

Article Calculation of Surface Offset Gathers Based on Reverse Time Migration and Its Parallel Computation with Multi-GPUs

Dingjin Liu^{1,*}, Bo Li¹ and Guofeng Liu^{2,*}



- ² School of Geophysics and Information Technology, China University of Geosciences, Beijing 100083, China
- * Correspondence: liudj.swty@sinopec.com (D.L.); liugf@cugb.edu.cn (G.L.)

Abstract: As an important method for seismic data processing, reverse time migration (RTM) has high precision but involves high-intensity calculations. The calculation an RTM surface offset (shot-receiver distance) domain gathers provides intermediary data for an iterative calculation of migration and its velocity building. How to generate such data efficiently is of great significance to the industrial application of RTM. We propose a method for the calculation of surface offset gathers (SOGs) based on attribute migration, wherein, using migration calculations performed twice, the attribute profile of the surface offsets can be obtained, thus the image results can be sorted into offset gathers. Aiming at the problem of high-intensity computations required for RTM, we put forth a multi-graphic processing unit (GPU) calculative strategy, i.e., by distributing image computational domains to different GPUs for computation and by using the method of multi-stream calculations to conceal data transmission between GPUs. Ultimately, the computing original efficiency was higher relative to a single GPU, and more GPUs were used linearly. The test with a model showed that the attributive migration methods can correctly output SOGs, while the GPU parallel computation can effectively improve the computing efficiency. Therefore, it is of practical importance for this method to be expanded and applied in industries.

Keywords: reverse time migration (RTM); surface offset gathers (SOGs); parallel computation; graphic processing unit (GPU)

1. Introduction

As an important branch of geophysics, seismic exploration uses seismic waves excited on the Earth's surface from seismic sources, such as explosions, and receives seismic waves reflected on the Earth's surface from underground to image underground structures. It is widely applied in the exploration of oil, gas, minerals, and the probing of urban underground structures [1].

As an important step in seismic data processing, migration image processing obtains image profiles of underground structures by using acquired seismic data, which are then used to explain geological structures. The imaging methods of exploratory seismic data include the Kirchhoff method, the one-way wave equation method, and reverse time migration (RTM) [2,3], of which RTM currently enjoys the highest precision.

In RTM calculations, although the velocity model of underground structures is one of the vital factors influencing imaging precision, a pre-estimated initial velocity model is often used to migrate, then the initial velocity model is updated with the imaging results, thus generating an iterative process in the calculation of migration and velocity models due to the precise velocity of underground models being unknown to us. For such iterations, an intermediary data volume is needed, which we refer to as surface offset gathers (SOGs) [4,5].

Currently, RTM-based SOGs are a research hotspot. Among these, the shot-domain RTM, the simplest method to output SOGs, divides receiver data into a few parts by



Citation: Liu, D.; Li, B.; Liu, G. Calculation of Surface Offset Gathers Based on Reverse Time Migration and Its Parallel Computation with Multi-GPUs. *Appl. Sci.* **2021**, *11*, 10687. https://doi.org/10.3390/ app112210687

Academic Editor: Forte Emanuele

Received: 4 August 2021 Accepted: 11 November 2021 Published: 12 November 2021

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



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



offsetting and then migrating them, respectively, and uses the mean offset to identify different results so as to generate SOGs. In this method, it is necessary to perform a migration calculation on data from every shot, so the calculation intensity is too large for it to be implemented on industry scale [6]. The improved algorithm by Zhao et al. only calculates a minority of sections of different offsets, and then supplements the necessary data in offset gathers through interpolation [7]. Another method to output SOGs is the wave equation method, called wave-equation Kirchhoff migration, which is named as such because the Born equation it uses is similar to the Kirchhoff integration equation. The difference between them is that in the calculation, the wave equation, rather than ray tracing, is used in the Green's function for propagation between shots and receivers [8,9]. The attribute migration maps the geometrical and physical attributes in seismic data, such as travel time, angles of reflection and incidence, and shot-receiver distance, to the migration results through migration calculations performed twice, and extracts attributes from the ratios of the results from imaging performed twice. This was proposed by Bleistein [10] and applied in the calculation of three-dimensional (3D) stereotomography [11]. Giboli et al. and Lemaistre et al. discussed the possibility of applying the method to output the attributes of the shot-receiver distances [12,13]. In this study, based on attributive migration, we investigated how to apply this method to the calculation of RTM SOGs.

