Comput Geosci (2012) 16:779–797
DOI 10.1007/s10596-012-9287-1

ORIGINAL PAPER

Multiple-point geostatistical modeling based on the cross-correlation functions

Pejman Tahmasebi · Ardeshir Hezarkhani · Muhammad Sahimi

Received: 11 June 2011 / Accepted: 10 February 2012 / Published online: 8 March 2012

Abstract An important issue in reservoir modeling is accurate generation of complex structures. The problem is difficult because the connectivity of the flow paths must be preserved. Multiple-point geostatistics is one of the most effective methods that can model the spatial patterns of geological structures, which is based on an informative geological training image that contains the variability, connectivity, and structural properties of a reservoir. Several pixel- and pattern-based methods have been developed in the past. In particular, pattern-based algorithms have become popular due to their ability for honoring the connectivity and geological features of a reservoir. But a shortcoming of such methods is that they require a massive data base, which make them highly memory- and CPU-intensive. In this paper, we propose a novel methodology for which there is no need to construct pattern data base and small data event. A new function for the similarity of the generated pattern and the training image, based on a cross-correlation (CC) function, is proposed that can be used with both categorical and continuous training images. We combine the CC function with an overlap strategy and a new approach, adaptive recursive template splitting along a raster path, in order to develop an algorithm, which we call cross-correlation simulation (CCSIM), for generation of the realizations of a reservoir with accurate conditioning and continuity. The performance of CCSIM is tested for a variety of training images. The results, when compared with those of the previous methods, indicate significant improvement in the CPU and memory requirements.

Keywords Multiple-point geostatistics · Cross-correlation · Training image · Conditional simulation · Adaptive recursive template splitting

P. Tahmasebi · A. Hezarkhani
Department of Mining, Metallurgy and Petroleum Engineering, Amir Kabir University of Technology, Tehran, Iran

M. Sahimi (✉)
Mork Family Department of Chemical Engineering and Materials Science, University of Southern California, 90089-1211 Los Angeles, CA, USA
e-mail: moe@usc.edu

1 Introduction

A most important problem in modeling of large-scale porous media, such as oil reservoirs, and simulation of fluid flow therein is the identification of the flow paths that consist of the zones with significant permeabilities and are distributed in the interwell regions. Kriging [39, 63] has been used for many years for extrapolating the existing data to the interwell regions. But because kriging produces permeability and porosity fields that are excessively smooth [29, 39, 44, 63], it is not an effective tool for modeling of the interwell regions of highly heterogeneous reservoirs. Therefore, methods based on stochastic or geostatistical simulation have been developed to overcome the problems associated with the smoothness problem [39, 40, 44, 52, 63]. Such methods generate many realizations of a reservoir that can reproduce the broad heterogeneity and spatial uncertainty in the properties of the reservoir, indicated by the existing data for the reservoir [7, 63]. Due to such a distinct advantage, geostatistical simulations are used

frequently in mining, reservoir modeling, and hydrogeology. In general, such methods are classified into three main groups:

In one group are object-based simulation methods that consist of marked point process [46] and the Boolean method in which the reservoir is represented as a collection of independent stochastic objects whose geometries are based on a specific statistical distribution [19, 30, 35, 64].
Pixel-based methods belong to the second group. They are based on defining an array of points or pixels in a regular grid [11, 18, 27], with the pixels representing various properties of a reservoir.
Such methods as conditional simulations via the LU decomposition of the covariance matrix [16], sequential Gaussian simulation [21], frequency-domain simulation [4, 12], simulated annealing [17, 31, 63], and the genetic algorithm (for a review, see [63]) are in the last group. The latter two methods are optimization techniques that determine the most plausible realization of a reservoir, given some data.

Each of such methods has some advantages, as well as limitations. For example, object-based simulations are accurate in reproducing geological features, but they are CPU-intensive and encounter difficulty when one tries to condition the model to dense hard (quantitative) data and integrate them with soft (qualitative) data. Pixel-based simulations work on one pixel at a time and, thus, are accurate for data conditioning, but because they use variograms that represent two-point statistics, they cannot capture complex and curvilinear features and thus fail when the models that they generate are used in the simulation of flow and transport problems in reservoirs. In addition, they are usually slow.

