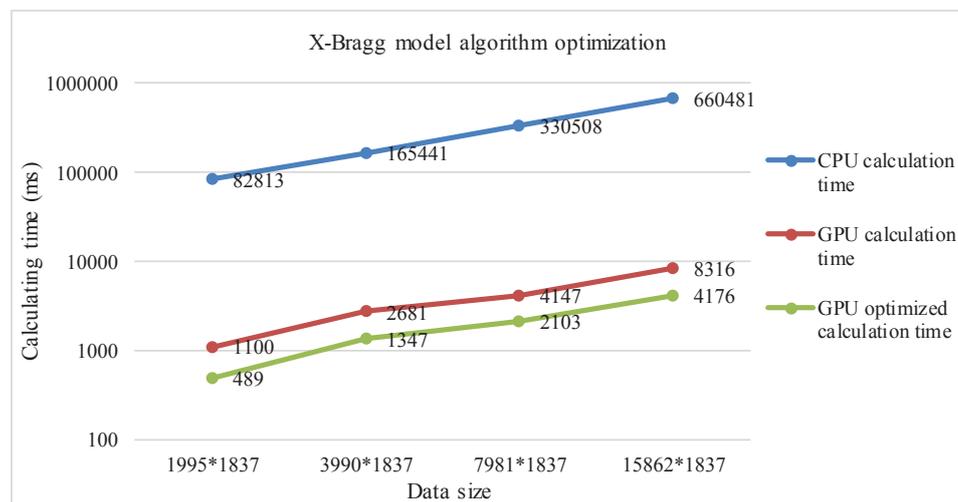


Figure 10. Acceleration performance comparison of the X-Bragg algorithm.

This experiment proposes four optimization methods based on the complexity of the algorithm. They are thread allocation, storage optimization, data conversion, and instruction optimization, respectively. Table 5 shows the time tested after each step optimization.

Table 5. Calculation time (ms) of the X-Bragg inversion after different optimization steps.

Size	V1	V2	V3	V4
$15,862 \times 1837$	7951	7799	4938	3808
7981×1837	3967	3888	2453	1911
3990×1837	2590	2511	1604	1250
1995×1837	1052	1034	557	441

In Table 5, the first column is the optimization result after reasonable thread allocation for hardware. The second column shows the effect after using constant memory. The third column shows the time after type conversion optimization. The fourth column is the final optimization result after the loop is expanded. From the first column to the fourth column optimization, the efficiency of the process is more than doubled. The acceleration performances of X-Bragg algorithm at different data size with different optimization methods are also shown in Figure 11.

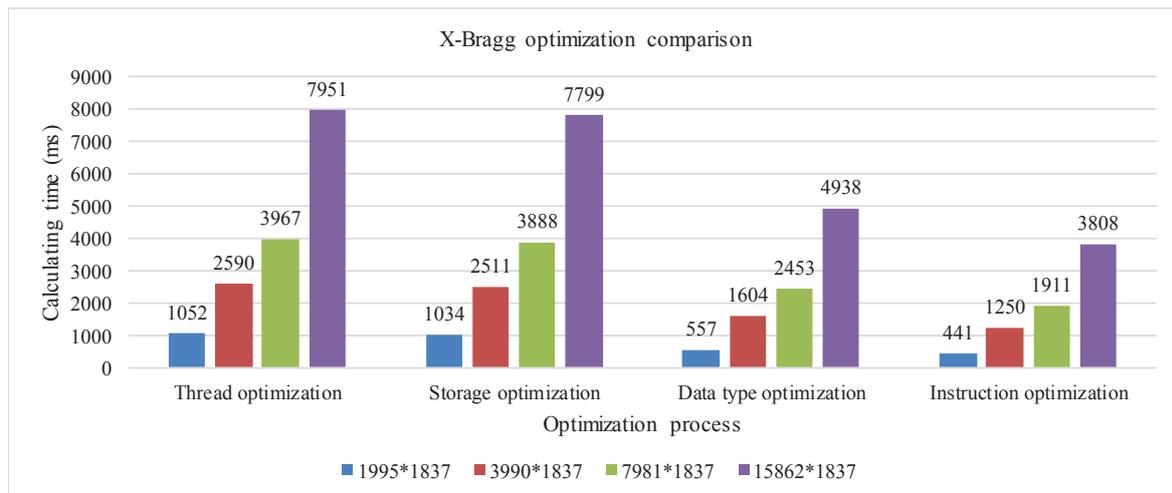


Figure 11. Results of using different optimization methods.