The seismic migration calculation enjoys high computational intensity, and both industry and academia have carried out much research on the acceleration of seismic imaging. Owing to their multiple kernels and bandwidth, as well as advantages of low power consumption and small occupied space, graphic processing units (GPUs) have been introduced in general computations [14]. The application of NVIDIA GPUs in seismic image processing has significantly improved computing efficiency, for instance, in time migration and RTM and waveform inversion. [15–17], especially RTM, because it has the highest calculation accuracy in the seismic migration but is also the most time-consuming. Therefore, when GPUs appeared, it was used to accelerate RTM computation [18] by including useof random boundary instead of storing wavefields to save GPU global memory [19], as well as some GPU computing strategies for the calculation of specific form of RTM tilted transversely isotropic TTI RTM and Q-RTM [20,21]. Each type of RTM attempts to take full advantage of the GPU's characteristics of adapting to high-density computing, the algorithm of RTM is also being changed to adapt to the GPU architecture [22].

Since a single GPU is small in memory and is unfit for the calculation of RTM SOGs that require large memory space, multiple GPUs allow the expansion of the processor memory so as to effectively solve this problem [23].

Based on the typical characteristics of multi-GPU programming and high memory demand of RTM SOGs, we propose a strategy to accelerate the calculation of RTM SOGs with multi-GPUs, the typical feature beings that this strategy can conceal the time required for data transmission between GPUs, thus achieving an increase in computing efficiency in linear proportion to the increase in the number of GPUs.

This paper first describes the attribute migration-based strategy for the calculation of SOGs and elucidates this process using a two-dimensional (2D) model. Then, based on the process flow of the image calculation, we propose a multi-GPU calculation strategy, for which we used a 3D salt model to test the efficiency and imaging effect.

2. Methodology

2.1. Theory of Shot-Domain Migration Aided by Attribute Migration to Output SOGs

The equation using the Born equation to express the prestack shot-domain depth migration is:

$$I(\boldsymbol{x},\,\boldsymbol{s}) = \iint G(\boldsymbol{x},\,t|\boldsymbol{s},\,0)G(\boldsymbol{x},-t|\boldsymbol{r},\,0) * \ddot{d}(\boldsymbol{r},\,t|\boldsymbol{s},\,0)\mathrm{d}\boldsymbol{r}\mathrm{d}t \tag{1}$$

where *x* is the location of an imaging point; *s* is the location of the shot; *r* is the location of the receiver; I(x, s) is the imaging result of the shot *s* at the point *x*; G(x, t | s, 0) is Green's function for forward propagation of shot-point data; G(x, -t | r, 0) is Green's function for

backward propagation of receiver data; d(r, t | s, 0) is the second derivative at the shot s and the receiver r versus time [24].

With a migration calculation performed twice, the quantitative parameters weighted on the data can be estimated, which is described as the following process:

$$I_{\mathbf{w}}(\boldsymbol{x},\,\boldsymbol{s}) = \iint G(\boldsymbol{x},\,t|\boldsymbol{s},\,0)G(\boldsymbol{x},-t|\boldsymbol{r},\,0) * (\ddot{d}(\boldsymbol{r},\,t|\boldsymbol{s},\,0)\cdot\boldsymbol{w}(\boldsymbol{x}|\boldsymbol{s},\,\boldsymbol{r}))\mathrm{d}\boldsymbol{r}\mathrm{d}t,\tag{2}$$

where w(x | s, r) are the weighting parameters associated with the locations of the shot and the receiver at the imaging point, acting directly on the pre-migration data, which may be the angles of incidence and reflection and the coordinates of the shot and the receiver. The weighting parameter can be estimated with the ratio of the results from the two migrations using Equations (1) and (2).

$$w(\mathbf{x}|\mathbf{s},\,\mathbf{r}) \approx \frac{I_{\mathrm{W}}(\mathbf{x},\,\mathbf{s})}{I(\mathbf{x},\,\mathbf{s})}.$$
(3)

If the weighting coefficient w(x | s, r) is the offset attribute, the value of the offset at the location of the imaging point can be calculated, thus outputting the SOGs.