The aforementioned shortcomings have led to the development of a more accurate method, based on multiple-point statistics (MPS) [29, 41, 65], which is an accurate algorithm for addressing some of the problems. The essence of the MPS method is to use useful features of both the variogram-based and object-based algorithms by using a tool—a training image (TI) of the system—that is the analog of variogram/covariance models in geostatistical models that utilize two-point statistics. In other words, TI is the conceptual information that depicts the subsurface heterogeneity and/or a prior numerical geological feature that mainly represents the spatial continuity, patterns, and distribution of the available features in a reservoir. It is then used for computing multiple-point properties of the system. Therefore, because the MPS method considers more than two points at a time, one may expect the complex facies in a reservoir to be reproduced more accurately.

However, the original MPS algorithm proposed by Guardiano and Srivastava [29] was impractical because to condition each new data event, the method had to scan the TI anew, and consequently, it was extremely CPU demanding. Note that we use the terminology of "data event" because during the simulations, the data change as the MPS algorithm progresses. Strebelle [68] proposed an alternative method—single normal equation simulation (SNESIM)—by using a search tree structure that is an efficient method of accessing high-dimensional data that can overcome the problem associated with the original MPS method [49, 59]. By using a search tree, the TI is scanned once, and all the replicas of the patterns are stored. Therefore, it is possible to apply SNESIM to 3D reservoirs [1, 10, 32, 33, 49, 69] and hydrogeological modeling [13, 25, 37, 60, 61]. However, SNESIM also suffers from intensive memory and CPU requirements, when used in the simulation of large 3D reservoirs. In addition, SNESIM is impractical for continuous TI and co-simulation (see below). One can improve SNESIM by several method; we will briefly describe such methods shortly.

Several methods have been proposed for reproducing the continuity of a TI by incorporating the multiple-point information, but they still have many limitations. For example, when one uses the simulated annealing method, one can generate a realization of a reservoir based on the previously extracted TI and some inferred MPS that are imposed as constraints and must be honored [17, 24]. Other iterative methods, such as Markov random fields [73], artificial neural networks [8, 9], and directional Metropolis–Hastings algorithm [14, 74, 75], and methods that belong to the Markov chain family, such as the Gibbs sampler [50] and mesh models [46], have been proposed, but most of them are highly CPU demanding and their convergence to the solution is slow. Other ideas, such as the integration of MPS with traditional simulation, e.g., indicator probabilities [55, 56] using the texture synthesis that is borrowed from computer vision field [15, 58], have also been studied. Dimitrakopoulos et al. [20] used spatial cumulants for modeling complex non-Gaussian and nonlinear phenomena and asserted that the method can generate nonlinear patterns and connectivity of highly permeable zones. Generally speaking, most of such works tried to capture the essential features of such systems by using a considerable amount of data.

As mentioned earlier, SNESIM has certain limitations, the most important of which is its CPU requirement. In addition, if the TI has a large number of facies (say, more than 4), it becomes impractical for 3D reservoirs and continuous applications. Some venues have been used, however, to overcome the limitations. For example, Strebelle [68] used multiple grids to capture the large-scale structures and connectivity of a reservoir, but because in his method it is not possible to access the coarse grid throughout the fine grid, the algorithm tends to generate some artifacts.

On the other hand, Journel [41] proposed the so-called Tau model by which one combines two data events B and C in order to assess an unknown event A through its conditional probability $P(A|B, C)$, given that the conditional probabilities $P(A|B)$ and $P(A|C)$ can be evaluated. But the model though elegant requires estimation of a crucial parameter τ. To address the problem with the required large memory, a method was proposed by Straubhaar [66, 67], which is called the Impala model that represents an extension of SNESIM and uses the Tau model for integration of hard and soft data with a TI in a probabilistic model, in order to construct a reservoir's model. In the Impala model, the allocated lists, rather than a search tree, are used to store the replicas of the patterns generated by the MPS method. One stores and then sorts the events in a list, which improves the computational speed, and decreases the computer memory requirements. Moreover, by using a portion of the sorted list, one constructs the conditional probability distribution function.