In Figure 11, bar graphs of different colors represent data of different sizes. With the implementation of the four optimization methods, the calculation time is continuously reduced. The number on the graph indicates the detailed calculation time (unit: ms). On the whole, the final optimization can reach more than $150\times$ speedup. Taking 15862×1837 data as an example, the inversion time is reduced from 11 min to 4 s.

5. Conclusions

In this paper, three classical forward scattering models and their corresponding inversion algorithms are analyzed. They are different in polarimetric data requirement, application situation, and performance. Through the further analysis of the structure of the three classical inversion algorithms based on scattering models, each algorithm is optimized, respectively. Then a framework for the parallel inversion method for polarimetric SAR imagery based on GPU is presented. The optimization combines the processing advantages of a GPU with computationally intensive imagery, so as to realize the parallel design of three inversion algorithms, including the entire inversion process from data transmission, computational instruction set, to GPU hardware structure. In the experiments, the calculation efficiency is increased by approximately 100-fold. For all widely used empirical models, the problems of large data volume and low computing efficiency are solved, including dual/multi/full polarization models. Experiments with real data fully demonstrate the tremendous advantages of combining GPU and PolSAR imagery for real-time processing. However, as far as the calculation time is concerned, the parallel X-Bragg method is still two orders of magnitude slower than the other two methods. In the future work, we will further optimize the diagonalization part to make it more efficient.

Author Contributions: Q.Y., F.Z. conceived of and supervised this study. Y.Z. and Q.Y. gave instructions to the basic framework. Y.W. performed the experiments. Q.Y. and Y.W. analyzed the results and wrote the paper. 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 under Grant No. 61801015, No. 61871413, and No. 61571422.

Acknowledgments: The PolSAR data was from the Advanced Training Course on Polarimetry sponsored by European Space Agency, with the contribution from Irena Hajnsek.

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

References

- Du, C.; Qin, Q.; Liu, M.; Feng, H.; Dong, H.; Wang, N. Soil moisture inversion and validation based on new remote sensing platform. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Melbourne, Australia, 21 July 2013; pp. 2728–2731.