Equations (1) to (3) are simplified to the cross-correlation imaging conditions of RTM and are specifically expressed as:

$$I(\boldsymbol{x}, \boldsymbol{s}) = \int u(\boldsymbol{x}, \boldsymbol{s}, t) \int v(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{r}, t) d\boldsymbol{r} dt, \qquad (4)$$

where u(x, s, t) is the shot wavefield at the time t; v(x, s, r, t) is the receiver wavefield at s at the time t.

Multiplying the shot–receiver distance h = s - r and the shot-domain receiver data and then migrating the product results in:

$$I_{\rm h}(\boldsymbol{x},\,\boldsymbol{s}) = \int u(\boldsymbol{x},\,\boldsymbol{s},\,t) \int \boldsymbol{h} \boldsymbol{v}(\boldsymbol{x},\,\boldsymbol{s},\,\boldsymbol{r},\,t) \,\mathrm{d}\boldsymbol{r} \mathrm{d}t. \tag{5}$$

Using the ratio between Equations (4) and (5), the shot–receiver distance obtained from a single shot at the imaging point x is:

$$h_A(\mathbf{x}, \mathbf{s}) = \frac{I_h(\mathbf{x}, \mathbf{s})}{I(\mathbf{x}, \mathbf{s})}.$$
(6)

According to the obtained values of the offset at various points in the imaging domain, imaging results are placed from every shot to the proper SOG. The final SOGs can be obtained by calculating the records at all shots successively and then iterating by

$$I(\mathbf{x}, \mathbf{h}) = \int I(\mathbf{x}, \mathbf{s}) \delta(\mathbf{h} - h_A(\mathbf{x}, \mathbf{s})) d\mathbf{s},$$
(7)

where δ is the pulse function.

As for 3D data, Equations (4)–(6) can be used to calculate the horizontal and vertical offset, respectively, and then obtain the total value of the offset.

According to Equation (5), the result from the weighted data migration of horizontal shot–receiver distances is

$$I_{\rm hx}(\mathbf{x}, \mathbf{s}) = \int u(\mathbf{x}, \mathbf{s}, t) \int (\mathbf{r}_{\rm x} - \mathbf{s}_{\rm x}) v(\mathbf{x}, \mathbf{s}, \mathbf{r}, t) d\mathbf{r} dt, \qquad (8)$$

Similarly, the result from the weighted data migration of vertical shot–receiver distances is

$$I_{\rm hy}(\mathbf{x},\,\mathbf{s}) = \int u(\mathbf{x},\,\mathbf{s},\,\omega) \int (\mathbf{r}_y - \mathbf{s}_y) v(\mathbf{x},\,\mathbf{s},\,\mathbf{r},\,\omega) \mathrm{d}\mathbf{r} \mathrm{d}t,\tag{9}$$

The values of the shot–receiver distances in two directions from Equation (6) are, respectively,

$$h_{Ax}(x, s) = \frac{I_{hx}(x, s)}{I(x, s)}, h_{Ay}(x, s) = \frac{I_{hy}(x, s)}{I(x, s)}.$$
 (10)

Based on the obtained attributes of the offset in two directions, the calculation is repeated for every shot to obtain the final imaging results, which are

$$I(\boldsymbol{x}, \boldsymbol{h}) = \int I(\boldsymbol{x}, \boldsymbol{s}) \delta\left(\boldsymbol{h} - \sqrt{h_{Ax}^2 + h_{Ay}^2}\right) \mathrm{d}\boldsymbol{s}.$$
 (11)

2.2. RTM SOG Test Using the 2D Marmousi Model

The aforesaid calculative process can be validated using the 2D Marmousi model. First, modulation calculation is performed to all of the 240 single-shot records by multiplying each trace data by the shot to receiver distance, the single-shot records before and after modulation, as shown in Figure 1. Second, RTM is performed on the original shot data and modulated shot data, leading to two migrated data of each shot being obtained. Third, the offset values can be calculated with two migrated results with Equation (6), the original shot data migrated data can be sorted to an offset gather by these values. Finally, every singleshot is summed up to the offset gather and obtain the final SOGs.



