energies

Article

A Method for Enhancing the Simulation Continuity of the Snesim Algorithm in 2D Using Multiple Search Trees

Chuanyou Zhou [1,*], Yongming He [1], Lu Wang [1], Shaohua Li [2], Siyu Yu [2], Yisheng Liu [1] and Wei Dong [1]

[1] College of Energy, Chengdu University of Technology, Chengdu 610059, China; heym@cdut.edu.cn (Y.H.)
[2] School of Geosciences, Yangtze University, Wuhan 430100, China
* Correspondence: 201772258@yangtzeu.edu.cn

Abstract: Multiple-point geostatistics (MPS) has more advantages than two-point geostatistics in reproducing the continuity of geobodies in subsurface reservoir modeling. For fluvial reservoir modeling, the more continuous a channel, the more consistent it is with geological knowledge in general, and fluvial continuity is also of paramount importance when simulating fluid flow. Based on the pixel-based MPS algorithm Snesim, this study proposes a method that utilizes multiple search trees (MSTs) to enhance simulation continuity in 2D fluvial reservoir modeling. The objective of the MST method is to capture complete data events from a training image (TI), which aims to achieve enhanced continuity in fluvial reservoir sublayer modeling. By resorting to search neighborhoods based on their proximity to the central node of the data template, multiple data templates that correspond to the MSTs will be generated. Here, four data templates were generated by arranging the relative search neighborhood coordinates in ascending and descending order with respect to the central node. Parallel computing was tried for the construction of the search trees. This work calculated the conditional probability distribution function (CPDF) of the simulating nodes by averaging the CPDFs derived from the MSTs, and double retrieval was employed to filter out the search trees that possessed an inaccurate local CPDF for the simulating nodes. In addition, the connected component labeling (CCL) method was introduced to evaluate the simulation continuity in MPS. The results indicated that the MST method can enhance the simulation continuity of the Snesim algorithm by reproducing the fine connectivity of channel facies in 2D fluvial reservoir modeling.

Keywords: multipoint geostatistical modeling; Snesim; search tree; conditional probability distribution function; simulation continuity

check for updates

Citation: Zhou, C.; He, Y.; Wang, L.; Li, S.; Yu, S.; Liu, Y.; Dong, W. A Method for Enhancing the Simulation Continuity of the Snesim Algorithm in 2D Using Multiple Search Trees. *Energies* **2024**, *17*, 1022. https://doi.org/10.3390/en17051022

Academic Editor: Philippe Leclère

Received: 30 December 2023
Revised: 13 February 2024
Accepted: 18 February 2024
Published: 22 February 2024

1. Introduction

Although there are advanced detection technologies such as seismic and well log analyses [1–3], the observation of subsurface geology remains challenging. Using geology data, researchers established geology models primarily based on the statistical method, which produced geostatistics. Traditional geostatistics is known as two-point geostatistics based on variogram algorithms, but the inability of the sequential Gaussian simulation and sequential indicator simulation (Sisim) to characterize complex geological structures and continuity reflects the limitations of traditional geostatistics [4–6]. Later, TI-based MPS was proposed as a solution to this challenge.

Guardiano and Srivastava [7] presented the first MPS algorithm, ENESIM, which captured patterns from a TI and derived probability distributions for the unknown nodes to be simulated. Using the ENESIM algorithm, the simulations of dune morphology, pore geometry, and fracture geometry were all accurately carried out. Due to rescanning of the TI for the simulating nodes, the ENESIM algorithm based on brute force is computationally intensive. Strebelle [8] presented the single normal equation simulation (Snesim) algorithm, which utilizes a search tree to store the CPDF inferred from a TI and requires only a moderate memory capacity for a single scan of the TI. In addition, the implementation of multiple grids conserves memory and CPU time. For the first time ever, the Snesim algorithm enabled the practical implementation of MPS for real reservoirs [9,10].

MPS has gained prominence in stochastic geological modeling following the proposal of the Snesim algorithm. Some MPS algorithms are based on patterns, like SIMPAT [10], FILTERSIM [11], DISPAT [12], and CIQ [13], while others are based on pixels, like DS [14], GOSIM [15], PCTO-SIM [16], and ASR3DRCS [17]. Liu [18] performed comprehensive sensitivity analyses on significant input parameters, thereby providing more in-depth guidance for understanding the Snesim algorithm. Bastante et al. [19] noted that Snesim produced simulations with a higher degree of connectivity between points than Sisim, which was more apparent when working with nonpoint supports. Regarding complex TIs, Boucher [20] proposed the search tree partitioning method, which made TI selection more flexible and accelerated the simulations. Hajizadeh et al. [21] reconstructed 3D pore space from a 2D TI using Snesim, which reproduced long-range connectivity and captured the characteristics of anisotropy in both horizontal and vertical dimensions. Then, Wu et al. [22] reconstructed 3D pore space from a 3D TI using Snesim, which was consistent with the X-ray computed tomography-scanned model. Huang et al. [23] accelerated the node-level simulation of Snesim by utilizing the GPU many-core architecture and parallel operation, while Cui et al. [24] designed and implemented a hybrid parallel framework that later successfully resolved a large-scale and high-resolution simulation with the DS method. Strebelle and Cavelius [25] enhanced Snesim by adding intermediate sub-grids and devising a new data template, allowing the user to resolve memory and performance issues. Combining pre-stack seismic inversion, Bayesian classification, and Snesim, Babu et al. [26] constructed an enhanced lithofacies model that reproduced the continuity of lithofacies with reliability.

For conditional simulations involving relatively dense datasets, MPS proves more favorable than two-point geostatistics and object-based modeling in simulating channel continuity [27]. Generally, pattern-based algorithms excel in reproducing simulation continuity compared to pixel-based algorithms [11,28], whereas pixel-based algorithms outperform pattern-based algorithms in honoring conditional data [25,29]. Taking into account conditional data in practice, this study focuses on increasing simulation continuity by employing the first practical MPS method, i.e., the pixel-based Snesim algorithm. The more continuous a channel is, the better it aligns with general geological knowledge [30]. Moreover, channel continuity is crucial for fluid flow simulation applications [31]. The simulation implementation of Snesim is dependent on parameter settings such as template size, multiple-grid number, etc. Wang et al. [32] investigated parameter optimization to determine the optimal number of search template nodes in Snesim based on a GLCM [33,34] and the deep learning method [35], which provided the most reasonable parameter to build the most continuous channel facies. This study presents a novel approach for determining the Snesim simulation parameters; however, it does not enhance the Snesim algorithm itself. In a recent study, Babu et al. [26] suggested using Snesim to characterize lithofacies to enhance the representation of geological facies continuity in meandering fluvial sandstone reservoirs. The reported seven groups of comparisons showed that in one or two small places, the simulation continuity of the Snesim realizations was slightly better than Bayesian classification. This approach integrated pre-stack seismic inversion, Bayesian classification, and Snesim and was a workflow for practical applications rather than an improvement of the Snesim algorithm. Shahraeeni [31] presented a robust algorithmic framework for generating continuous channels based on Snesim. During the simulation procedure, it monitored for dead-end pixels causing channels to be terminated. If dead-end pixels occurred or were anticipated, retrace would be performed. Clearly, this method of constant retracing increases the computing requirements significantly. Kumar and Srinivasan [36] developed the InDA-MPS approach, primarily based on InDA-Snesim, to enhance channel continuity through the incorporation of secondary production data. The selection of the production data poses a critical challenge due to the time-varying nature of the involved variables, such as fluid flow rates and wellhead values. It is increasingly clear that generative adversarial networks

(GANs) can substantially aid in geological modeling [37,38]. Recently, Zheng et al. [39] employed GANs to estimate hydraulic conductivity in non-Gaussian groundwater fields using a training image. This approach demonstrated the high efficiency of GANs in producing geological models compared to Snesim. However, there is a notable issue, as GANs tend to replicate the training image without taking conditional data into account.

Work to improve the simulation continuity of Snesim is still ongoing. In order to improve channel continuity in fluvial reservoir sublayer modeling, this study proposes a method for enhancing the simulation continuity of the Snesim algorithm using MSTs. The MST method provides an advantage by enhancing the accuracy of the local CPDFs. Firstly, this paper introduces the MST method, subsequently utilizing it to compare the simulation outcomes with those achieved using the original Snesim algorithm, through both theoretical and practical simulations. Parallel computing was tested for search tree construction during the simulations. Additionally, this study explores double retrieving for enhancing the precision of the local CPDFs and evaluates simulation continuity using CCL analysis. Finally, a summary of this work is provided, along with future research topics.

2. Methodology Based on Multiple Search Trees

Strebelle [8] proposed the Snesim algorithm because two-point geostatistical simulation could not reproduce the different shapes and sizes of complex geological bodies. Figure 1 depicts the flowchart of the Snesim algorithm. According to the flowchart, the Snesim algorithm will be reviewed briefly here, based on several important factors. A TI is an a priori geological mode that reflects the actual subsurface reservoir characteristics. A data template is a collection of ordered nodes used to scan a TI, and every replicate found in the TI is referred to as a data event. The search tree is a dynamic data structure that facilitates the retrieval of all data events. The TI, data template, and data event are essential in the establishment of a search tree, depicted in Figure 2. Liu [18] provided a more detailed description of the input parameters for Snesim. The utilization of a search tree is fundamental to the implementation of the Snesim algorithm based on Monte Carlo sampling. A search tree precisely implements the dynamic storage of data events, significantly saving computing time and decreasing the memory demand. In fact, the establishment of a search tree is intricately linked to the characteristics of the data template, specifically, its size and sequence settings. Notably, the size of the data template determines the levels of the search tree, whereas the sequence of the data template determines the expansion order of the search tree. Based on the information provided in Figure 2, a search tree can be uniquely determined when both the size and the sequence are known. In this paper, the search tree in the original Snesim algorithm is referred to as a single tree.

