Shahrood University of Technology
Iranian Society of Mining Engineering (IRSME)
Journal of Mining and Environment (JME)
journal homepage: www.jme.shahroodut.ac.ir
THOMSON REUTERS
Vol. 10, No. 1, 2019, 139-149.
DOI: 10.22044/jme.2018.7335.1584

A comprehensive benchmark between two filter-based multiple-point simulation algorithms

M. Bavand Savadkoohi[1*], B. Tokhmechi[1], E. Gloaguen[2] and A.R. Arab-Amiri[1]

1. Faculty of Mining, Petroleum & Geophysics Engineering, Shahrood University of Technology, Shahrood, Iran
2. Centre Eau Terre Environnement, Institut National de la Recherche Scientifique, Québec, Canada

Received 7 August 2018; received in revised form 28 August 2018; accepted 29 August 2018

Keywords

Multiple-Point Simulation
Filter-Based Algorithms
Geostatistical Simulation
FILTERSIM Algorithm
CCWSIM Algorithm

Abstract

Computer graphics offer various gadgets to enhance the reconstruction of high-order statistics that are not correctly addressed by the two-point statistics approaches. Almost all the newly developed multiple-point geostatistics (MPS) algorithms, to some extent, adapt these techniques to increase the simulation accuracy and efficiency. In this work, a scrutiny comparison between our recently developed MPS algorithm, the cross-correlation-wavelet simulation (CCWSIM), and a well-known MPS algorithm, FILTERSIM, is performed. The main motivation to benchmark these two algorithms is that both exploit some digital image processing filters for feature extraction. Indeed, both algorithms compute the similarity (or dissimilarity) between data events in simulation grid and training image in the feature space. In order to compare the accuracy of the algorithms, some statistics such as facies proportion, variogram, and connectivity function are computed. The results obtained reveal an excellent agreement of the CCWSIM realizations with the training image rather than FILTERSIM. Furthermore, on average, the required simulation runtime for CCWSIM is at least 10 times less than that for FILTERSIM.

1. Introduction

Most geoscientific applications require the numerical modeling of geology. For decades, this modeling has been done using two-point statistics algorithms, which are not appropriate to model the complex geological features. The classical geostatistical techniques just consider the linear correlation of a pair of points in space through variogram-based methods. [1-3]. However, they have not been designed to reconstruct high-order statistics relating to complex spatial structures [4]. Hence, high-order geostatistics has been suggested to go beyond the bivariate moments and reconstruct such complex geological formations [5, 6].

As the data itself does not permit to extract its high-order statistics due to scarcity, multiple-point geostatistics (MPS) borrows the higher-order statistics from a conceptual *training image*. This image contains prior numerical information (architecture, dispersion, proportion, etc.) that is collected from previous explorations or related cases [7]. The first MPS algorithm utilizes the training image to infer the conditional distribution function (CDF) of a random variable [5]. However, the computational cost of the first algorithm is extremely high and makes it non-functional. The second MPS algorithm, SNESIM, relies on a tree structure to eliminate the first algorithm problems and improve the computational efficiency [8]. In fact, it uses a tree structure to store the data events before starting the simulation process. Later, in order to decrease the computational load and improve the accuracy of SNESIM, several modified versions were proposed [9, 10]. Another effective pixel-based MPS algorithm, direct sampling (DS), inspired its original idea from Shannon [11], who introduced a method to re-build the text by conditioning to

Corresponding author: mojtababavand@yahoo.com (M. Bavand Savadkoohi).

previous stitched letters. This algorithm samples its favorite data event directly from the training image pixel-by-pixel. DS is the first MPS algorithm without the pattern database [12]. Although DS decreases RAM consumption, due to its pixel-based background, it is not a very fast algorithm [13]. A creative remedy has been proposed by Rezaee *et al*. to enhance the efficiency of DS. In their algorithm, one or more layers of pixels around the desired pixel in the training image are patched into the simulation grid [14].

With the growth of computer hardware, the pixel-based algorithms [8, 12] were substituted with pattern-based ones [14-19]. Pattern-based algorithms, similar to a jigsaw puzzle, reconstruct stochastic realization through stitch patterns into the simulation grid regarding the desired data events [20]. Although primary pattern-based algorithms produce more visually-appealing realizations, the lack of an effective plan for retrieval patterns leads to increase in the computational cost, i.e. search the entire patterns to find the eligible pattern [21]. In the FILTERSIM algorithm, the first pattern-based algorithm, the computational load is considerably decreased through some digital image-processing filters [16]. In this algorithm, several filters are utilized to extract, cluster, and reduce the dimension of the most important features in patterns. In a similar way, wavelet decomposition was proposed as a pre-processor to reduce the dimension of the patterns before the clustering step in some pattern-based algorithms [22, 23]. However, these approaches are RAM-demanding and do not have a significant enhancement in the performance of pattern-based algorithms.

Although pattern-based algorithms reduce the CPU cost, the data conditioning problem has not been addressed properly in these algorithms. In the meantime, some algorithms give rise to hard and soft data conditioning through image processing techniques and optimization-based methods [24-27].

A brief review of the MPS algorithms evolution shows their close relation with computer graphics algorithms in texture synthesis [20, 26, 28, 29]. Mariethoz and Lefebvre have conducted a scrutiny study around the historical development and comparable concepts of both disciplines [30]. A typical application of texture synthesis methods named cross-correlation function (CCF) and image quilting (IQ) [31] has been successfully tested in the Tahmasebi *et al.*'s work (CCSIM) [19]. However, CCSIM suffers from the recursive manner for data conditioning. For this goal, Tahmasebi et al. have developed a multi-scale algorithm to improve the computational efficiency as well as data conditioning [24].

The accuracy and computational efficiency of the MPS algorithms depend directly upon how the similarity distance (SD) is computed [17, 19]. Actually, the scheme of evaluation and search dimension affect the simulation accuracy and yield cost, respectively. Tremendous efforts have been made to address this critical issue [32-35]. In this work, we compared the quality and efficiency of our recently developed algorithm, CCWSIM, with a fast version of the original FILTERSIM algorithm [16, 20]. CCWSIM exploits two effective digital image-processing techniques to improve the simulation results in terms of accuracy and efficiency. In CCWSIM, an effective combination of CCF and discrete wavelet transform (DWT) leads to a significant enhancement in the simulation results [36]. However, since each one of these tools has been separately used in different MPS methods [19, 23, 25], we developed a novelty hybrid of them.