Figure 1. (**a**) The original single-shot records from the Marmousi model and (**b**) the results from the modulation of offsets.

The original data and modulated data are migrated using the same imaging parameters, respectively (e.g., Figure 1a,b) and solve the ratio of the results (Figure 2a,b) from the two migrations to obtain the value of the offset at the imaging point. The imaging result is sortedby the calculated values of the surface offsets (Figure 2c) into SOGs, and the records of all the shot gathers are calculated and inserted following the aforesaid processes to obtain the SOGs of the imaging results.



Figure 2. (**a**) Results from the migration of the original single shots; (**b**) results from the migration the of modulated single shots; (**c**) attributes of the surface offsets obtained from ratios of (**a**,**b**).

In the calculation of SOGs, for gathers, the minimum offset was set at 200 m, the maximum offset was set at 2600 m, and the step of the offset was 100 m. Figure 3 shows two SOGs obtained from a calculation at common middle point CMP 150 and CMP 350, respectively. RTM SOGs were mainly used to update the velocity model in the offset calculations, and the events in gathers were horizontal, indicating that the velocity model used in the migration was accurate. Where the velocity was low, the event features warped upward, and where it was high, the event tilted downward. Figure 4 shows the SOGs calculated after the values of the velocity model value were adjusted up and down by 5%, respectively, from which we can see the changes of the gathers; when the velocity is higher than the correct velocity, the events in the gathers will bend down, and when the velocity is lower than the correct velocity, the events will bend up, which also demonstrates that the gathers calculated with RTM can be used for velocity updates. Some methods, such as tomography or residual velocity analysis, will convert the event curvature to the velocity, and acquire the updated velocity model.



Figure 3. RTM SOGs calculated with attribute migration: (**a**) gathers at about CMP 150, and (**b**) gathers at about CMP 350.

2.3. Process Flow for RTM-Based Calculation to Output SOGs

The RTM calculation is high in intensity, requiring parallel computing. With the development of computer technology, especially the emergence of GPUs, the RTM calculation can be improved significantly by accelerating the RTM finite difference with multiple kernels, substantially raising the computing efficiency of RTM, thus promoting its calculative application in industries. In the RTM calculation using a GPU, to break through the bandwidth bottleneck between the central processing unit (CPU) and the GPU, random boundary processing conditions [19] are used to calculate the wavefield propagation of shots twice to exchange computation for storage, which saves time consumed for data transmission. The RTM calculation process on the GPU is as follows: the shot wavefield is forward-propagated to the boundary where it is randomized and reversed; the receiver wavefield is propagated backward; both are imaged at the same time; the shot-by-shot migration is completed to obtain the entire imaging profile.



Figure 4. Imaging profiles and gathers at the velocity increased (**a**) and the enlarged display of partial gathers (**b**) decreased by 5% (**c**), and (**d**) enlarged display of partial gathers (**d**).

The shot-domain RTM migration outputs the zero shot-receiver distance profile, namely, the stacked profile, and in this study, the strategy of attribute migration was used to output SOGs from the RTM calculation. According to the aforesaid theory on attribute migration, it was necessary to calculate the values of offset at imaging points using the ratio of the results from the migration of the modulated and original shot data, thus sorting to obtain the migrated SOGs. Compared with the stacked profile RTM calculation process, the two kinds of migration results are the same, except for the input receiver data, and other imaging elements are the same. Therefore, only the RTM calculation process needs to be modified in order to be used in a calculation to output the offset gathers; thus, the calculation process is now as follows: the shot wavefield is forward-propagated to the boundary where it is randomized and then backward-propagated; the wavefields of both the receivers and the modulated receivers are backward-propagated, respectively; one shot wavefield and two receiver wavefields are coherently imaged at the same time, outputting results from two kinds of single-shot migration; the ratio of the two wavefields is used to obtain the offset attribute, thus sorting to generate SOGs. The above calculation and sorting are completed shot by shot and obtain the final results of the imaged SOGs.