Following such works, Vargas-Guzmán [76] (see also [77]) proposed a new method, referred to as the Kappa model, in order to quantify the probability distribution of the joint occurrence of multifacies events. The model is based on both the cumulants and co-cumulants and was used for modeling of the chronostratigraphic sequences of sediments and rock. Higher-order cross-correlations (CC; see below) were also utilized as a product of residual random variables on the template, which indicated that studying the residual cumulants for the correlations may be a useful approach to such problems.

Alternatives to SNESIM have also been proposed. For example, the SIMPAT method was proposed by Arpat [2] and Arpat and Caers [3], which is based on patterns and a distance function, instead of pixels as in SNESIM, and does not use any probabilistic formulation. Such improvements helped the problem of the continuity of the pattern and decreased the computer memory requirement. SIMPAT is, however, highly CPU-intensive, particularly for 3D simulations. For this reason, a filter-based classification of TI patterns (FILTERSIM) was proposed [79] that uses some predefined filters in order to reduce the dimension of the data and then classifies the patterns into some local classes. In other words, the filters carry out a dimensional reduction and partition and summarize a high-dimensional data set into small vectors. Then, by using a classification method, such as cross-partitioning or k-means, the filters classify and group the data into predefined classes and represent the average of each class by a prototype that helps the algorithm to be much faster. To speed up the computations and reduce memory requirement, Wu et al. [78] applied the filters to both the local data and the prototypes and obtained more accurate results using shorter simulations. Gloaguen and Dimitrakopoulos [26], on the other hand, quantified the spatial dependence of TI by using the wavelet transformation (for a review of the wavelet transformations of their properties, see, for example, [62]) and computing the wavelet coefficients at various scales and carried out both 2D unconditional and conditional simulations.

Filter-based algorithms do have some problems. For example, in FILTERSIM, the most similar pattern is selected randomly form its prototype, which may not be the most suitable one. Moreover, the filters reduce the impact of data with large values. The realizations may also be strongly affected by clustering. Such problems may lead to a realization that is devoid of the rich patterns expected for a highly heterogeneous reservoir. Honarkhah and Caers [35] presented an efficient solution based on the SIMPAT by first representing the pattern in a Cartesian space by multidimensional scaling and then improving the pattern classification by dimensional reduction in the kernel space using clustering and other features of FILTERSIM. While it appears that their method improves the connectivity of the patterns and reduces the CPU requirement, some significant discontinuity in the patterns that are obtained can still be identified. Mariethoz et al. [51] proposed the direct sampling (DS) method in which as soon as the pattern that is most similar to the given data event in the TI is identified, it is inserted directly into the grid that is being constructed. In this method, there is no need to use multiple grids, as it works with both categorical and continuous variables. However, several user-dependent parameters are used in the method that must be tuned.

One may view pattern-based simulation as the analog of template-matching problem in speech recogni-

tion, robot vision, and/or video coding. Furthermore, there are several similarity or dissimilarity distance measures, such as Euclidean, Frobenius, or Manhattan distances, which can be used. They are not, however, adequate in that most of them are involved with some complex patterns that may change over time. For example, in the text-matching problem, a specific word may be written in different styles by different persons. Thus, one must develop a method that can handle such shortcomings. One such method is a measure based on the correlations between the various zones. In this paper, we propose a MPS method based on the cross-correlation measures that replaces the distance functions used previously. The method generates very accurate realizations of a reservoir in terms of its connectivity and is more efficient computationally than the previous methods. We refer to the method as cross-correlation simulation (CCSIM) method.

The rest of this paper is organized as follows: We first describe the cross-correlation function that is utilized in the simulation. Section 3 describes the new algorithms for both unconditional and conditional simulations with a raster path. The performance of the algorithms is demonstrated in Section 4, where it is also compared with that of FILTERSIM and some of the other previous methods. The paper is summarized in Section 5, where we enumerate the advantages of the CCSIM.

2 Cross-correlation-based simulation

Before describing the new method, let us define the notations that are used in the rest of this paper. **G** represents the grid used in the simulation. The template by which the TI is scanned is denoted by **T**. $\mathrm{dev}_T(\mathbf{u})$ is the data event at **u** using template **T**, while $\mathrm{hdev}_T(\mathbf{u})$ represents the hard—quantitative—data event. Finally, the overlap region between dev_T and the previously simulated pattern is denoted by OL.