Observations indicated that the data events collected by a single tree are not extensive enough to fully capture the geological–spatial relationship. According to Figure 2, as the search tree expands from the root node (level 0) to level 1, the first child node corresponds to No. 1 in the data template sequence. Similarly, when the search tree further expands from level 1 to level 2, the second child node corresponds to No. 2 in the sequence. Clearly, the growth of the search tree disregards the fact that every child node is equivalent, as the distance between the search neighborhoods and the central node u is identical in the data template. In other words, the four nodes have the same spatial relationship to the root node. This growth mode results in the loss of several data events when only one single tree is considered. Boucher [20] also noted that the majority of data events are insufficient. Figure 3 illustrates the data events not captured at level 1 of the search tree in Figure 2. Each data event is accompanied by a number, indicating the quantity of missed occurrences. In the original Snesim algorithm, Strebelle [8] derived the probability for the simulating node whose CPDF cannot be inferred directly from the data events by combining CPDFs, as shown in Figure 4. Here, each data event is accompanied by a number, indicating the quantity of observed occurrences. The probability, P(u = green | 2 = green) = 1/3 can be deduced from the combination of CPDFs by expanding the neighboring nodes around the simulating node in the absence of specific data events. However, if the sequence of the data template in Figure 2 makes an equivalent transformation by exchanging node 1 and node 2, the CPDF can be directly inferred from the data events in the new search tree as P(u = green | 1 = green) = 2/5, which is not equal to the value (1/3) obtained with the CPDF combination approach in the original Snesim algorithm.

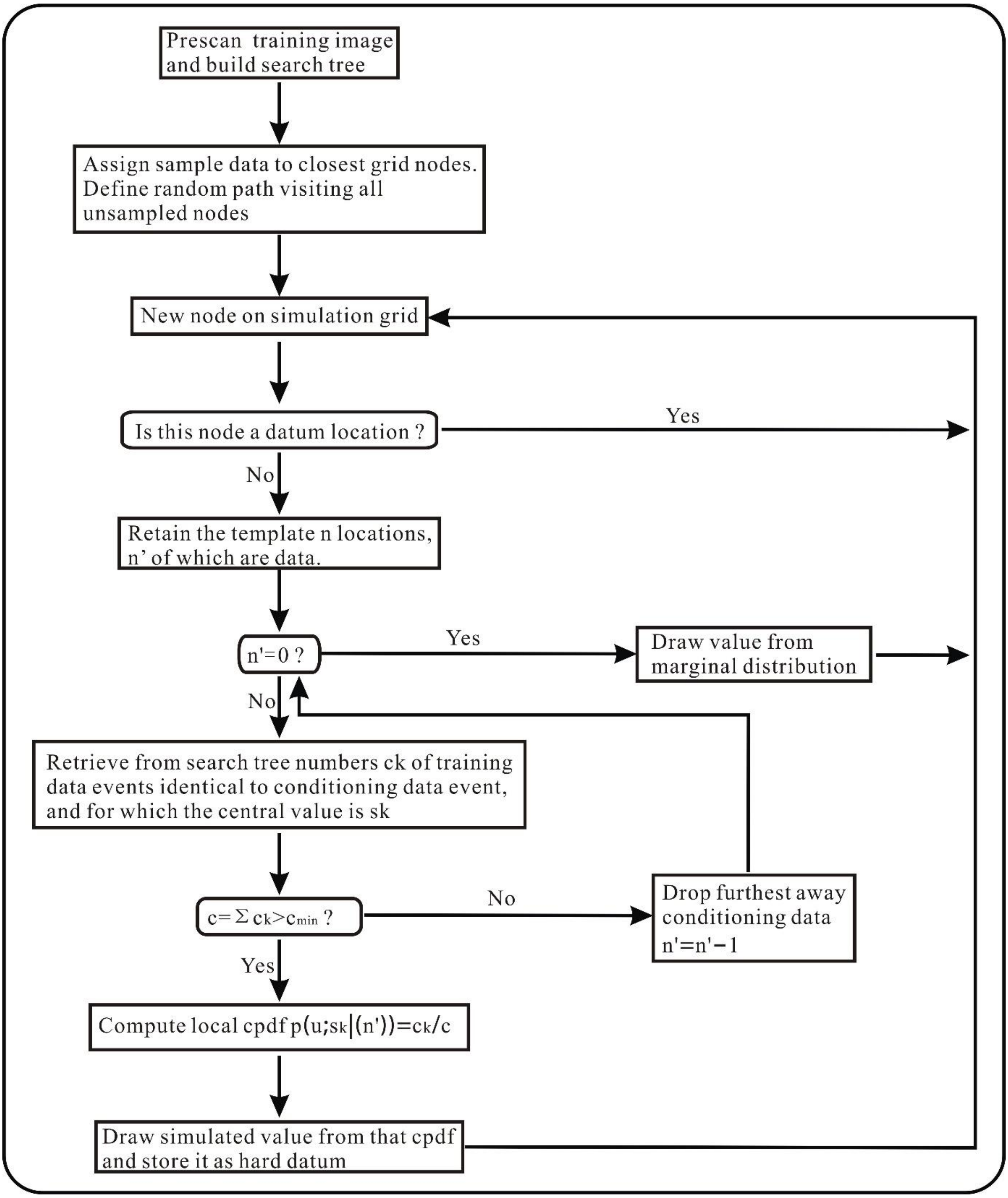

Prescan training image and build search tree
Assign sample data to closest grid nodes. Define random path visiting all unsampled nodes
New node on simulation grid
Is this node a datum location ?
Yes
No
Retain the template n locations, n' of which are data.
n'=0 ?
Yes
Draw value from marginal distribution
No
Retrieve from search tree numbers ck of training data events identical to conditioning data event, and for which the central value is sk
c=Σ ck>cmin ?
No
Drop furthest away conditioning data n'=n'−1
Yes
Compute local cpdf p(u;sk|(n'))=ck/c
Draw simulated value from that cpdf and store it as hard datum

Figure 1. Flowchart of the Snesim algorithm [8].

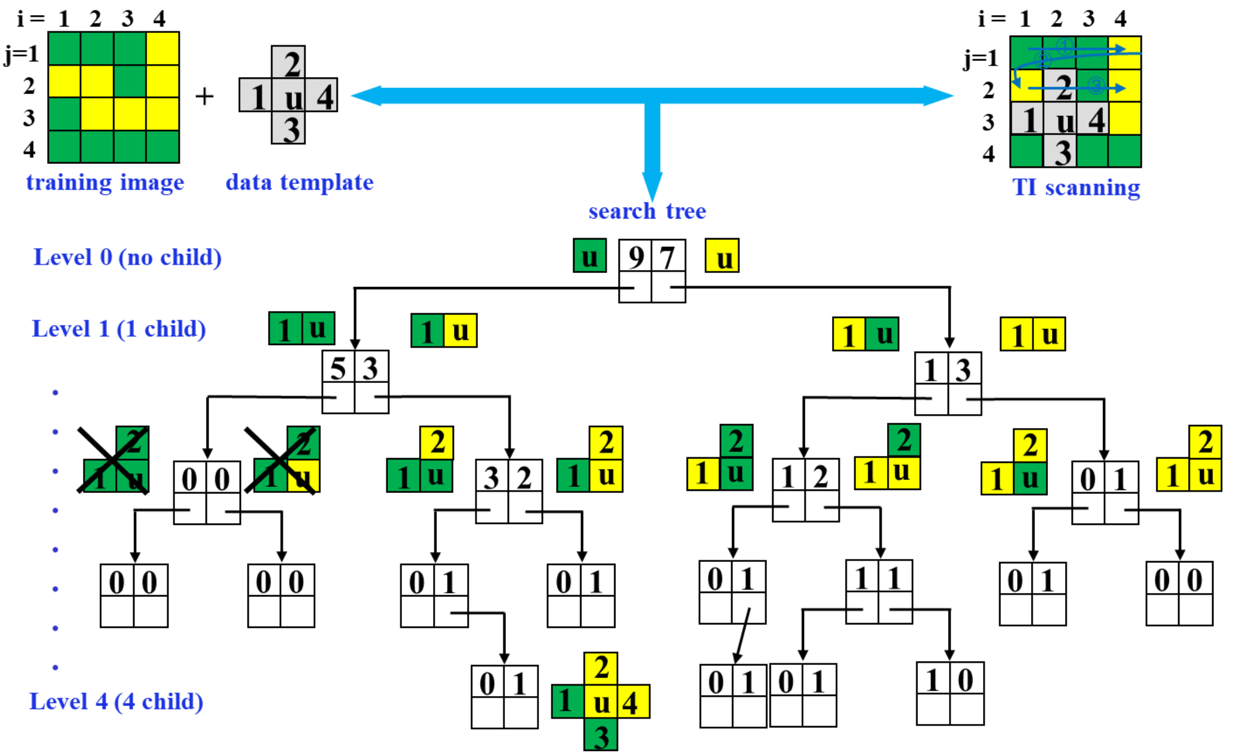


Figure 2. Establishment of a search tree (the yellow square represents sand, while the green square signifies mud).

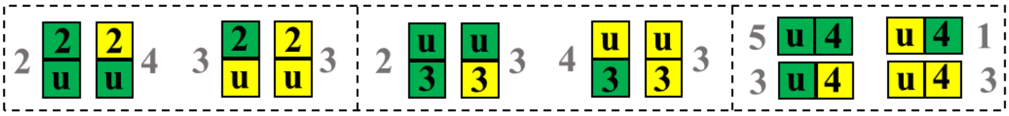


Figure 3. Data events missed at the first level shown in Figure 2.

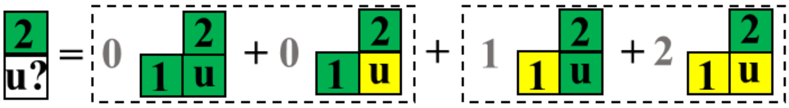


Figure 4. CPDF combination approach by expanding the neighboring nodes.