Figure 5 shows the process for the RTM-based calculation to output SOGs. Obviously, if the imaging part of the modulated wavefields is removed, it is a common RTM. Since the forward and reverse propagations of the shot wavefield are unchanged, only the modulated wavefield is added to the forward propagation of the receivers, so the total calculation is only increased by about one-third. In the next section, we will introduce how to improve these wave propagation kernels on GPUs.



Figure 5. The process for RTM-based calculation to output SOGs.

2.4. Calculation of RTM SOGs with Multi-GPUs

In the RTM calculation, the kernel of the wavefield propagation is a finite-difference calculation. Among them, the best GPU calculation strategy for spatial difference has been achieved in previous studies with a single GPU [25]; it can achieve nearly 100% occupancy, defined as the ratio of active warps on an SM to the maximum number of active warps supported by the SM. However, with an increasing demand on computation, the memory of a single GPU cannot meet the requirements for larger storage space. A GPU enables two solutions: (1) it provides a unified memory space in which a memory pool is built using larger CPU memory, thereby enabling data in the memory pool to be read by the CPU and GPU simultaneouslysolving the demand for large memory, but reading the memory slowly; (2) it introduces a multi-GPU calculation, which achieves the common calculation by dividing the computational domain into different sub-cells, placing them in different GPUs within the same network for calculation, and exchanging data in the overlapped domain when necessary. We used two GPUs as an example and focused on expounding the second solution. In the calculation, the wavefield space to be calculated was divided into two mutually overlapped subspaces, in the z direction, and they were calculated on two GPUs, respectively. For the single GPUs in the multi-GPU calculation, the finite-difference calculation was still based on the conventional methods, and the only distinction was that after the finite difference of each time step was completed, data exchange in the overlapping area was needed.

The difference calculation space was divided in two GPUs as shown in Figure 6, in which Figure 6a shows the entire computational space, and Figure 6b shows the computational space partitioned into two equal subspaces. However, based on the order in the finite-difference calculation, an overlapped domain was needed for data exchange and updating, as the green part shows in Figure 6b. Figure 6c is a slice along the z direction, in which the red is the computational domain, and the green is the overlapped domain for data exchange.



Figure 6. The schematic showing the partitioned computational space in the depth direction: (**a**) the entire computational domain; (**b**) two parts of the computational domain which are partitioned in the z direction, with an overlapped domain; (**c**) slices displayed in the z direction.

In terms of memory management methods, for multi-GPU computations, the computational process can be illustrated as shown in Figure 7. The entire imaging space was divided into two independent parts in the z direction, which were calculated with GPU0 and GPU1, respectively. Some of the overlapped domain was used to exchange boundary data at different times of propagation, and the radius was determined by the finite-difference order. For each GPU computation, we focused on the management and calculation of data in the overlapped domain, which involved data transmission between two GPUs. To realize the overlap of calculation and data transmission, it was necessary to adopt different methods when dividing the computational domain. For example, for GPU1 computations, the computational domain partitioned to it was further divided into three parts, of which, d_wf11 and d_wf21 were calculated in different job streams, respectively, and after d_wf20 was computed, it was directly exchanged to the corresponding domain of GPU1 through point-to-point transmission, and GPU1 was calculated using the same process as for GPU0. Here, the magnitude of d_wf20 and d_wf11 was by the order of spatial finite difference and was much smaller than that of d_wf10 and d_wf20. If the computation and transmission of a small domain on a single GPU are placed in the same stream, when the computation is not completed in the larger domain, the computation and transmission are completed in the small area, so the transmission of data between different GPUs is completely concealed.



Figure 7. Memory management methods for the calculation of partitioned imaging spaces in two GPUs.