In this paper, first, the mathematical aspect of DWT is described. Next, the DWT role in dimension reduction of the CCF is highlighted. Then the CCWSIM algorithm is explained, and a modified equation is presented to compute CC through wavelet coefficients. Next, to validate the simulation results, the reproduction of essential statistics is investigated. Subsequently, the capability of CCWSIM in honoring hard data is compared with FILTERSIM. In the last section, the performance of the two filter-based algorithms is evaluated in the case of an exhaustive collection of TIs.

2. Background of CCWSIM

As mentioned earlier, CCWSIM utilizes two effective digital image-processing techniques to improve the simulation results in terms of accuracy and efficiency. Let us have a brief review of the CCWSIM principle and explain the important duty of DWT in the improvement of MPS results.

2.1. Cross-correlation-wavelet-simulation (CCWSIM)

Cross-correlation function (CCF) is widely applied as a criterion to measure the degree of similarity between patterns. [19, 37]. Due to the inherent properties of CCF to find the best-adapted pattern, it can be quite expensive for large images. Hence, different non-exhaustive families

of CCF have been developed to improve the computational efficiency [38]. In our algorithm, a simple but effective DWT-based remedy is proposed to reduce the computational load of CCF. We use the DWT property that allows for quick compresses of a regionalized image by skipping some non-significant coefficients.

2.2. DWT principle

Wavelet transform decomposes any signal into a set of orthogonal basis functions. A 2D signal can be regarded as a square integrable function in the Hilbert space. Thanks to the multi-resolution characteristic of the wavelet decomposition, it presents a set of orthonormal function through translating and dilating of primary basis function, recognized as the mother wavelet Ψ^B (where $B=\{H,V,D\}$).

Considering *pat(x, y)* as a 2D dataset of size $N \times N$, DWT of the *pat* at level j has been formulated by Fan and Xia [39]:

$$pat(x,y) = \sum_{k,i=0}^{N_j-1} a^k_{(J,k,i)} \phi^{LL}_{(J,k,i)}(x,y) + \sum_{B=\{H,V,D\}} \sum_{j=1}^{J} \sum_{k,i=0}^{N_j-1} Z^B_{(j,k,i)} \Psi^B_{(j,k,i)}(x,y) \quad (1)$$

Where Ψ^B and ϕ^{LL} are the mother wavelet and scaling function, respectively, $N_j = N/2^j$, $a_{J,k,i}$ are the approximate coefficients (a) and $Z^B_{j,k,i}$ are the detail coefficients at level j in horizontal, vertical and diagonal directions, respectively *(H,V,D)*.

Practically, the 2D DWT on any discrete signal proceeds in two major steps. In the first step, the signal is decomposed by some low- and high-pass filters. Next, the down-sampling procedure is applied to the resulting frequency sub-band rows and columns. Indeed, after each decomposition level j, one omnidirectional low frequency and three directional high frequency sub-bands that are called the approximate and detail coefficients, respectively, are obtained [40]. Then according to the decomposition level, the process is continued using the resulting approximate coefficients to provide the approximate and detail coefficients for the next decomposition level. It is worth mentioning that after every filtering step, 75% of the detail coefficients are lost but due to the intrinsic property of wavelet decomposition, the vital potential patterns are carefully preserved [41]. The DWT approximate coefficients try to preserve the most important local and global variability in a given pattern. The two levels of DWT of an image and its corresponding sub-bands are shown in Figure 1.

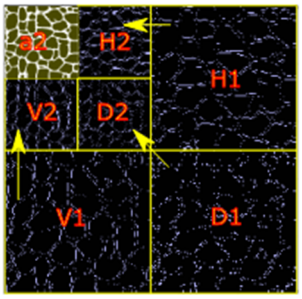


Figure 1. Two levels (*j*) of DWT: approximate coefficients (a2) and directional (H, V, D) detail coefficients.

DWT has some interesting features: (1) In order to reach an acceptable level of decomposition, the number of levels can be increased according to the size of the image and the scale of the features. (2) Due to the different attributes and structures in an image, there are significant numbers of mother wavelet function to maximize feature extraction. (3) At each decomposition level, the decomposed image can be perfectly restored by *Inverse*-DWT (IDWT).

In this work, the *Haar* wavelet was selected for wavelet decomposition. The main reason is that the *Haar* wavelet has the simplest and the most appropriate wavelet function to analyze signals with unforeseen changes. The *Haar* wavelet Ψ^B and scaling function ϕ^{LL} are presented as [42]:

$$\Psi^s(t) = \begin{cases} 1 & 0 \le t \le 1/2 \\ -1 & 1/2 \le t \le 1 \\ 0 & otherwise \end{cases}, \quad \phi^{LL}(t) = \begin{cases} 1 & 0 \le t \le 1 \\ 0 & otherwise \end{cases} \quad (2)$$

2.3. CCWSIM algorithm

CCWSIM can handle both unconditional and conditional simulations. Similar to some pattern-based algorithms, our algorithm renovates the patterns in a one-sided path referred to as the "unilateral" or "raster" path [26]. The simulation procedure initializes from one corner in SG and continues in a straightforward order. After reaching the end of the path, the simulation restarts at the beginning of the next track. The one-sided path in a lattice grid is shown in Figure 2.

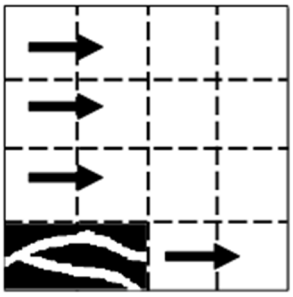


Figure 2. Lattice SG with pre-defined one-sided path.