By repositioning the equivalent nodes within the data template, the generation of diverse yet equivalent data events is facilitated. In fact, for the determined size of the data template depicted in Figure 2, a total of 4! = 24 search trees are required to exhaustively retrieve all data events through the sequence transformation of the data template, which can be easily counted using permutations and combinations in mathematics. According to Figure 5, there are 21 nodes in the data template (including the central point u) where the neighbors of each cluster connected with a circle share the same distance from the central point u. The number of search neighborhoods is properly organized as $N_1 = N_2 = N_3 = 4$ (equivalent nodes 1–4 constituting N_1; equivalent nodes 5–8 constituting N_2; equivalent nodes 9–12 constituting N_3) and $N_4 = 8$ (equivalent nodes 13–20 constituting N_4). Theoretically, a total of $4! \times 4! \times 4! \times 8! = 557{,}383{,}680$ search trees are required to collect all the data events. Starting from the center and going outward, the number of equivalent nodes $N_1, N_2, \ldots, N_n$ is ordered based on the distances between the search neighborhoods and the central point u in the data template. From the theoretical calculation, Formula (1) can be used to ascertain the total number of search trees by scanning the TI independently using equivalent data templates. The mathematical analysis described above forms the foundation of the MST method.

$$N_t = \prod_{i=1}^{n} N_i! \quad (1)$$

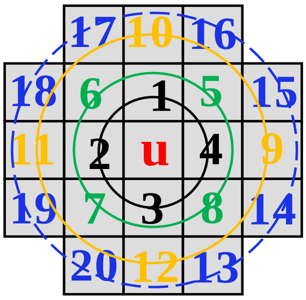


Figure 5. Search neighborhood clustering.

The MST method can improve the exhaustive retrieval of data events from TI scans using equivalent templates, resulting in more accurate geostatistical simulations. To achieve better simulation continuity of geobodies in the original Snesim algorithm, it is customary to use a larger data template or a multiple-grid approach [40]. However, as the size of the data template increases, the computation of the CPDF from the search tree becomes more time-consuming, and additional memory is required to construct the search tree [25]. From the perspective of geological–spatial relationships, it is evident that the spatial correlation diminishes as the distance between the conditional nodes and the simulating node increases, which is analogous to the effect of the lag distance in variogram analysis. Therefore, selecting a large data template without specific criteria will introduce extra geological constraints into the simulation without any spatial correlation, while using multiple grids (more than four) will cause the same issue as choosing a large data template. This study introduces the MST method to enhance simulation continuity in 2D space. Due to the potentially large number of N_t, as indicated in Formula (1), it is impractical to create many trees to collect all data events. In this work, four search trees were employed in the MST method.

Hansen used the i and j indices in the open source code mGstat (https://github.com/cultpenguin/mGstat, accessed on 4 August 2015) to locate the search neighborhoods (Figure 6a) relative to the central point u in the data template. For example, (1, 1) is located at the '+1' position relative to u in both the x and y directions, while (1, 1), (1, −1), (−1, 1), and (−1, −1) have the same distance to u. Here, the indices i and j are arranged in ascending and descending order using permutations and combinations. To provide a clearer understanding, i↑&j↑, i↑&j↓, i↓&j↑, and i↓&j↓ are utilized to sort the search neighborhood sequence based on the distance d (Appendix A Equation (A1)). This approach facilitates the establishment of a manageable number of search trees. The implementation of this function is displayed in Appendix A Listing A1. This simplified approach can generate four equivalent data templates (Figure 6b–e), which correspond to four search trees. This study calculated the CPDF of the simulating nodes by averaging the CPDFs derived from the four search trees, as depicted in formula (2) and Figure 7. The following sections will present a series of tests conducted on multiple ($N_t \le 4$, referring to Section 4.1) search trees, enhancing the simulation continuity of geobodies.

$$\mathrm{cpdf}_{avg} = \left[\sum_{i=1}^{N_t} \mathrm{cpdf}_i\right] / N_t \quad (2)$$

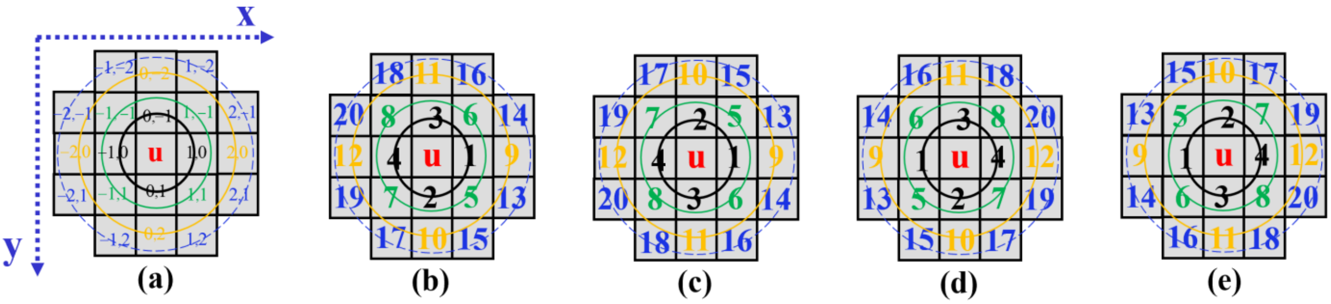


Figure 6. Search neighborhood sorting (**a**) with i and j indices relative to the central point u, (**b**) with i↑&j↑, (**c**) i↑&j↓, (**d**) i↓&j↑, and (**e**) i↓&j↓.

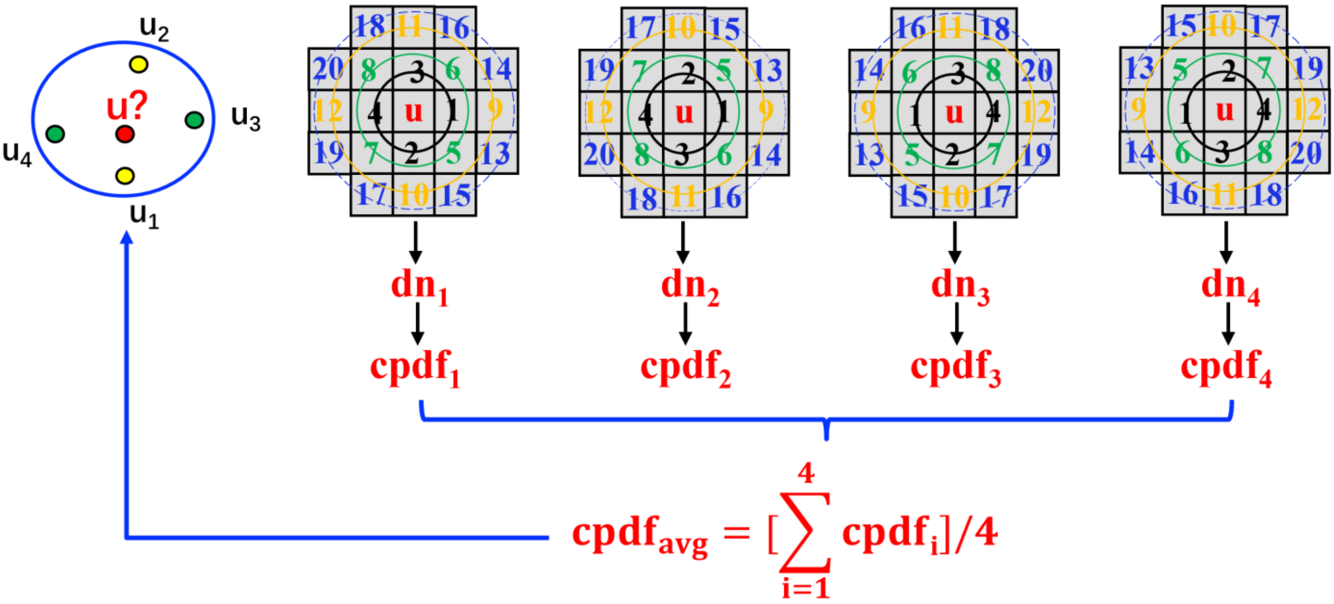


Figure 7. Jointly averaging the CPDFs of the 4 search trees, dn representing data event.

3. Results

3.1. Unconditional Simulation

Figure 8 depicts the TI used in this study, which consists of 125 × 125 = 15,625 pixels and characterizes a horizontal 2D section of a fluvial reservoir containing sand-filled channels against a mudstone background. Here, the sand/mud ratio was set to 0.38 for the simulation. The data template contained 81 nodes. The simulation was performed on the finest grid to easily identify the effect that MSTs had on enhancing the simulation continuity of the Snesim algorithm. The simulation work area measured 60 × 60 pixels. The parameter settings, including TI, data template, and so on, were consistent between the MST method and the original Snesim.

Figure 9 illustrates the unconditional simulation realizations of the Snesim algorithm, including the realizations of four search trees and a single tree. The implementation of the MST method (Figure 9a) enhanced the channel continuity in three regions (A, B, and C) compared to the single-tree simulation implementation (Figure 9b). By calculating the experimental variograms of the x and y directions as depicted in Figure 10, the spatial variability of the simulated results was quantitatively compared. The experimental variograms of the simulated realizations produced by both the MST and the original Snesim algorithm exhibited similar trends as the TI. For the most part, furthermore, they closely coincided in both the x and the y directions, as the MST method primarily restored specific local regions of the sand channel.

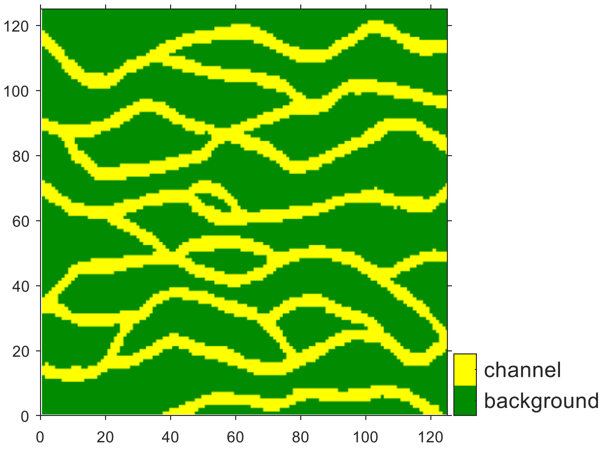
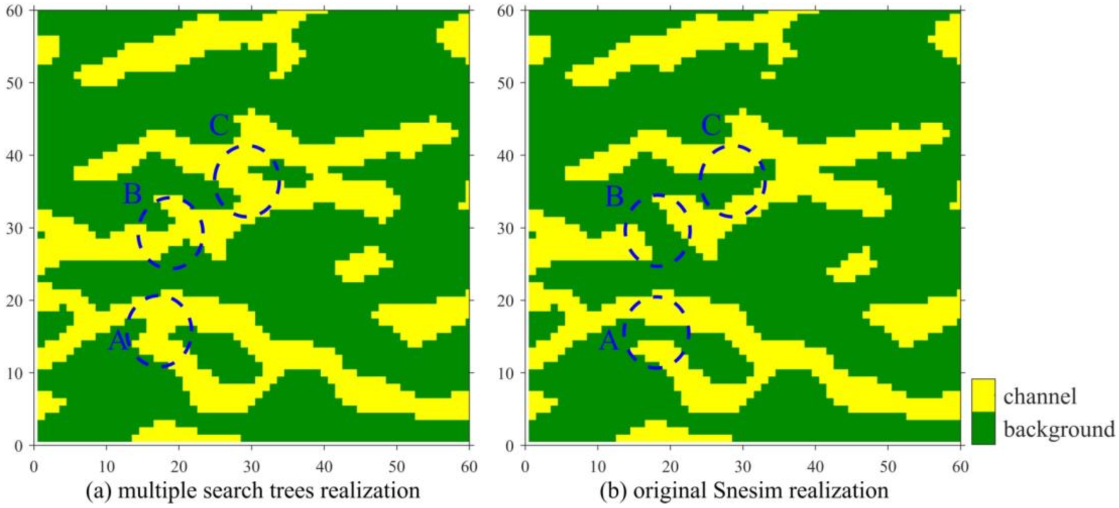


Figure 8. Training image.