Based on the above analysis, in the RTM calculation of offset gathers, we used two GPUs to establish job streams to achieve the overlap of all calculations and transmissions between GPU0 and GPU1, and realized the calculation of wavefields of shots and receivers

in GPUs at the same time.Each GPU contained seven independent job streams, and the content of the computation included the propagation of shots and receivers and the P2P transmission between two GPUs. Both GPUs had the same types of jobs, but they computed on different parts of the imaging space; the entire process is shown in Figure 8.



Figure 8. The process flow for RTM calculation of SOGs on dual GPUs.

Figure 9 shows the shot wave-filed propagation slice processing on two GPUs based on the process shown in Figure 8. GPU0 calculated the upper half of the shot propagation, while GPU1 calculated the lower half, and stream 0 calculated the small zone 0-0 of the imaging space and copied the data in the overlapped domain to zone 1-1 of GPU1, while stream1 calculated the main part of the shot propagation. Stream 7 of GPU1 calculated the small zone 1-0 in the lower half and copied it to zone 0-1 of GPU0, while stream 8 calculated the main part of the lower half. The composition for the calculation of the propagation of the two receivers was consistent with that of the shot propagation.



Figure 9. Calculation and data exchange of shot wavefields on 2 GPUs.

2.5. Testing of Calculation with 3D Salt Model and Analysis of Efficiency of Multi-GPUs Computation

The 3D salt model is a model usually used in calculation tests. The velocity model grid size was 901 (subline) \times 901 (crossline) \times 250 (depth), at the intervals of $15 \times 15 \times 10$ m³, respectively. For the forward calculation, a total of 3134 shots were collected, with the interval between shots was 120 m. There were 16,081 traces for each shot, and the interval between receivers was 30 m. The record of a single trace was 8m long, and the sampling interval was 4 ms. The RTM image calculation was in the shot domain, the imaging aperture was changed to test the efficiency of migration, and the imaging space grid for a single shot was changed from 96 \times 96 \times 250 to 450 \times 450 \times 250.

The hardware used to run the KPSDM is GPU/CPU clusters that consists of 3.40 GHz Intel (R) Xeon (R) E3-1231 v3 CPUs, which had 16 GB of DDR3 memory, andNVIDIA Tesla K40 GPUs; the architecture is Kepler, and the global memory was 12 GB; in total, it had 2880 cores.

Figure 10 shows a comparison between the computational efficiency of the CPU and the GPU, and the propagation steps of both were constantly 16,000. Figure 10a shows a comparison between the computing time of a single CPU and GPU, while Figure 10b shows a comparison between the computing time of a single GPU and dual GPUs. It was observed from this that the GPU had significantly higher computing efficiency than the CPU, and the dual GPU had about two-times higher computing efficiency than that of a single GPU, indicating that the application of a stream completely conceals the data transfer between GPUs.



Figure 10. Comparison between the computing efficiency of (**a**) a single CPU and a single GPU; (**b**) comparison between computing time of a single GPU and dual GPU.

When the GPU number are more than two, the image space can be evenly distributed to each GPU nodes, and the source and receiver fields can be calculated parallelly, but because there is chaos between each two GPU, it may need some time for data transfer; the time for data transfer is less than the main space finite difference computation. Thus, in this paper, we have used steams and P2P feature of GPU to conceal this time. Figure 11 is a computation time comparison when the imaging space is $384 \times 384 \times 384$; we can see that when more GPUs are used, the speedup can be nearly linear.

Figure 12 shows a slice (d) of the migration results of the original single-shot record (a) in a 3D salt model, and the data (b,c) modulated with the two offset directions, respectively, as well as the results (e, f) from their migration. Figure 13 shows a profile and some SOGs of it from the image calculation with the 3D salt model. It was found that in the 3D calculation, because the velocity model used in the migration was the correct velocity, the events in the SOGs were flat, which is the ideal status. If the velocity is lower or higher than the correct velocity model, the events will bend up or down, and it will affect the image quality of the seismic profile, that can be achieved by summing up all of the gathers in the same SOGs.



Figure 11. Computation time comparison when use different number of GPUs for the imaging space size is $384 \times 384 \times 384$ and time steps is 16,000.