Since CCWSIM is a pattern-based algorithm, at each iteration, a piece of pixel was patched to SG. Similar to different IQ approaches, we also applied an overlap region (OR) instead of the whole data event to compute SD. However, the main difference and what makes the CCWSIM more efficient is the use of the cross-correlation-wavelet (CCW) coefficients to compute SD. Actually, CCWSIM utilizes filters to capture the most important features in a given pattern and maps them in a reduced space. Indeed, computation of SD between patterns is done in the wavelet feature space. Given a TI and OR of size $M \times N$, our modified version of CC, referred to as CCW, at pixel position *(s,t)* can be defined as:

$$CCW(x,y) = \sum_{t=0}^{N_j-1} \sum_{s=0}^{M_j-1} cA_j^{TI}(x+s, y+t)\, cA_j^{OR}(s,t) \qquad (3)$$

where $N_j = N/2^j$, $M_j = M/2^j$, cA_j^{TI} is the wavelet approximate coefficients of TI in level *j*, and cA_j^{OR} is the wavelet approximate coefficient of OR at level *j*.

It is to be noted that the final vector length of OR after *j* levels of decomposition will be $M_j \times N_j$. Hence, depending on the level of the wavelet decomposition *j*, the dimension of the primary pattern can be reduced significantly. For instance, a 2D TI of size 1000 × 1000 has a vector length of 1,000,000. If 3 levels of wavelet decomposition are done, the length of the final vector will be 15,625, which is dramatically less than the original length. In addition, there are the same benefits for OR. A Scheme of the CCW coefficients computation is depicted in Figure 3.

Henceforth, there are two different scenarios depending on the type of simulation (unconditional or conditional). For the unconditional simulation, some of the high-ranked SDs are selected, and the final pattern is randomly drawn from the pool of candidates.

In this work, the hard data (HD) conditioning was performed in a similar fashion as introduced by Tahmasebi *et al.* [24]. As mentioned earlier, computation of SD between patterns is done through the CCW coefficients. Indeed, the preserved patterns must see the following essential conditions: (1) SD between patterns and OR should be minimized; and (2) The patterns must respect HD not only inside the search template but also within a certain pioneer area. For more details about the conditioning method, the readers are invited to see [24]. Referring to our previous remark, DWT has an excellent ability to back-transform the decomposed patterns into the original space. More precisely, DWT is a bijective decomposition that permits to switch to the original space in each decomposition level. According to this feature, shifting the desired pattern to the original space is done by IDWT in the final stages of simulation. The switching procedure of a qualified pattern from the wavelet space to the original space is shown in Figure 4. To summarize, a pseudo-code for CCWSIM is given in Table 1.

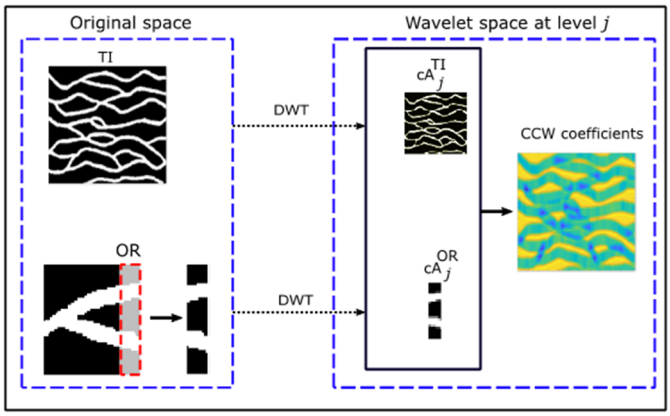


Figure 3. Computation of CCW coefficients using decomposed TI and OR (red-dashed line) in the wavelet feature space.

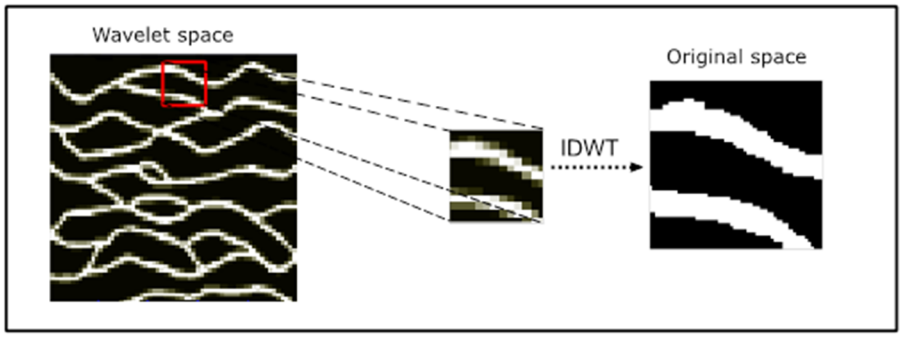


Figure 4. Desired pattern in wavelet space (red line) and reconstruction procedure to original space.

Table 1. A pseudo-code for CCWSIM algorithm.

Require: (*TI*, size$_{TI}$, size$_{OL}$, realization_number, *SG*, size$_{SG}$, wavelet_level)
for *i* = 1 to last uninformed block in *SG* ***do***
define a unilateral path in *SG*
if *i* = 1
drop a random patch to *SG*
else
extract *OR*
compute DWT (*TI)* and DWT (*OL)*
compute *CCW (TI, OL)*
sort (SD) and find best-matched pattern
calculate IDWT (desired pattern)
past pattern into *SG*
…… **end if**
end for

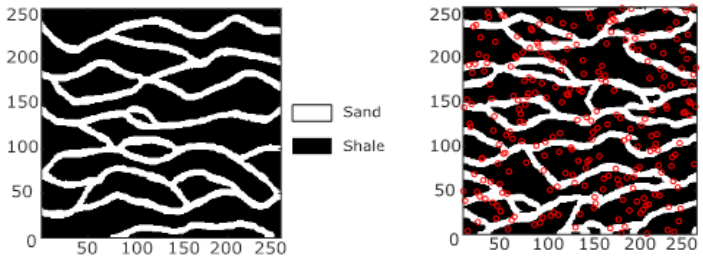
3. Numerical results and discussion

In this work, all the training images were obtained from an open-source training image library (http://www.trainingimages.org/training-images-library.html). The first exercise was conducted with a binary training image (TI). In this TI, the sinuous sandy channels are surrounded by a shaly background, which represents a real oil reservoir. We used the same procedure as Mariethoz *et al.* to generate TI [12]. We applied the Stanford geostatistical modeling software (SGeMS-v.2.1) to provide the FILTERSIM realizations, firstly developed by Remy *et al.* [43]. The reference image, TI, and 250 HD are demonstrated in Figure 5. We developed a new MATLAB code to generate the CCWSIM realizations.