(a) multiple search trees realization

(b) original Snesim realization

Figure 9. Unconditional simulation realizations.

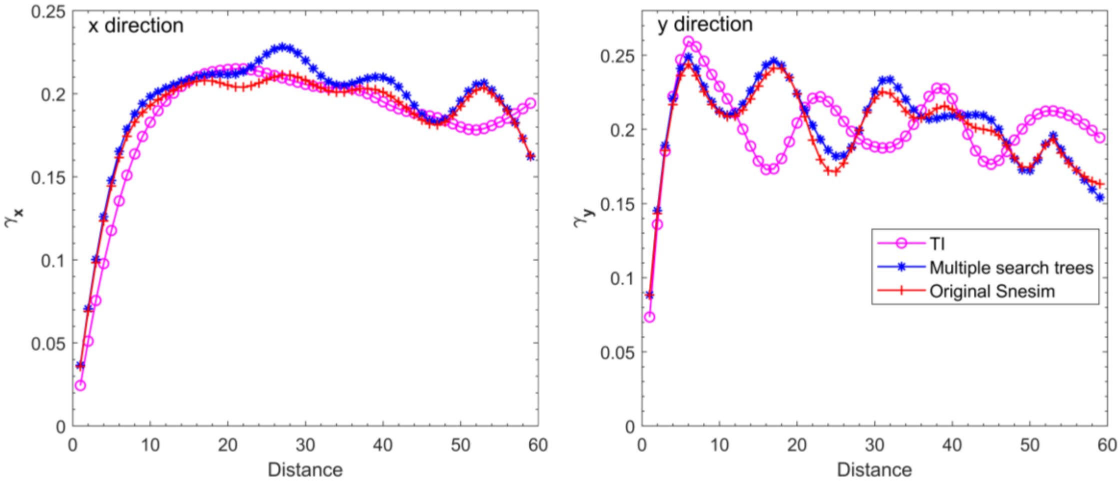


Figure 10. Variograms of the x and y directions in the unconditional simulation realizations.

3.2. Conditional Simulation

Figure 11a shows the produced and sampled conditioning data, showing a small portion of the produced geobody on the left and some of the sampling points generated by a random algorithm on the right. Other parameter settings were the same as those in the unconditional simulation. Figure 11b,c illustrates the conditional simulation realizations. The implementation of the MST method was found to enhance the channel continuity in a specific region A compared to the implementation of a single tree. The simulation continuity of the MST method was superior to that of the original Snesim algorithm after several runs. The experimental variograms in the x and y directions, depicted in Figure 12, also exhibited a similar pattern.

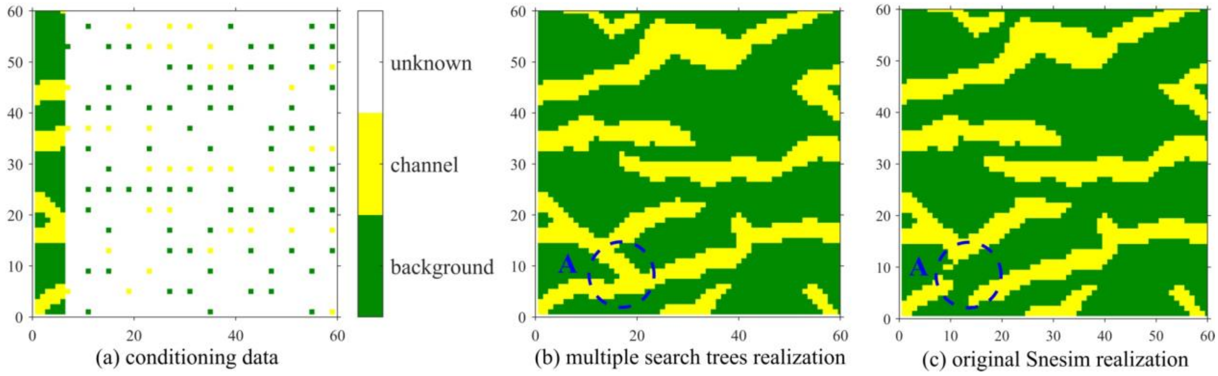

unknown
channel
background
A
A
(a) conditioning data
(b) multiple search trees realization
(c) original Snesim realization

Figure 11. Conditioning data and conditional simulation realizations.

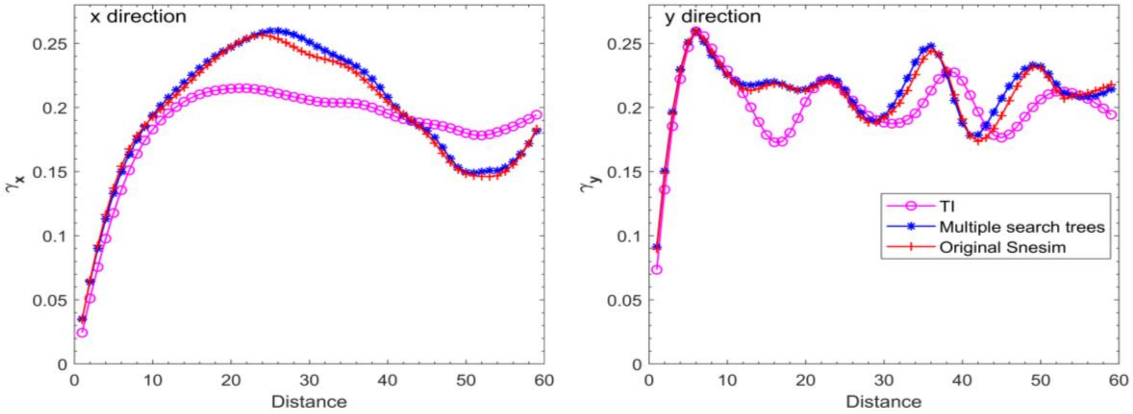


Figure 12. Variograms of the x and y directions in the conditional simulation realizations.

3.3. Practical Simulation

The preceding Sections 3.1 and 3.2 described the theoretical tests for the MST method; the Ng3^3 sublayer at one well site in the Zhongyi District of the Gudao oilfield served as an example for the practical geological simulation test. Near the well, the Ng3^3 sublayer covers nearly 4 km^2 in the study area, characterized by a meandering channel. The point bar, natural levee, and crevasse splay are distributed along the margin of the meandering channel. There are 85 wells in the Ng3^3 sublayer, which has an average sediment thickness of 7.1 m. From a geological perspective, the sublayer is characterized as a high-quality reservoir with an average porosity of 35.2% and a permeability of 12,173 mD. An analysis of the well data revealed that the point bar exhibited a greater thickness of sand compared to the channel, while the crevasse splay displayed a slightly thicker sand body than the natural levee. The thickness of the sand body exhibited significant variation across three distinct levels. To derive the overall sand body distribution within the control range of the channel, the point bar, natural levee, and crevasse splay were incorporated into the channel to construct the predominant sand body distribution.

Figure 13 depicts the TI selected for reservoir modeling, which is a stationary meandering channel with high curvature and a sand/mud ratio of approximately 1.0. As shown in Figure 14, the conditioning data were obtained from seismic data, core, logging, and virtual wells. Both the MST method and the original Snesim algorithm were utilized to simulate the distribution of facies in the Ng3^3 sublayer using the same parameter settings.

On the *X*-axis (from 6.0 km to 7.9 km) and *Y*-axis (from 3.0 km to 4.9 km), the Ng3^3 sublayer was divided into 99 and 94 pixelated grids, respectively. The data template sequentially incorporated multiple grids (G = 3) with 14, 22, and 25 nodes. Figure 15(A(a),A(b)) depicts the simulation realizations, and Figure 15(B(a),B(b)) depicts the smoothed outcomes. Based on the simulation results, it was evident that the MST realization (Figure 15(B(a))) successfully restored region M when compared to the original Snesim realization (Figure 15(B(b))). Both were more closely aligned with the meandering fluvial pattern than the realization presented in the Sisim simulation (Figure 15(B(c))). The experimental variograms in the x and y directions are displayed in Figure 16, also exhibiting a similar pattern, as discussed earlier in the Sections 3.1 and 3.2. Practice has demonstrated that the MST method better depicts the distribution of sand bodies in the Ng3^3 sublayer of the Gudao oilfield.

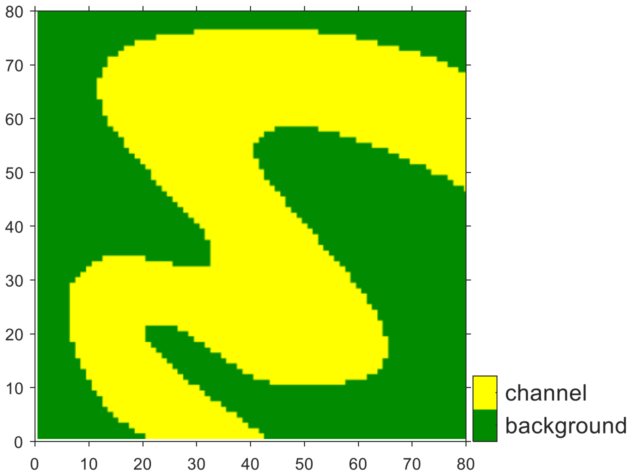


Figure 13. Training image of the Ng3^3 sublayer.