Figure 12. Part of single-shot records, modulated results, and imaged depth slices in the calculation of 3D SOGs: (**a**) part of the original single-shot records; (**b**) part of the single-shot records modulated with transverse offset; (**c**) part of the single-shot records modulated with horizontal offset; (**d**) depth slice of the results from the migration of original single shots; (**e**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration of Figure 12b; (**f**) depth slice of the results from the migration slice sl



Figure 13. Imaging profile of a subline and three SOGs from the 3D salt model RTM migration. (**a**) is one of subline image, and (**b**) is three SOGs at the black line of (**a**).

3. Conclusions

We have described an RTM method for the calculation of SOGs based on attribute migration. This method has been shown to be simple in terms of computation and has high efficiency compared with current methods. Based on the computing intensity demands required by the RTM calculation itself, we designed a strategy to accelerate the calculation using multiple GPUs and GPU features, such as stream and P2P data transmission between GPUs, to calculate the image space by dividing it into different blocks, while concealing the data transmission process between GPUs so that the efficiency improvement of multi-GPU computing was linearly related to the number of GPUs. Additionally, the calculation using multiple GPUs also solves the problems that occur when the image space is too large and the memory of a single GPU is insufficient. The 2D Marmousi model and the 3D salt model were applied to demonstrate the correctness and efficiency of the method in the calculation used to output SOGs. In this paper, we did not introduce the usage of the SOGs, but it can be used to update the velocity model, such as the Kirchhoff-based SOG using methods such as tomography or residual velocity analysis. Furthermore, for the GPU implement of RTM SOG calculation, we compared the efficiency with our CPU version, which may be not the best solution. We will continue this work on new GPU hardware and new CUDA features.

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

Funding: This research was funded by Sinopec external assistance plan (Grant No.3-4-2020-025).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: We thank the reviewers for their comments and opinions.

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

References

- 1. Yilmaz, Ö.Z. Seismic Data Analysis, Processing, Inversion and Interpretation of Seismic Data; Society of Exploration Geophysicists: Tulsa, OK, USA, 2001. [CrossRef]
- 2. Bednar, J.B. A brief history of seismic migration. *Geophysics* 2005, 70, 3MJ–20MJ. [CrossRef]
- Ke, X.; Shi, Y. Forward simulation and reverse time migration imaging based on one step wavefield extrapolation. *Chin. J. Geophys.* 2017, 60, 4468–4479. [CrossRef]
- 4. Sava, P.C.; Fomel, S. Angle-domain common gathers by wavefield continuation methods. *Geophysics* 2003, *68*, 1065–1074. [CrossRef]
- Sirgue, J.; Jousselin, P.; Audebert, F. Understanding wave-equation images gathers: A prerequisite for advanced post-processing and velocity model building. In Proceedings of the 72nd EAGE Conference and Exhibition, Barcelona, Spain, 14 June 2010; European Association of Geoscientists & Engineers: Houten, The Netherlands, 2010; pp. 437–441. [CrossRef]
- 6. Yang, Z.P.; Huang, S.T.; Yan, R. Improved subsalt tomography using reverse time migration surface offset gathers. In *SEG Technical Program Expanded Abstracts*; Society of Exploration Geophysicists: Tulsa, USA, OK, 2015; pp. 5254–5258. [CrossRef]
- Zhao, Y.; Zhang, H.Z. Cost reduction of reverse time migration surface offset gathers by local coherent events based interpolation. In SEG Technical Program Expanded Abstracts; Society of Exploration Geophysicists: Tulsa, OK, USA, 2017; pp. 4589–4593. [CrossRef]
- 8. Ehinger, A.; Lailly, P.; Marfurt, K.J. Greens function implementation of common-offset, wave-equation migration. *Geophysics* **1996**, *61*, 1813–1821. [CrossRef]
- 9. Etgen, J.T. 3D wave equation Kirchhoff migration. In *SEG Technical Program Expanded Abstracts;* Society of Exploration Geophysicists: Tulsa, OK, USA, 2012. [CrossRef]
- 10. Bleistein, N. On the imaging of reflectors in the earth. *Geophysics* 1987, 52, 931–942. [CrossRef]