First of all, the input parameters should be tuned to run the algorithms. The simulation grid (SG) dimension was 256x256 for both algorithms. In the CCWSIM algorithm, the sizes of the template and OR area were selected to be 20x20 and 8, respectively. Since the size of TI was relatively small, one level of DWT was computed. The template and inner template sizes in the FILTERSIM algorithm must be odd numbers, thus their sizes were tuned to 21x21 and 7x7, respectively. The number of pattern clusters was considered to be 100. The SGeMS input parameters for FILTERSIM are reported in Table 2.

50 realizations for each algorithm were generated. The mean of the realizations (an E-type map) was also calculated. Two random realizations and the E-type map are depicted in Figure 6. This figure shows that CCWSIM reproduces convoluted structures much more accurate rather than FILTERSIM. It is obvious that CCWSIM preserves better the connectivity of the sandy channels. We could choose a bigger template to improve the reproduction of the connectivity in FILTERSIM but it will dramatically increase the simulation time.

As shown in the CCWSIM *E*-type map, it reproduces the curved and complex architecture of the TI well. The runtimes for the CCWSIM and FILTERSIM algorithms for each realization were about 5.8 and 61 seconds, respectively.



(a) **(b)**

Figure 5. TI with two facies: sandy channel in a shaly background (a) and the reference image and 250 HD (red cycles) randomly selected from that (b).

Table 2. SGeMS input parameters for FILTERSIM algorithm.

Algorithm name	filtersim_cate
GridSelector_Sim value	sim_grid
Property_Name_Sim value	sim_data
Nb_Realizations value	50
Seed value	211175
PropertySelector_Training grid	ti
Property	v
Scan_Template value	[21 21 1]
Patch_Template_ADVANCED value	[7 7 1]
Nb_Facies value	2
Treat_Cate_As_Cont value	0
Trans_Result value	1
Hard_Data grid	hd
property	z
Use_SoftField value	0
SoftData_properties count	0
TauModelObject value	[1 1]
Region_Indicator_Prop value	simdata__real0
Use_Region value	0
Nb_Multigrids_ADVANCED value	3
Debug_Level value	0
Cmin_Replicates value	[10 10 10]
Data_Weights value	[0.5 0.3 0.2]
CrossPartition value	0
KMeanPartition value	1
Nb_Clusters_ADVANCED value	100
Nb_Clusters_ADVANCED2 value	2
Use_Normal_Dist value	0
Use_Score_Dist value	1
Filter_Default value	1
Filter_User_Define value	0

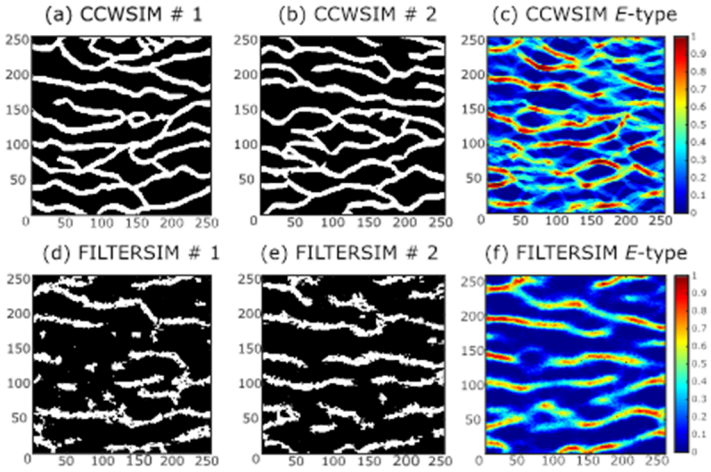


Figure 6. Two realizations and ensemble average of 50 realizations obtained from CCWSIM (a, b, and c) and the FILTERSIM algorithm (d, e, and f).

3.1. Validation tests

In the next investigation, in order to validate and also compare the simulated realization results, three statistics were checked. These criteria include: (1) facies proportion, (2) variogram, and (3) connectivity function. Although reproduction of these single- and two-point statistics cannot guarantee the reproduction of the high-order ones, they are still informative on the overall reproduction of spatial pattern reproduction.

For the first criterion, the proportion of the sandy channels was computed in 50 realizations and compared with the RI proportion, as shown in Figure 7 (a): (1) Both algorithms have a stable behavior in reproducing sandy channel proportion in realizations. (2) Although the proportion of the channels in RI is about 0.27, FILTERSIM systematically tends to reproduce less sandy channel facies in realizations (0.18). (3) The intervals between the channel proportion in the CCWSIM realizations and in TI are really close. However, CCWSIM reproduces a little more the channel proportion.

Another paramount criterion for measuring two-point statistics is the variogram. We selected 10 realizations randomly from each algorithm and compared their variograms with the one computed with RI. The variograms were computed in the North-South and East-West directions, as presented in Figure 7 (b, c). One can see that CCWSIM better reproduces the variogram of RI compared with FILTERSIM. Indeed, the slope and sill of the CCWSIM variograms are in more conformity with the RI one.

For last investigating around the traditional statistics, the connectivity function of the realizations was computed. The connectivity function is defined as the conjunction probability of a pixel located at a certain lag and direction of other pixels [44]. Since the long-range connectivity of channels was in the East–West direction, the connectivity function was computed along this direction, as shown in Figure 7 (d). A visual inspection shows that the connectivity of the RI and CCWSIM realizations are in good agreement. It seems that FILTERSIM fails to reproduce the connectivity of channels in realizations. One reason can be the unrealistic reproduced proportion of facies in the FILTERSIM realizations that can impact on the shape and decay of the connectivity function [45].