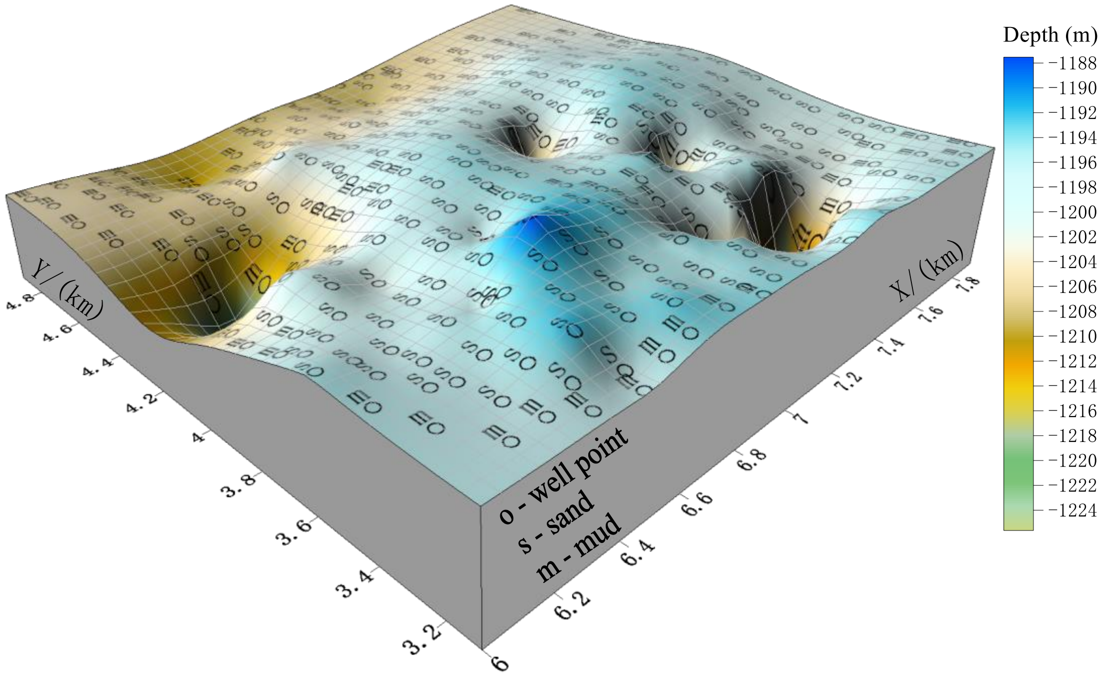
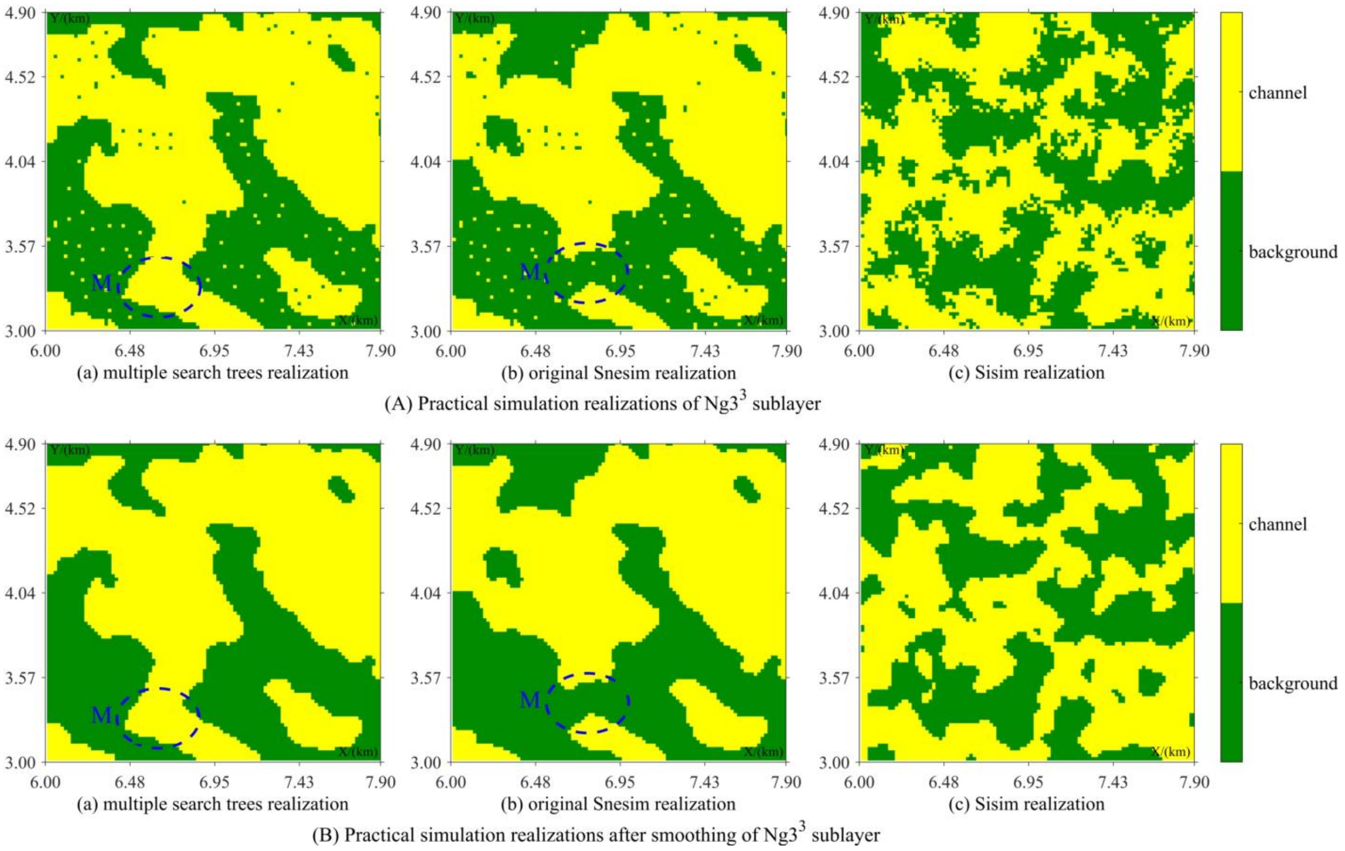


Figure 14. Conditioning data of the Ng3^3 sublayer.



(a) multiple search trees realization (b) original Snesim realization (c) Sisim realization

(A) Practical simulation realizations of Ng3^3 sublayer

(a) multiple search trees realization (b) original Snesim realization (c) Sisim realization

(B) Practical simulation realizations after smoothing of Ng3^3 sublayer

Figure 15. Practical simulation realizations in the Gudao oilfield.

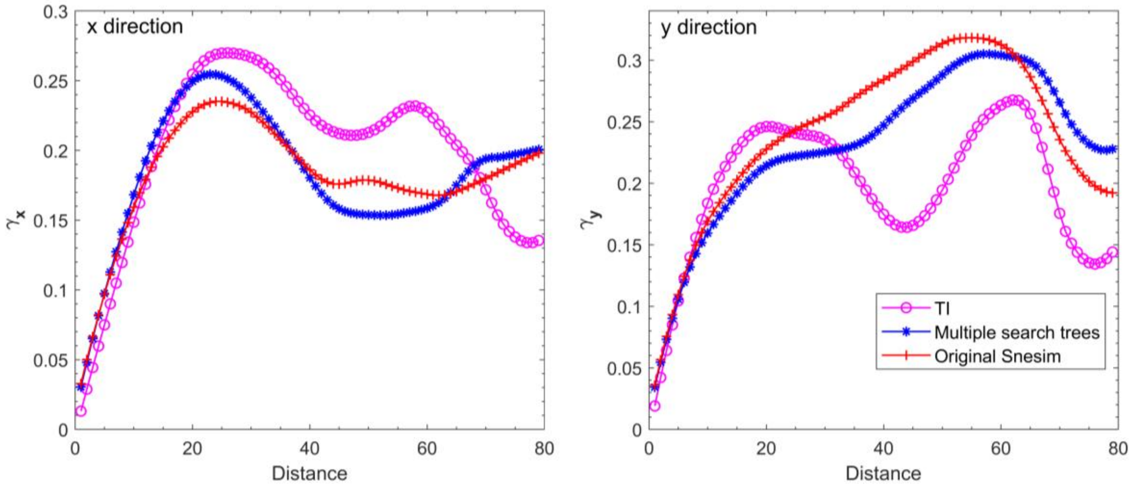


Figure 16. Variograms of the x and y directions in practical simulation realizations of the Gudao oilfield.

All the simulations were conducted on a laptop equipped with an Intel® Core$^{\text{TM}}$ i5-1035G1 CPU operating at 1.00 GHz, and 8 GB of RAM. The workflows and algorithms were implemented using the MATLAB R2023b programming language. The computation time and RAM consumption for each simulation are presented in Table 1. Based on the data presented in Table 1, it can be observed that the non-parallel MST method required a search tree construction time that was more than four times greater than that of the original Snesim algorithm in the unconditional simulation. This occurrence is not surprising because the MST method proposed in this study referred to four distinct search trees, in addition to handling some additional processing. To reduce the computation time, the parallel MST method utilizing four cores was employed, resulting in a moderate decrease in the processing time. However, it is possible that due to the relatively small size of the tasks, the overhead associated with initiating and managing the parallel pool may surpass the actual computation time of the tasks themselves. Consequently, parallelization did not yield significant performance improvements in this context. In both conditional simulation and practical simulation, the results were like those observed in the unconditional simulation. For the search tree construction procedure, both the non-parallel and the parallel MST methods necessitated greater RAM allocations than the original Snesim algorithm, with the parallel MST method, in particular, requiring significantly larger RAM. This increased RAM demand can be attributed to factors such as data duplication, overheads from managing multiple threads, and the need for synchronization mechanisms. Although parallel computing effectively reduces the execution time by utilizing multiple processors, it comes with a trade-off of heightened memory usage to accommodate the infrastructure needed for a parallel execution.