2. Kweon, S.K.; Oh, Y. Estimation of soil moisture and surface roughness from single-polarized radar data for bare soil surface and comparison with dual-and quad-polarization cases. *IEEE Trans. Geosci. Remote Sens.* **2013**, *52*, 4056–4064. [[CrossRef](#)]
3. Verstraeten, W.; Veroustraete, F.; Feyen, J. Assessment of evapotranspiration and soil moisture content across different scales of observation. *Sensors* **2008**, *8*, 70–117. [[CrossRef](#)] [[PubMed](#)]
4. Yin, Q.; Cao, F.; Hong, W. Analysis of Valid Ranges in Soil Inversion Models Based on the Cloude-Pottier Decomposition. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Boston, MA, USA, 7–11 July 2008; pp. II-821–II-823.
5. Dobriyal, P.; Qureshi, A.; Badola, R.; Hussain, S.A. A review of the methods available for estimating soil moisture and its implications for water resource management. *J. Hydrol.* **2012**, *458*, 110–117. [[CrossRef](#)]
6. Jiao, J.; Zhang, Y.; Sun, H.; Yang, X.; Gao, X.; Hong, W.; Fu, K.; Sun, X. A Densely Connected End-to-End Neural Network for Multiscale and Multiscene SAR Ship Detection. *IEEE Access* **2018**, *6*, 20881–20892. [[CrossRef](#)]
7. Li, H.; Krylov, V.; Fan, P.; Zerubia, J.; Emery, W. Unsupervised Learning of Generalized Gamma Mixture Model with Application in Statistical Modeling of High-Resolution SAR Images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 2153–2170. [[CrossRef](#)]
8. Chen, H.; Zhang, F.; Tang, B.; Yin, Q.; Sun, X. Slim and efficient neural network design for resource-constrained SAR target recognition. *Remote Sens.* **2018**, *10*, 1618. [[CrossRef](#)]
9. Gao, F.; Xue, X.; Sun, J.; Wang, J.; Zhang, Y. A SAR image despeckling method based on two-dimensional S-transform shrinkage. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3025–3034. [[CrossRef](#)]
10. Gao, F.; Yang, Y.; Wang, J.; Sun, J.; Yang, E.; Zhou, H. A Deep Convolutional Generative Adversarial Networks (DCGANs)-Based Semi-Supervised Method for Object Recognition in Synthetic Aperture Radar (SAR) Images. *Remote Sens.* **2018**, *10*, 846. [[CrossRef](#)]
11. Srivastava, P.K.; Pandey, P.C.; Petropoulos, G.P.; Kourgialas, N.N.; Pandey, V.; Singh, U. GIS and Remote Sensing Aided Information for Soil Moisture Estimation: A Comparative Study of Interpolation Techniques. *Resources* **2019**, *8*, 70. [[CrossRef](#)]
12. Piles, M.; Petropoulos, G.P.; Sanchez, N.; Gonzalez-Zamora, A.; Ireland, G. Towards improved spatio-temporal resolution soil moisture retrievals from the synergy of SMOS and MSG SEVIRI spaceborne observations. *Remote Sens. Environ.* **2016**, *180*, 403–417. [[CrossRef](#)]
13. Yin, Q.; Hong, W.; Zhang, F.; Pottier, E. Optimal Combination of Polarimetric Features for Vegetation Classification in PolSAR Image. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2019**, *12*, 3919–3931. [[CrossRef](#)]
14. 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](#)]
15. Jung, S.G.; Hong, J.Y.; Oh, Y. Verification of Surface Scattering Models and Inversion Algorithms with the Polarimetric Backscatter Measurements of a Bare Soil Surface. In Proceedings of the Asia-Pacific Microwave Conference, Bangkok, Thailand, 11 December 2007; pp. 1–3. [[CrossRef](#)]
16. Qiu, C.; Chen, Y.; Tong, L.; Jia, M.; Pang, S. The method for soil moisture inversion based on ground-based scattering measurement. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Vancouver, BC, Canada, 24 July 2011; pp. 3086–3088.
17. Kweon, S.K.; Park, S.M.; Oh, Y. Improvement of soil moisture inversion for single-polarized SAR data of bare soil surfaces using DInSAR technique. In Proceedings of the IEEE Geoscience and Remote Sensing Symposium, Quebec City, QC, Canada, 13 July 2014; pp. 3236–3238.
18. Xue, Q.; Sheng, Q.; Ma, S. Application of Optimum Perturbation Algorithm for Parameter Inversion Identification of Soil Moisture Model under Ecological Slope Protection. In Proceedings of the IEEE International Conference on Artificial Intelligence and Computational Intelligence, Shanghai, China, 7 November 2009; Volume 1, pp. 204–207.
19. Wang, Y.; Li, Y.; Sun, S.; Zhou, Q.; Han, N. Research on inversion of soil water dynamic parameters by field soil moisture content. In Proceedings of the IEEE International Conference on New Technology of Agricultural, Zibo, China, 27 May 2011; pp. 338–343.
20. Oh, Y.; Jung, S.G. Inversion Algorithm for Soil Moisture Retrieval from Polarimetric Backscattering Coefficients of Vegetation Canopies. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Boston, MA, USA, 7 July 2008; Volume 2, pp. II-402–II-405.