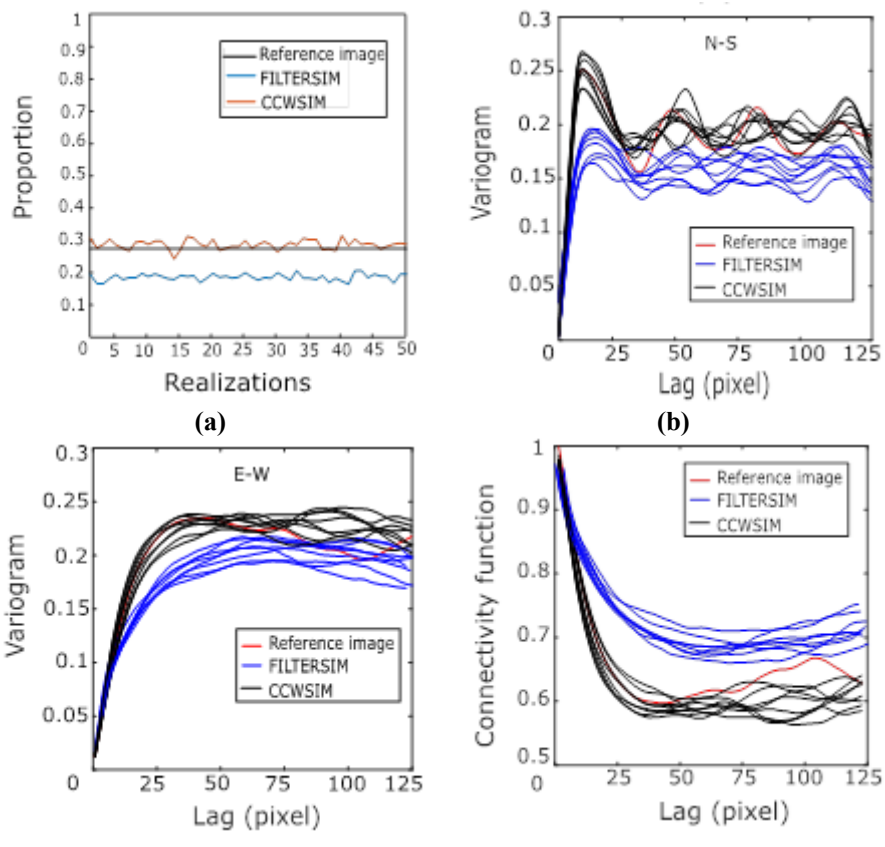


Figure 7. Proportion of channels (a), variograms of realization in NS (b), EW (c) and probability of connection in reference image, FILTERSIM, and CCWSIM realizations (d).

3.2. Data conditioning

In this section, the performance of the CCWSIM and FILTERSIM algorithms was compared in terms of data conditioning. We considered 4 sparse hard data (HD) in channel facies as the conditional points. Then 50 conditional realizations were generated, and so their E-type map. The E-type map and two randomly selected realizations are shown in Figures 8(a-f). Both algorithms highlight the presence of channels in the four pre-defined locations. More precisely, CCWSIM shows the explicit structures in the neighborhood of HD but FILTERSIM simply superimposes HD in the simulating process without considering the spatial continuity of the channels.

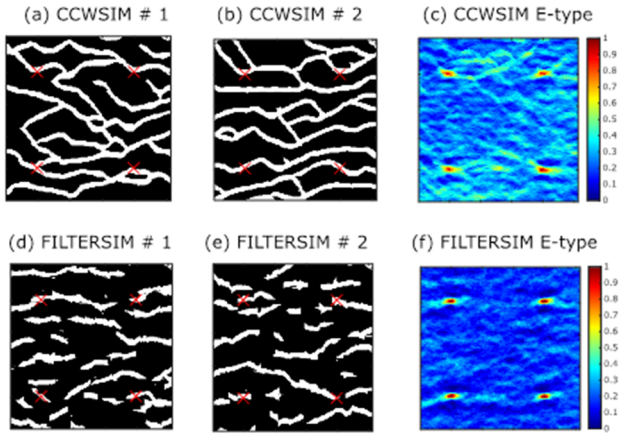
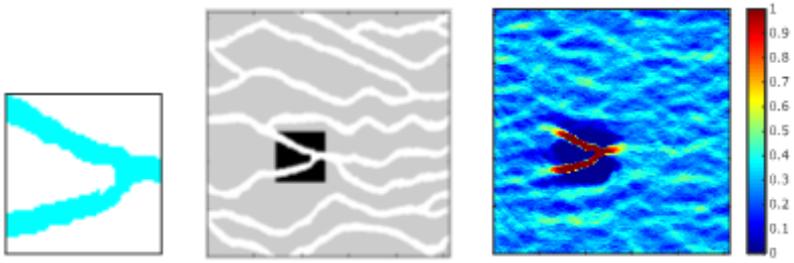


Figure 8. Conditional realizations forced by 4 HD (red crosses) for CCWSIM (a and b) and FILTERSIM (d and e) and their *E*-type maps obtained from 50 realizations (c and f), respectively.

3.3. Conditioning to geobody

Geobody extraction is an effective remedy for enhancing reservoir characterization, detecting anomalies, and determining facies. An independent geobody (object piece) is dropped in SG as a conditional patch to examine the behavior of our algorithm in terms of reproduction of a given geobody. Theoretically, it is expected that a relatively reliable behavior appears near the geobody. Using an object piece, 50 realizations were generated, and then the corresponding E-type map was computed. Thanks to the FILTERSIM algorithm freezing all the conditional data in SG, we did not do the geobody test for FILTERSIM. The geobody, a single realization, and an E-type map are presented in Figure 9 (a-c). Obviously, CCWSIM emphasizes the given geobody in the realizations perfectly.



(a) (b) (c)

Figure 9. Utilized geobody for conditioning (a), one single realization conditioned to the geobody (b), and its corresponding *E*-type map (c).