Table 1. Computation time and RAM usage of both theoretical and practical simulation tests.

Test	Search Tree Construction Time (s)	RAM (Random-Access Memory) (MB)
Unconditional simulation (non-parallel MST/parallel MST/original Snesim)	28.5/15.7/5.3	2675/4028/1711
Conditional simulation (non-parallel MST/parallel MST/original Snesim)	29.1/15.8/5.4	2667/4106/1169
Practical simulation (non-parallel MST/parallel MST/original Snesim)	7.7/4.4/1.8	1255/2870/1030

4. Discussion

4.1. Double Retrieving for Improving the Accuracy of Local CPDFs

The test results displayed above illustrate not only theoretical but also practical simulation implementations of the MST method. The method for enhancing the simulation continuity of the Snesim algorithm by utilizing MSTs is based on local CPDF precision. Strebelle [8] made considerable effort to implement a reasonable local CPDF in his Fortran 90 code. In addition to the multiple-grid approach, Strebelle simulated nodes whose CPDF could not be inferred directly from the search tree by using a combination of CPDFs (Figure 4) and dropping the farthest distant datum. When using the MST method to model geological bodies, if the local CPDF at each unsampled node is derived from more than one search tree simply by jointly averaging (Figure 7), the potential export of extraneous information poses a challenge. Consequently, this study employed a compromise strategy called double retrieving. Initially, the search trees containing the most conditioning data for the simulating node was retrieved from the four search trees, which was followed by an additional retrieval of the retrieved search trees. Finally, the number of search trees used for the simulation could be less than 4, which is why $N_t \leq 4$ is stated in Section 2.

The benefit of a double retrieval lies in the ability to effectively filter out search trees that possess inaccurate local CPDFs for the simulating nodes. Figure 17 illustrates the frequency of nodes simulated by the MST method after the double retrieval process. For the unconditional simulation in Section 3.1 and the conditional simulation in Section 3.2, the four search trees were retrieved for most of the simulated nodes, and from two to four search trees accounted for over 70% of the simulated nodes. Even though two search trees accounted for most of the nodes simulated in the practical simulation of the $Ng3^3$ sublayer, from two to four search trees still accounted for 71.7% of the total number of simulated nodes. The double retrieval approach significantly enhanced the precision of local CPDFs when simulating nodes.

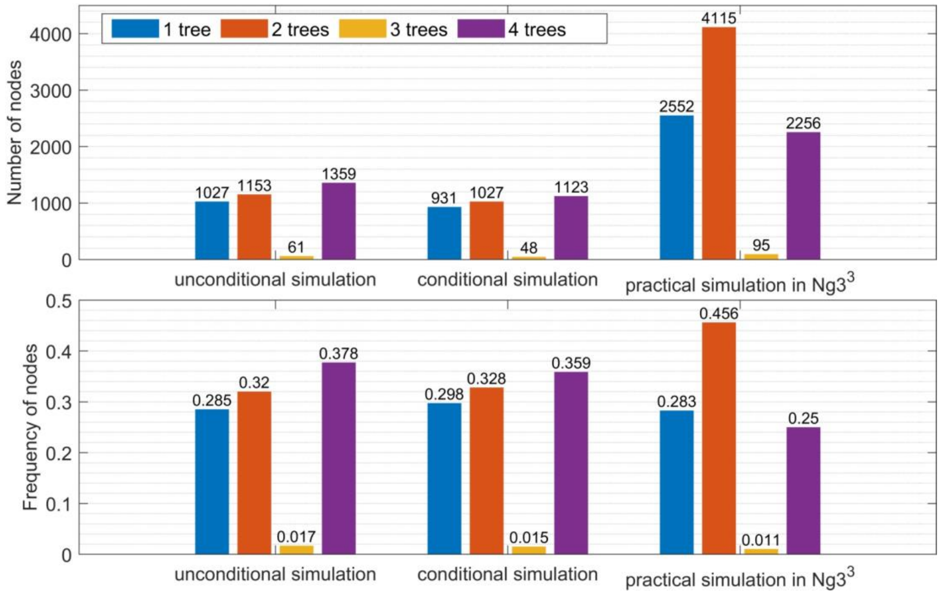
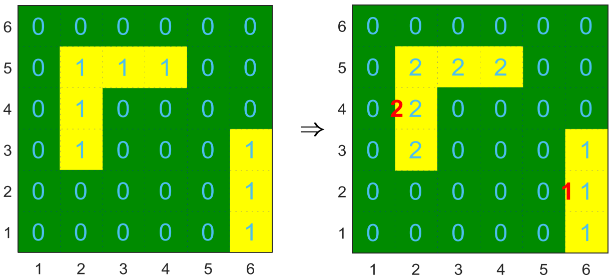


Figure 17. Frequency distribution of the nodes simulated by the MST method after double retrieving.

4.2. Assessing Simulation Continuity with CCL

The aforementioned test results (Figures 9, 11 and 15) were based on comparing a single pair of simulation realizations. To further facilitate a comprehensive comparison, a total of 600 pairs of realizations for each type of test were executed. The fluvial reservoir consisted of two facies, with 0 representing the mudstone background, and 1 representing sand-filled channels. Connected component labeling (CCL) has been widely used in image processing and computer vision, particularly for 2D binary images [41–43]. CCL is a technique for labeling each distinct connected component in a binary image with a unique identifier. The input and output using CCL are depicted in Figure 18.



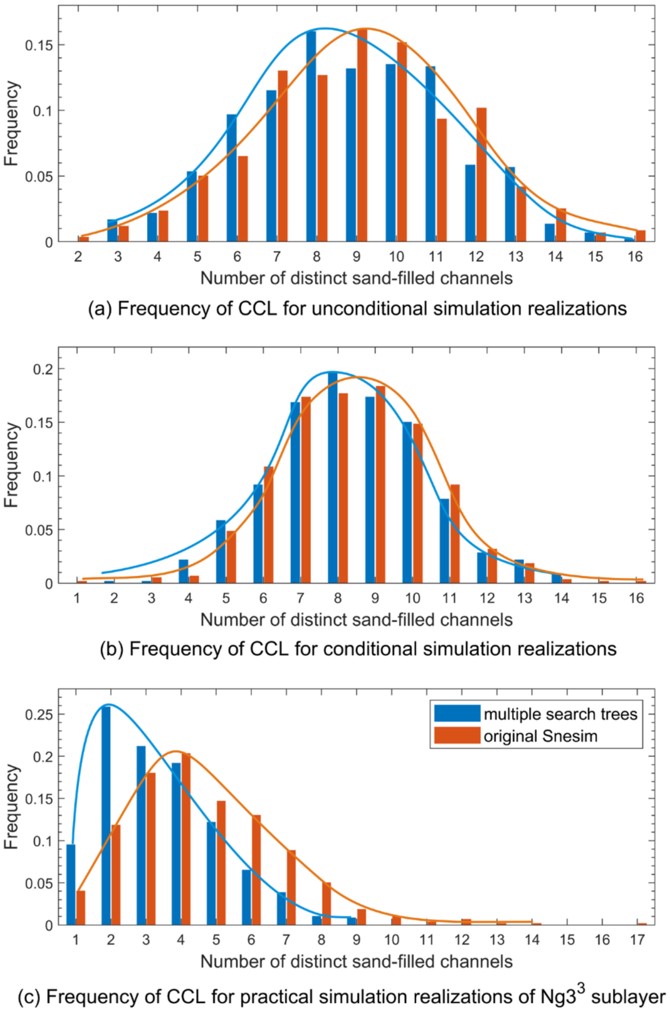
6	0	0	0	0	0	0
5	0	1	1	1	0	0
4	0	1	0	0	0	0
3	0	1	0	0	0	1
2	0	0	0	0	0	1
1	0	0	0	0	0	1
	1	2	3	4	5	6

⇒

6	0	0	0	0	0	0
5	0	2	2	2	0	0
4	0	2 2	0	0	0	0
3	0	2	0	0	0	1
2	0	0	0	0	0	1 1
1	0	0	0	0	0	1
	1	2	3	4	5	6

Figure 18. CCL operation (the green color signifies a mudstone background, yellow indicates sand-filled channels, and red numbers mark labels for identifying each distinct channel).

CCL could be used to precisely identify all the distinct sand-filled channels. As for pairs of realizations, the fewer the distinct sand-filled channels, the better the simulation continuity. For a quantitative comparison between the three types of tests conducted using the MST method and the original Snesim, the CCL algorithm was utilized to analyze 600 pairs of realizations in each type to precisely identify the frequency of unique sand-filled channels. Figure 19a–c depicts the frequency of distinct sand-filled channels for three types of tests using the MST method and the original Snesim. As observed in Figure 19, the three types of simulation realizations using the MST method were distributed to the left of the original Snesim implementation. Table 2 displays the most important distributional characteristics, indicating that the mean of the CCL numbers in the three types of simulation realizations using the MST method was lower than that of the original Snesim algorithm. The results demonstrated that the realizations by the MST method in both theoretical and practical simulations had a higher level of continuity than those obtained with the original Snesim algorithm. As mentioned previously, to compare the MST method with the original Snesim algorithm, a total of 600 pairs of realizations were executed for each type of test. The average experimental variograms for both the MST method and the original Snesim algorithm in theoretical and practical tests are displayed in Figure 20. Figure 21 illustrates the integration of the average experimental variograms obtained from both the MST method and the original Snesim algorithm, combined with that of the TI, in a single graph. The experimental variograms of the realizations calculated from both the MST method and the original Snesim algorithm exhibited similar trends to that of the TI in each type of simulation, highlighting that the MST method primarily restored specific local regions of the sand channels compared to the original Snesim algorithm.



(a) Frequency of CCL for unconditional simulation realizations

(b) Frequency of CCL for conditional simulation realizations

(c) Frequency of CCL for practical simulation realizations of Ng3^3 sublayer

Figure 19. Frequency of CCL for the theoretical and practical tests.

Table 2. CCL distribution characteristics of the theoretical and practical tests.