21. Oh, Y. Comparison of two inversion methods for retrieval of soil moisture and surface roughness from polarimetric radar observation of soil surfaces. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Anchorage, AL, USA, 20 September 2004; Volume 2, pp. 807–810.
22. Geddert, N.; Jeffrey, I. A Dynamically Balanced OpenMP-CUDA Implementation of PDE-Based Contrast Source Inversion for Microwave Imaging. In Proceedings of the International Symposium on Antenna Technology and Applied Electromagnetics, Waterloo, ON, Canada, 19 August 2018; pp. 1–2.
23. Li, M.; Wang, X.Y.; Abubakar, A. Accelerating nonlinear inversion algorithms on GPU platform for electromagnetic data. In Proceedings of the International Symposium on Antennas and Propagation, Okinawa, Japan, 24 October 2016; pp. 574–575.
24. Wang, X.Y.; Li, M.; Abubakar, A. Acceleration of multiplicative regularized contrast source inversion algorithm using paralleled computing architecture. In Proceedings of the IEEE Progress in Electromagnetic Research Symposium, Shanghai, China, 8 August 2016; pp. 1739–1743.
25. Ries, F.; De Marco, T.; Zivieri, M.; Guerrieri, R. Triangular matrix inversion on graphics processing unit. In Proceedings of the ACM Conference on High Performance Computing Networking, Storage and Analysis, Portland, OR, USA, 14 November 2009; pp. 1–10.
26. Gao, C.; Li, L.; Zhao, Z.; Huang, H. Parallel computation of the large grounding grids in multi-layer soil using moment method. In Proceedings of the IEEE World Automation Congress, Hawaii, HI, USA, 28 September 2008; pp. 1–4.
27. Gomez-Calvino, J.; Colominas, I.; Navarrina, F.; Casteleiro, M.; Cela, J. Parallel computing aided design of earthing systems for electrical substations in non-homogeneous soil models. In Proceedings of the IEEE International Workshop on Parallel Processing, Toronto, ON, Canada, 21 August 2000; pp. 381–388.
28. Zhang, F.; Hu, C.; Wu, P.; Zhang, H.; Wong, M. Accelerating aerial image simulation using improved CPU/GPU collaborative computing. *Comput. Electr. Eng.* **2015**, *46*, 176–189.
29. Hu, C.; Zhang, F.; Li, G.; Li, W.; Cui, Z. Computation Reduction Oriented Circular Scanning SAR Raw Data Simulation on Multi-GPUs. *J. Radars* **2016**, *5*, 434–443. [[CrossRef](#)]
30. Li, Z.; Su, D.; Zhu, H.; Li, W.; Zhang, F.; Li, R. A Fast Synthetic Aperture Radar Raw Data Simulation using Cloud Computing. *Sensors* **2017**, *17*, 113.
31. Tang, H.; Li, G.; Zhang, F.; Hu, W.; Li, W. A spaceborne SAR on-board processing simulator using mobile GPU. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Beijing, China, 10 July 2016; pp. 1198–1201. [[CrossRef](#)] [[PubMed](#)]
32. Zhang, F.; Hu, C.; Li, W. A deep collaborative computing based SAR raw data simulation on multiple CPU/GPU platform. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *10*, 387–399.
33. 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](#)]
34. Li, G.; Zhang, F.; Ma, L.; Hu, W.; Li, W. Accelerating SAR imaging using vector extension on multi-core SIMD CPU. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Milan, Italy, 26 July 2015; pp. 537–540. [[CrossRef](#)]
35. Hu, C.; Zhang, F.; Ma, L.; Li, G.; Hu, W.; Li, W. Efficient SAR raw data parallel simulation based on multicore vector extension. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Milan, Italy, 26 July 2015; pp. 4719–4722.
36. Oh, Y.; Sarabandi, K.; Ulaby, F.T. An empirical model and an inversion technique for radar scattering from bare soil surfaces. *IEEE Trans. Geosci. Remote Sens.* **1992**, *30*, 370–381.
37. Smith, J.R.; Mirotznic, M.S. Rough surface scattering models. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Anchorage, AL, USA, 20 September 2004; Volume 5, pp. 3107–3110. [[CrossRef](#)]
38. Hajnsek, I.; Pottier, E.; Cloude, S.R. Inversion of surface parameters from polarimetric SAR. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 727–744.