4. Other TIs

In this section, two algorithms using other TIs that are widely used in MPS exercises are compared. These TIs have two different structures: (1) Small-scale structures like pores and grains (a, c, and d). (2) Large-scale fabrications like fractures or channels (b, e, and f). For each algorithm, one single randomly conditional realization is shown. These realizations can better reveal the pros and cons of our algorithm and FILTERSIM.

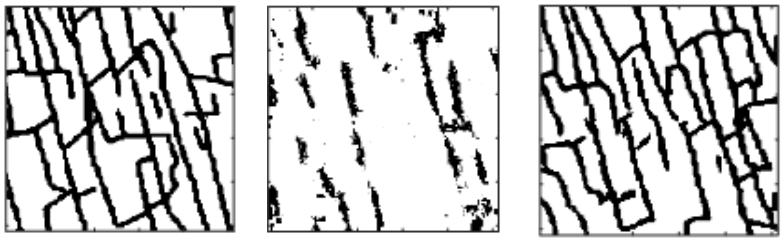
Fractures TI contains some fractures along a regular trend. All the input parameters for both algorithms were considered equal. TI, one realization for CCWSIM, and FILTERSIM are shown in Figure 10. CCWSIM reproduces well the fractured architecture compared to FILTERSIM.

Circles A random distribution of circles is selected, which mimics a porous media. The TI and realizations provided by FILTERSIM and CCWSIM are shown in Figure 11. The CCWSIM realization is in good agreement with TI compared with FILTERSIM. Furthermore, the simulation time is 10 times faster for CCWSIM.

Four categorized facies In the next test, in order to evaluate the performance of two comparative algorithms in the case of multi-facies categorical setting, a TI consisting of four facies is chosen. The TI and simulation results using FILTERSIM and CCWSIM are shown in Figure12. Similar to the previous exercises, CCWSIM reproduces the TI structures more accurately than FILTERSIM.

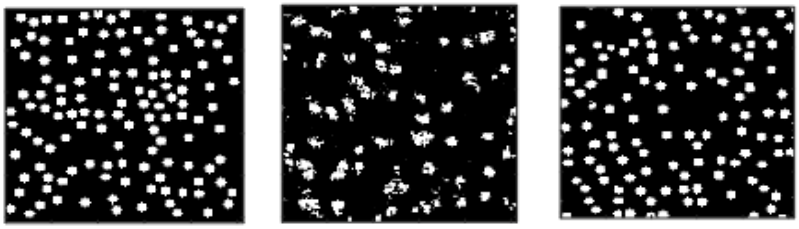
Quasi-stationary meandering channels In the last test, a more complicated TI with meandering channels is selected. These meandering channels are shaped by obstacles along the river path, and impose a quasi-stationary behavior on the channel structure. Due to the complexity of the structures, one promotion was considered to improve the results of FILTERSIM. We increased the number of clusters 10 times. The meandering channels and a single realization using both algorithms are shown in Figure 13. Despite increasing the clusters, FILTERSIM failed to reproduce the TI patterns.

The results obtained reveal a more visually-appealing outputs of our algorithm compared to FILTERSIM. Additionally, reconstruction of the facies sidelong connectivity is reproduced well by CCWSIM. From the computational achievement, we remark that simulation of the patterns through CCWSIM is at least 10 times less than FILTERSIM.



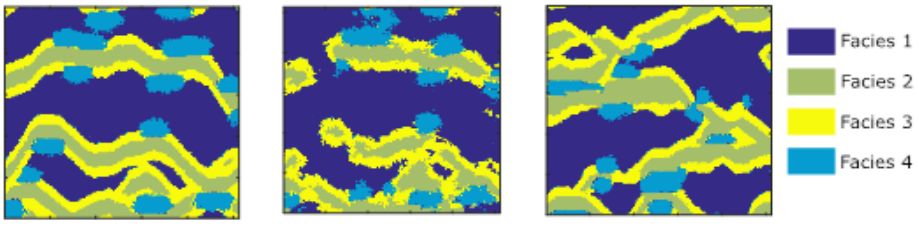
(a) Training image (b) FILTERSIM (c) CCWSIM

Figure 10. TI contains fractures (a), one single FILTERSIM and CCWSIM realization (b, c).



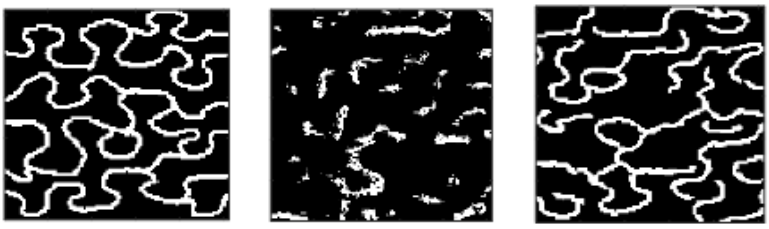
(a) Training image (b) FILTERSIM (c) CCWSIM

Figure 11. TI with randomly distributed circles (a), one single FILTERSIM and CCWSIM realization (b, c).



(a) Training image (b) FILTERSIM (c) CCWSIM

Figure 12. Four categorized facies TI (a), one single FILTERSIM, and CCWSIM realization (b, c).



(a) Training image (b) FILTERSIM (c) CCWSIM

Figure 13. Meandering channels TI (a), one single FILTERSIM and CCWSIM realization (b, c).

5. Conclusions

In this work, an exhaustive benchmarking between our MPS algorithm, CCWSIM, and a well-known filter-based algorithm, FILTERSIM, was conducted. Our algorithm exploits two effective image-processing techniques, DWT and CCF. A novel hybrid of late tools led to a significant improvement in the accuracy and efficiency of MPS simulation. Both algorithms are MPS pattern-based algorithms that utilize image-processing filters to feature extraction and dimension reduction in pattern. Two main characteristics of our algorithm are as what follow. (1) The similarity distance between the patterns is computed in wavelet space. (2) CCWSIM directly samples pattern from TI and does not need to build the pattern catalog. Therefore, in contrast to FILTERSIM, our algorithm is much less memory-demanding. In order to reach generalizable results, several conditional realizations are done using a variety range of TIs with completely different structures. Comparative results confirm an excellent reconstruction of the TI essential patterns with CCWSIM rather than FILTERSIM. In addition, CCWSIM is much more efficient than FILTERSIM, so that, on average, the average runtimes for each realization of CCWSIM are at least 10 times faster than FILTERSIM.