Test	**Distribution Characteristics**			
	Range [1]	**Mean** [2]	**Std** [3]	**N** [4]
Unconditional simulation (MST/original Snesim)	[3–16]/[2–16]	8.82/9.04	2.49/2.56	600/600
Conditional simulation (MST/original Snesim)	[2–14]/[1–16]	8.35/8.42	2.04/2.01	600/600
Practical simulation (MST/original Snesim)	[1–9]/[1–17]	3.44/4.62	1.70/2.21	600/600

[1] The numerical interval of the distribution of distinct sand-filled channels. [2] The mean of the probability distribution. [3] The standard deviation of the probability distribution. [4] The number of simulation realizations.

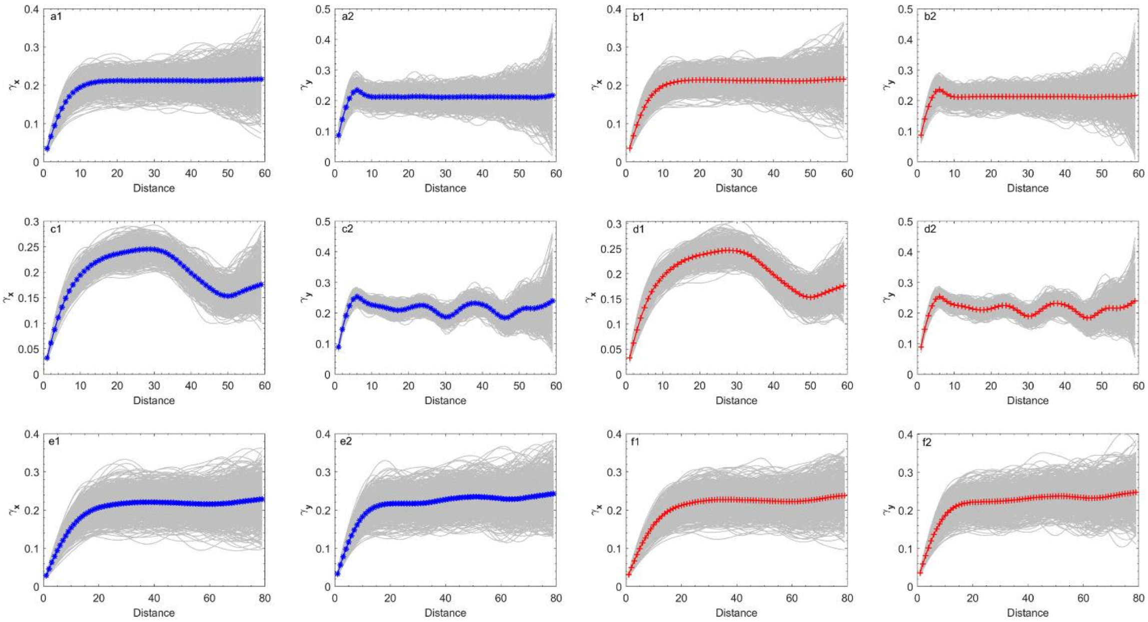


Figure 20. Average experimental variograms of the theoretical and practical tests. (**a1**) MST method for the unconditional simulation in the x direction, (**a2**) MST method for the unconditional simulation in the y direction, (**b1**) original Snesim algorithm for the unconditional simulation in the x direction, (**b2**) original Snesim algorithm for the unconditional simulation in the y direction, (**c1–d2**) series of conditional simulations, (**e1**–**f2**) series of practical simulations.

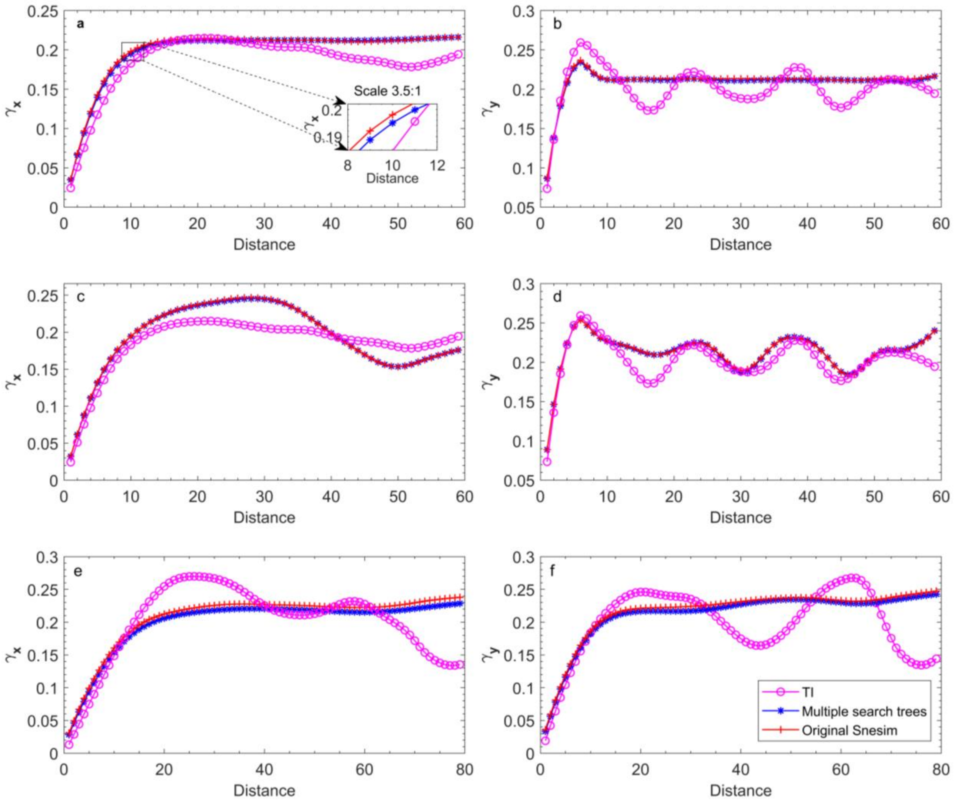


Figure 21. Average experimental variograms from the MST method, original Snesim algorithm, and TI. (**a**) Unconditional simulation in the x direction, (**b**) unconditional simulation in the y direction, (**c**) conditional simulation in the x direction, (**d**) conditional simulation in the y direction, (**e**) practical simulation in the x direction, (**f**) practical simulation in the y direction.

5. Conclusions

The idea of using MSTs to enhance the simulation continuity of the Snesim algorithm was based on a mathematical analysis of data events collected from a TI. It was demonstrated that the MST method could enhance the simulation continuity of the Snesim algorithm by reproducing the fine connectivity of channel facies in 2D fluvial reservoir modeling. To assess the feasibility of implementing a tree-based computing system, four search trees were created by sorting the search neighborhoods according to their proximity to the central node of the data template. The local CPDF of the simulating node was determined by taking the averages of the CPDFs derived from the MSTs. Both theoretical tests of unconditional and conditional simulations, as well as the practical simulation test of the $Ng3^3$ sublayer, confirmed the effectiveness of the MST method in enhancing simulation continuity in 2D fluvial reservoir modeling. Considering the precision of the local CPDF at each unsampled node, a compromise strategy called double retrieving was adopted. The number of search trees retrieved for the simulating nodes was reduced to some extent, further enhancing the accuracy of the CPDF. The CCL method was introduced to evaluate the simulation continuity of the theoretical and practical tests by the MST method and the original Snesim algorithm, demonstrating the robust capability of the MST method.

The utilization of MSTs was shown to enhance the simulation continuity of geobodies by restoring specific local regions of the sand channels compared to the original Snesim algorithm. However, it is essential to acknowledge that this method is susceptible to speed and memory constraints. Parallel computing was deployed for the construction of the search trees. However, due to the relatively small size of the tasks, alongside issues such as data duplication, thread overheads, and the necessity for synchronization mechanisms, this approach did not result in substantial performance enhancements in this particular context. In the future, the MST method will be utilized for performing both multiple-categories facies modeling and 3D facies modeling.

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

Funding: This research was funded by the China Scholarship Council, grant number 202308510303.

Data Availability Statement: The data presented in this study are openly available in GitHub at the link: https://github.com/ZhouChuanyou/MPS_Snesim, accessed on 17 February 2024.

Acknowledgments: The authors would like to express their gratitude to the editors and reviewers for their valuable comments and insights. The corresponding author extends great appreciation to the China Scholarship Council for supporting visiting study in the UK.

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

Appendix A

Here, i↑&j↑, i↑&j↓, i↓&j↑, and i↓&j↓ were used to sort the search neighborhood sequence according to the distance d in Equation (A1) in ascending and descending order using permutations and combinations. The letter d represents the Euler distance. The i and j indices are relative coordinates (Δx & Δy) of search neighborhoods relative to the central node of the data template. Listing A1 illustrates the implementation of this function.

$$\mathrm{d} = (\Delta \mathrm{x})^2 + (\Delta \mathrm{y})^2 \tag{A1}$$

This simplified approach can generate four equivalent data templates that correspond to four search trees. The local conditional probability distribution function (CPDF) of the simulating node is accurately derived from the data events of the four search trees.

The MST method has the advantage of increasing the precision of local CPDFs, which contributes to simulation continuity.