- 11. Nguyen, S.; Baina, R.; Alerini, M.; Lambaré, G.; Devaux, V.; Noble, M. Stereotomography assisted by migration of attributes. *Geophys. Prospect.* 2008, *56*, 613–625. [CrossRef]
- Giboli, M.; Baina, R.; Nicoletis, L.; Duquet, B. Reverse time migration surface offset gathers: Part 1-A new method to produce "classical" common image gathers. In SEG Technical Program Expanded Abstracts; Society of Exploration Geophysicists: Tulsa, OK, USA, 2012. [CrossRef]
- 13. Lemaistre, L.; Lencrerot, R.; Prat, P.D.; Jousselin, P.; Gomes, C. Reverse time migration surface offset gathers part 2: Using the full power of wide azimuth information. In *SEG Technical Program Expanded Abstracts*; Society of Exploration Geophysicists: Tulsa, OK, USA, 2012. [CrossRef]
- 14. Cheng, J.; Grossman, M.; McKercher, T. *Professional CUDA C Programming*; Wiley & Sons, Inc.: Indianapolis, IN, USA, 2014; pp. 71, 73, 74, 268, 391.
- 15. Liu, H.W.; Li, B.; Liu, H.; Tong, X.L.; Liu, Q.; Wang, X.W.; Liu, W.Q. The issue of prestack reverse time migration and solutions with Graphic Processing Unit implementation. *Geophys. Prospect.* **2012**, *60*, 906–918. [CrossRef]
- 16. Liu, G.F.; Li, C. Practical implementation of prestack Kirchhoff time migration on a general purpose graphic processing unit. *Acta Geophys.* **2016**, *64*, 1051–1063. [CrossRef]
- 17. Liu, L.R.; Ding, W.; Liu, H.W.; Liu, H. 3D hybrid-domain full waveform inversion on GPU. *Comput. Geosci.* 2015, 83, 27–36. [CrossRef]
- 18. Liu, G.F.; Liu, Y.N.; Ren, L.; Meng, X.H. 3D seismic reverse time migration on GPGPU. Comput. Geosci. 2013, 59, 17–23. [CrossRef]
- Clap, R.G. Reverse time migration with random boundaries. In SEG Technical Program Expanded Abstracts; Society of Exploration Geophysicists: Tulsa, OK, USA, 2009; pp. 3235–3239.
- 20. Liu, G.F.; Li, C. A practical implementation of 3D TTI reverse time migration with multi-GPUs. In SEG Technical Program Expanded Abstracts; Society of Exploration Geophysicists: Dallas, TX, USA, 2016; pp. 4440–4444.
- Wang, Y.F.; Zhou, H.; Zhao, X.B.; Zhang, Q.C.; Zhao, P.R.; Yu, X.C.; Chen, Y.K. CuQ-RTM: A CUDA-based code package for stable and efficient Q-compensated reverse time migration. *Geophysics* 2019, 84, 1JF–Z5. [CrossRef]
- 22. Available online: https://developer.nvidia.com/blog/reducing-costs-with-one-pass-reverse-time-migration (accessed on 27 May 2021).
- 23. Nakata, N.; Tsuji, T.; Matsuoka, T. Acceleration of computation speed for elastic wave simulation using a graphics processing unit. *Explor. Geophys.* **2011**, *42*, 98–104. [CrossRef]
- 24. Schuster, G. Reverse-time migration = generalized diffraction stack migration. In *SEG Technical Program Expanded Abstracts;* Society of Exploration Geophysicists: Tulsa, OK, USA, 2002; pp. 1280–1283. [CrossRef]
- 25. Micikevicius, P. 3D Finite difference computation on GPUs using CUDA. In Proceedings of the 2nd Workshop on General Purpose Processing on Graphics Processing Units, GPGPU-2, Washington, WA, USA, 8 March 2009; Association for Computing Machinery: New York, NY, USA, 2009; pp. 79–84. [CrossRef]