References

[1]. Goovaerts, P. (1998). Geostatistical tools for characterizing the spatial variability of microbiological and physico-chemical soil properties. Biology and Fertility of soils. 27 (4): 315-334.

[2]. Isaaks, E.H. and Srivastava, R.M. (1989). An introduction to applied geostatistics. Oxford university press. 471 P.

[3]. Mariethoz, G. (2018). When should we use multiple-point geostatistics, In: Sagar, B. D., Cheng, Q., & Agterberg, F. (Eds.), Handbook of Mathematical Geosciences: Fifty Years of IAMG., 645 P.

[4]. Journel, A.G. (1993). Geostatistics: roadblocks and challenges, In: Soares, A. O. (Ed.). Geostatistics Troia'92., Springer. 213 P.

[5]. Guardiano, F.B. and Srivastava, R.M. (1993). Multivariate geostatistics: beyond bivariate moments, In: Soares, A. O. (Ed.). Geostatistics Troia'92., Springer. 133 P.

[6]. Krishnan, S. and Journel, A. (2003). Spatial connectivity: from variograms to multiple-point measures. Mathematical Geology. 35 (8): 915-925.

[7]. Mariethoz, G. and Caers, J. (2014). Multiple-point geostatistics: stochastic modeling with training images. John Wiley & Sons. 27 P.

[8]. Strebelle, S. (2002). Conditional Simulation of Complex Geological Structures Using Multiple-Point Statistics. Mathematical Geology. 34 (1): 1-21.

[9]. Straubhaar, J., Renard, P., Mariethoz, G., Froidevaux, R. and Besson, O. (2011). An improved parallel multiple-point algorithm using a list approach, Mathematical Geosciences. 43 (3): 305-328.

[10]. Cordua, K.S., Hansen, T.M. and Mosegaard, K. (2015). Improving the pattern reproducibility of multiple-point-based prior models using frequency matching. Mathematical Geosciences. 47 (3): 317-343.

[11]. Shannon, C.E. (1948). A mathematical theory of communication, Bell Syst, Tech.J. 27: 379-423.

[12]. Mariethoz, G., Renard, P. and Straubhaar, J. (2010). The direct sampling method to perform multiple-point geostatistical simulations. Water Resources Research. 46 (11).

[13]. Meerschman, E., Pirot, G., Mariethoz, G., Straubhaar, J., Van Meirvenne, M. and Renard, P. (2013). A practical guide to performing multiple-point statistical simulations with the Direct Sampling algorithm. Computers & Geosciences. 52: 307-324.

[14]. Arpat, G.B. and Caers, J. (2005). A multiple-scale, pattern-based approach to sequential simulation, In: Leuangthong, O., & Deutsch, C. V. (Eds.), Geostatistics Banff 2004, Springer. pp. 255-264.

[15]. Rezaee, H., Mariethoz, G., Koneshloo, M. and Asghari, O. (2013). Multiple-point geostatistical simulation using the bunch-pasting direct sampling method. Computers & Geosciences. 54: 293-308.

[16]. Zhang, T., Switzer, P. and Journel, A. (2006). Filter-based classification of training image patterns for spatial simulation. Mathematical Geology. 38 (1): 63-80.

[17]. Honarkhah, M. and Caers, J. (2010). Stochastic simulation of patterns using distance-based pattern modeling. Mathematical Geosciences. 42 (5): 487-517.

[18]. Mahmud, K., Mariethoz, G., Caers, J., Tahmasebi, P. and Baker, A. (2014). Simulation of Earth textures by conditional image quilting. Water Resources Research. 50 (4): 3088-3107.

[19]. Tahmasebi, P., Hezarkhani, A. and Sahimi, M. (2012). Multiple-point geostatistical modeling based on the cross-correlation functions. Computational Geosciences. 16 (3): 779-797.

[20]. Wu, J., Zhang, T. and Journel, A. (2008). Fast FILTERSIM simulation with score-based distance. Mathematical Geosciences. 40 (7): 773-788.

[21]. Arpat, G.B. and Caers, J. (2007). Conditional simulation with patterns. Mathematical Geology. 39 (2): 177-203.

[22]. Chatterjee, S., Dimitrakopoulos, R. and Mustapha, H. (2012). Dimensional reduction of pattern-based simulation using wavelet analysis. Mathematical Geosciences. 44 (3): 343-374.

[23]. Mustapha, H., Chatterjee, S., Dimitrakopoulos, R. and Graf, T. (2013). Geologic heterogeneity recognition using discrete wavelet transformation for subsurface flow solute transport simulations. Advances in water resources. 54: 22-37.

[24]. Tahmasebi, P., Sahimi, M. and Caers, J. (2014). MS-CCSIM: Accelerating pattern-based geostatistical simulation of categorical variables using a multi-scale search in Fourier space. Computers & Geosciences. 67: 75-88.

[25]. Gloaguen, E. and Dimitrakopoulos, R. (2009). Two-dimensional conditional simulations based on the wavelet decomposition of training images. Mathematical Geosciences. 41 (6): 679-701.

[26]. Parra, A. and Ortiz, J.M. (2011). Adapting a texture synthesis algorithm for conditional multiple point geostatistical simulation. Stochastic environmental research and risk assessment. 25 (8): 1101-1111.

[27]. Kalantari, S. and Abdollahifard, M.J. (2016). Optimization-based multiple-point geostatistics: A sparse way. Computers & Geosciences. 95: 85-98.

[28]. Efros, A.A. and Leung, T.K. (1999). Texture synthesis by non-parametric sampling, Conference in iccv, 20 September, IEEE. pp. 1033-1046.

[29]. Pourfard, M., Abdollahifard, M.J., Faez, K., Motamedi, S.A. and Hosseinian, T. (2017). PCTO-SIM: Multiple-point geostatistical modeling using parallel conditional texture optimization. Computers & Geosciences. 102: 116-138.