Listing A1: Search neighborhood sorting.

```
% d: the distance between search
% neighborhood and the central node u
d = sqrt(i.^2 + j.^2);
ad = {'ascend', 'descend'};
c = 1;
% sort the search neighborhood
for i = 1: size(ad,2)
     for j = 1: size(ad,2)
          sort_data{c} = sortrows([d(:), i(:), j(:)], [1,2,3], {'ascend', ad{i},ad{j}});
           c = c + 1;
      end
 end
 % 4 data templates/the base of 4 trees
 for i = 1: size(sort_data,2)
      % n_cd: the number of conditional data
      template{i} = [sort_data{i}(2:(n_cd+1), 2: end)];
 end
```

References

1. Wang, Z.; Chen, T.; Hu, X.; Wang, L.; Yin, Y. A Multi-Point Geostatistical Seismic Inversion Method Based on Local Probability Updating of Lithofacies. *Energies* **2022**, *15*, 299. [CrossRef]
2. Kang, Q.; Hou, J.; Liu, L.; Hou, M.; Liu, Y. Quantitative Prediction of Braided Sandbodies Based on Probability Fusion and Multi-Point Geostatistics. *Energies* **2023**, *16*, 2796. [CrossRef]
3. Wang, X.; Zhang, F.; Li, S.; Dou, L.; Liu, Y.; Ren, X.; Chen, D.; Zhao, W. The Architectural Surfaces Characteristics of Sandy Braided River Reservoirs, Case Study in Gudong Oil Field, China. *Geofluids* **2021**, *2021*, 8821711. [CrossRef]
4. Caers, J.; Zhang, T. Multiple-point Geostatistics: A Quantitative Vehicle for Integrating Geologic Analogs into Multiple Reservoir Models. In *Integration of Outcrop and Modern Analogs in Reservoir Modeling*; American Association of Petroleum Geologists: Tulsa, OK, USA, 2004. [CrossRef]
5. Caers, J.K.; Srinivasan, S.; Journel, A.G. Geostatistical Quantification of Geological Information for a Fluvial-Type North Sea Reservoir. *SPE Reserv. Eval. Eng.* **2000**, *3*, 457–467. [CrossRef]
6. Strebelle, S.B. Sequential Simulation for Modeling Geological Structures from Training Images. In *Stochastic Modeling and Geostatistics: Principles, Methods, and Case Studies, Volume II*; The American Association of Petroleum Geologists: Tulsa, Oklahoma, USA, 2006.
7. Guardiano, F.B.; Srivastava, R.M. Multivariate Geostatistics: Beyond Bivariate Moments. In *Geostatistics Tróia '92: Volume 1*; Soares, A., Ed.; Springer: Dordrecht, The Netherlands, 1993; pp. 133–144.
8. Strebelle, S.B.; Journel, A.G. Reservoir Modeling Using Multiple-Point Statistics. In Proceedings of the SPE Annual Technical Conference and Exhibition, New Orleans, LA, USA, 30 September–3 October 2001; p. SPE-71324-MS.
9. Jef Caers, S.S.; Payrazyan, K. Stochastic integration of seismic data and geologic scenarios: A West Africa submarine channel saga. *Lead. Edge* **2003**, *22*, 192–196. [CrossRef]
10. Burc Arpat, G. SIMPAT: Stochastic simulation with patterns. In *17 SCRF Meeting Stanford Center for Reservoir Forecasting*; Stanford University: Stanford, CA, USA, 2004.
11. Zhang, T.; Switzer, P.; Journel, A. Filter-Based Classification of Training Image Patterns for Spatial Simulation. *Math. Geol.* **2006**, *38*, 63–80. [CrossRef]
12. Honarkhah, M.; Caers, J. Stochastic Simulation of Patterns Using Distance-Based Pattern Modeling. *Math. Geosci.* **2010**, *42*, 487–517. [CrossRef]
13. Mahmud, K.; Mariethoz, G.; Caers, J.; Tahmasebi, P.; Baker, A. Simulation of earth textures by conditional image quilting. *Water Resour. Res.* **2014**, *50*, 20. [CrossRef]
14. Mariethoz, G.; Renard, P. Reconstruction of Incomplete Data Sets or Images Using Direct Sampling. *Math. Geosci.* **2010**, *42*, 245–268. [CrossRef]
15. Yang, L.; Hou, W.; Cui, C.; Cui, J. GOSIM: A multi-scale iterative multiple-point statistics algorithm with global optimization. *Comput. Geosci.* **2016**, *89*, 57–70. [CrossRef]
16. Pourfard, M.; Abdollahifard, M.J.; Faez, K.; Motamedi, S.A.; Hosseinian, T. PCTO-SIM: Multiple-point geostatistical modeling using parallel conditional texture optimization. *Comput. Geosci.* **2017**, *102*, 116–138. [CrossRef]

17. Wang, L.; Yin, Y.; Wang, H.; Zhang, C.; Feng, W.; Liu, Z.; Wang, P.; Cheng, L.; Liu, J. A method of reconstructing 3D model from 2D geological cross-section based on self-adaptive spatial sampling: A case study of Cretaceous McMurray reservoirs in a block of Canada. *Pet. Explor. Dev.* **2021**, *48*, 407–420. [CrossRef]
18. Liu, Y. Using the Snesim program for multiple-point statistical simulation. *Comput. Geosci.* **2006**, *32*, 1544–1563. [CrossRef]
19. Bastante, F.G.; Ordóñez, C.; Taboada, J.; Matías, J.M. Comparison of indicator kriging, conditional indicator simulation and multiple-point statistics used to model slate deposits. *Eng. Geol.* **2008**, *98*, 50–59. [CrossRef]
20. Boucher, A. Considering complex training images with search tree partitioning. *Comput. Geosci.* **2009**, *35*, 1151–1158. [CrossRef]
21. Hajizadeh, A.; Safekordi, A.; Farhadpour, F.A. A multiple-point statistics algorithm for 3D pore space reconstruction from 2D images. *Adv. Water Resour.* **2011**, *34*, 1256–1267. [CrossRef]
22. Wu, Y.; Lin, C.; Ren, L.; Yan, W.; An, S.; Chen, B.; Wang, Y.; Zhang, X.; You, C.; Zhang, Y. Reconstruction of 3D porous media using multiple-point statistics based on a 3D training image. *J. Nat. Gas Sci. Eng.* **2018**, *51*, 129–140. [CrossRef]
23. Huang, T.; Lu, D.-T.; Li, X.; Wang, L. GPU-based SNESIM implementation for multiple-point statistical simulation. *Comput. Geosci.* **2013**, *54*, 75–87. [CrossRef]
24. Cui, Z.; Chen, Q.; Liu, G.; Mariethoz, G.; Ma, X. Hybrid parallel framework for multiple-point geostatistics on Tianhe-2: A robust solution for large-scale simulation. *Comput. Geosci.* **2021**, *157*, 104923. [CrossRef]
25. Strebelle, S.; Cavelius, C. Solving Speed and Memory Issues in Multiple-Point Statistics Simulation Program SNESIM. *Math. Geosci.* **2014**, *46*, 171–186. [CrossRef]
26. Nagendra Babu, M.; Ambati, V.; Nair, R.R. An integrated approach to lithofacies characterization of a sandstone reservoir using the Single Normal Simulation equation: A Case study. *J. Pet. Sci. Eng.* **2022**, *208*, 109626. [CrossRef]
27. Zhou, F.; Shields, D.; Tyson, S.; Esterle, J. Comparison of sequential indicator simulation, object modelling and multiple-point statistics in reproducing channel geometries and continuity in 2D with two different spaced conditional datasets. *J. Pet. Sci. Eng.* **2018**, *166*, 718–730. [CrossRef]
28. Naderi, H.; Fathianpour, N.; Tabaei, M. MORPHSIM: A new multiple-point pattern-based unconditional simulation algorithm using morphological image processing tools. *J. Pet. Sci. Eng.* **2019**, *173*, 1417–1437. [CrossRef]
29. Walsh, D.A.; Manzocchi, T. A method for generating geomodels conditioned to well data with high net:gross ratios but low connectivity. *Mar. Pet. Geol.* **2021**, *129*, 105104. [CrossRef]
30. Renard, P.; Allard, D. Connectivity metrics for subsurface flow and transport. *Adv. Water Resour.* **2013**, *51*, 168–196. [CrossRef]
31. Shahraeeni, M. Enhanced Multiple-Point Statistical Simulation with Backtracking, Forward Checking and Conflict-Directed Backjumping. *Math. Geosci.* **2019**, *51*, 155–186. [CrossRef]
32. Wang, X.; Yu, S.; Li, S.; Zhang, N. Two parameter optimization methods of multi-point geostatistics. *J. Pet. Sci. Eng.* **2022**, *208*, 109724. [CrossRef]
33. Haralick, R.M.; Shanmugam, K.; Dinstein, I. Textural Features for Image Classification. *IEEE Trans. Syst. Man Cybern.* **1973**, *SMC-3*, 610–621. [CrossRef]
34. Zhang, P.; Qian, X.; Guo, X.; Yang, X.; Li, G. Automated demarcation of the homogeneous domains of trace distribution within a rock mass based on GLCM and ISODATA. *Int. J. Rock Mech. Min. Sci.* **2020**, *128*, 104249. [CrossRef]
35. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity Mappings in Deep Residual Networks. In Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016; pp. 630–645.
36. Kumar, D.; Srinivasan, S. Indicator-based data assimilation with multiple-point statistics for updating an ensemble of models with non-Gaussian parameter distributions. *Adv. Water Resour.* **2020**, *141*, 103611. [CrossRef]
37. Zhang, T.; Shen, T.; Dong, Y.; Du, Y. 3D-FGAN: A 3D stochastic reconstruction method of digital cores. *Geoenergy Sci. Eng.* **2024**, *233*, 212590. [CrossRef]
38. Sun, C.; Demyanov, V.; Arnold, D. Geological realism in Fluvial facies modelling with GAN under variable depositional conditions. *Comput. Geosci.* **2023**, *27*, 203–221. [CrossRef]
39. Zheng, N.; Li, Z.; Xia, X.; Gu, S.; Li, X.; Jiang, S. Estimating line contaminant sources in non-Gaussian groundwater conductivity fields using deep learning-based framework. *J. Hydrol.* **2024**, *630*, 130727. [CrossRef]
40. Strebelle, S. Conditional Simulation of Complex Geological Structures Using Multiple-Point Statistics. *Math. Geol.* **2002**, *34*, 1–21. [CrossRef]
41. Kesheng, W.; Ekow, O.; Arie, S. Optimizing connected component labeling algorithms. In Proceedings of the SPIE—The International Society for Optical Engineering, San Diego, CA, USA, 29 April 2005; pp. 1965–1976.
42. Rakhmadi, A.; Othman, N.Z.S.; Bade, A.; Rahim, M.S.M.; Amin, I.M. Connected Component Labeling Using Components Neighbors-Scan Labeling Approach. *J. Comput. Sci.* **2010**, *6*, 3088–3107. [CrossRef]
43. Zhang, D.; Ma, H.; Pan, L. A gamma-signal-regulated connected components labeling algorithm. *Pattern Recognit.* **2019**, *91*, 281–290. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.