[30]. Mariethoz, G. and Lefebvre, S. (2014). Bridges between multiple-point geostatistics and texture synthesis: Review and guidelines for future research. Computers & Geosciences. 66: 66-80.

[31]. Efros, A.A. and Freeman, W.T. (2001). Image quilting for texture synthesis and transfer, 28th annual conference on Computer graphics and interactive techniques. pp. 341-346.

[32]. El Ouassini, A., Saucier, A., Marcotte, D. and Favis, B.D. (2008). A patchwork approach to stochastic simulation: a route towards the analysis of morphology in multiphase systems. Chaos, Solitons & Fractals. 36 (2): 418-436.

[33]. Faucher, C., Saucier, A. and Marcotte, D. (2014). Corrective pattern-matching simulation with controlled local-mean histogram. Stochastic environmental research and risk assessment. 28 (8): 2027-2050.

[34]. Gardet, C., Le Ravalec, M. and Gloaguen, E. (2016). Pattern-based conditional simulation with a raster path: a few techniques to make it more efficient. Stochastic environmental research and risk assessment. 30 (2): 429-446.

[35]. Moura, P., Laber, E., Lopes, H., Mesejo, D., Pavanelli, L., Jardim, J., Thiesen, F. and Pujol, G. (2017). LSHSIM: A Locality Sensitive Hashing based method for multiple-point geostatistics. Computers & Geosciences. 10: 749-760.

[36]. Bavand Savadkoohi, M., Tokhmechi, B., Gloaguen, E. and Arab-Amiri, A.R. (2018). New simulation of lithology based on wavelet decomposition, 19th symposium of JSTE, Institut National de la Recherche Scientifique, Québec, Canada.

[37]. Theodoridis, S. and Koutroumbas, K. (2008). Pattern recognition, Academic Press, New York.

[38]. Di Stefano, L. and Mattoccia, S. (2003). Fast template matching using bounded partial correlation. Machine Vision and Applications. 13 (4): 213-221.

[39]. Fan, G. and Xia, X.G. (2003). Wavelet-based texture analysis and synthesis using hidden Markov models. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications. 50 (1): 106-120.

[40]. Mallat, S. (1999). A wavelet tour of signal processing, San Diego, CA: Academic Press.

[41]. Crouse, M.S., Nowak, R.D. and Baraniuk, R.G. (1998). Wavelet-based statistical signal processing using hidden Markov models. IEEE Transactions on signal processing. 46 (4): 886-902.

[42]. Tran, T., Mueller, U. and Bloom, L. (2002). Multi-level Conditional Simulation of Two-Dimensional Random Processes using Haar Wavelets. Proceedings Geostatistical Association Symposium Perth. pp. 56-78.

[43]. Remy, N., Boucher, A. and Wu, J. (2009). Applied geostatistics with SGeMS: a user's guide. Cambridge University Press.

[44]. Allard, D. (1994). Simulating a geological lithofacies with respect to connectivity information using the truncated Gaussian model, In: Soares, A. O. (Ed.). Geostatistics Troia'92. Springer. 197 P.

[45]. Western, A.W., Blöschl, G. and Grayson, R.B. (2001). Toward capturing hydrologically significant connectivity in spatial patterns. Water Resources Research. 37 (1): 83-97.

یک ارزیابی جامع بین دو الگوریتم شبیه‌سازی چندنقطه‌ای فلیتر-مبنا

مجتبی باوندسوادکوهی[1*]، بهزاد تخم‌چی[1]، اروان گلوآگوان[2] و علیرضا عرب‌امیری[1]

۱- دانشکده مهندسی معدن، نفت و ژئوفیزیک، دانشگاه صنعتی شاهرود، ایران

۲- مرکز آب، زمین و محیط‌زیست دانشگاه INRS، کبک، کانادا

ارسال ۲۰۱۸/۸/۷، پذیرش ۲۰۱۸/۸/۲۹

* نویسنده مسئول مکاتبات: mojtababavand@yahoo.com

چکیده:

گرافیک‌های کامپیوتری ابزارهای متنوعی برای بهبود بازتولید آماره‌های مرتبه بالاتری که زمین‌آمار دونقطه‌ای به طور دقیقی قادر به تولید آن نیست، پیشنهاد می‌کنند. تقریباً تمامی الگوریتم‌های جدید زمین‌آمار چند نقطه‌ای (MPS) تا حدودی از این تکنیک‌ها برای افزایش دقت و راندمان شبیه‌سازی استفاده می‌کنند. در این پژوهش، یک مقایسه دقیق بین الگوریتم اخیراً توسعه داده شده، CCWSIM و یک الگوریتم شناخته شده دیگر شبیه‌سازی چند نقطه‌ای، FILTERSIM، انجام شده است. مهم‌ترین دلیل مقایسه این دو الگوریتم، بهره‌گیری هر دو الگوریتم از یک سری فیلتر برای استخراج ویژگی است. به طور دقیق‌تر، هر دو الگوریتم فاصله شباهت (عدم شباهت) بین پیشامدهای داده‌ای در تصویر آموزشی و شبکه شبیه‌سازی را در فضای ویژگی محاسبه می‌کنند. به منظور مقایسه دقت محاسباتی دو الگوریتم، برخی آماره‌ها مانند نسبت رخساره‌ها، واریوگرام و تابع پیوستگی محاسبه شده است. نتایج حاصل از شبیه‌سازی سازگاری عالی بین نتایج تحقق‌های الگوریتم CCWSIM و تصویر آموزشی در مقایسه با FILTERSIM را نشان می‌دهد. علاوه بر این، به طور میانگین زمان اجرای شبیه‌سازی برای هر تحقق CCWSIM حداقل ۱۰ بار کمتر از زمان شبیه‌سازی FILTERSIM است.

کلمات کلیدی: شبیه‌سازی چند نقطه‌ای، الگوریتم‌های فیلتر-مبنا، شبیه‌سازی زمین‌آماری، الگوریتم فیلترسیم، الگوریتم CCWSIM.