Several techniques have been proposed in recent years for determining the most similar and/or matched patterns (see, for example, [2, 3, 6, 35, 57, 70, 71]). Those that are based on the transform function, although useful, have some shortcomings. For example, some of such techniques are applicable to only binary images. If they are to be used for a multiple-category TI, they have to separate it into its categorical style and then simulate it. They are also dependent upon on the size of the scanning **T**. On the other hand, correlation-based methods have been used extensively in several distinct fields, such as object recognition and signal processing [5, 22, 28, 53, 54]. In the present paper, we use the CC functions along a raster path and, as described below, combine it with efficient strategies to honor the continuity and pattern reproducibility, in order to generate realizations of the system that match the TI.

In the present paper, we use the CC functions along a raster path and, as described below, combine it with efficient strategies to honor the continuity and pattern reproducibility, in order to generate realizations of the system that match a given TI. We aim to determine a pattern or a collection (set of realizations) of patterns that match a given data event that is contained in a TI. We do this based on a CC function, which is a measure of the similarity of two images. Suppose that $\mathrm{TI}(x, y)$ represents the value of a quantity at point (x, y) of a TI of size $L_x \times L_y$, with $x \in \{0, \cdots, L_x - 1\}$ and $y \in \{0, \cdots, L_y - 1\}$. The method is based on scanning the given TI, focusing on a portion $\mathrm{dev}_T(\mathbf{u})$ with a size $\ell_x \times \ell_y$, in order to determine the pattern in TI that matches it. For this purpose, we use a CC function to quantify the similarity between TI and dev_T, defined by [22]

$$\mathcal{C}(i, j) = \sum_{x=0}^{\ell_x - 1} \sum_{y=0}^{\ell_y - 1} \mathrm{TI}(x + i, y + j)\mathrm{dev}_T(x, y) \,, \tag{1}$$

where i and j represent the shift steps in the x and y directions, with $0 \le i < L_x + \ell_x - 1$ and $0 \le j < L_y + \ell_y - 1$. On the other hand, a Euclidean distance between each point (i, j) of TI and dev_T may be defined through the usual relation,

$$d^2(i, j) = \sum_{x=0}^{\ell_x - 1} \sum_{y=0}^{\ell_y - 1} \left[\mathrm{TI}(x + i, y + j) - \mathrm{dev}_T(x, y)\right]^2 \,. \tag{2}$$

If we expand Eq. 2, we obtain

$$d^2(i, j) = \sum_{x=0}^{\ell_x - 1} \sum_{y=0}^{\ell_y - 1} \left[\mathrm{TI}^2(x + i, y + j) - 2\mathrm{TI}(x + i, y + j) \times \mathrm{dev}_T(x, y) + \mathrm{dev}_T^2(x, y)\right] \,. \tag{3}$$

Since $\mathrm{dev}_T^2(x, y)$ is constant, if $\sum_{x=0}^{L_x - 1} \sum_{y=0}^{L_y - 1} \mathrm{TI}^2(x + i, y + j)$ (which is sometimes referred to as the "energy" of the local image) is also roughly constant, then the distance is directly expressible in terms of the CC function. It is, therefore, not difficult to see that computations with Eq. 1 is less CPU-intensive than those with

Eq. 3. This aspect is one of the differences between the CCSIM and the previous MPS methods. In addition, the CC function may also be expressed in a normalized form, defined by

$$\mathcal{C}_N(i,j)=\frac{\sum_{x=0}^{\ell_x-1}\sum_{y=0}^{\ell_y-1}\{[\mathrm{TI}(x+i,y+j)-\langle \mathrm{TI}_{i,j}\rangle][\mathrm{dev}_T(x,y)-\langle \mathrm{dev}_T\rangle]\}^2}{\left\{\left(\sum_{x=0}^{\ell_x-1}\sum_{y=0}^{\ell_y-1}[\mathrm{TI}(x+i,y+j)-\langle \mathrm{TI}_{i,j}\rangle]^2\right)\left(\sum_{x=0}^{\ell_x-1}\sum_{y=0}^{\ell_y-1}[\mathrm{dev}_T(x,y)-\langle \mathrm{dev}_T\rangle]^2\right)\right\}^{1/2}}\,. \tag{4}$$

Here, $\langle \mathrm{TI}_{i,j}\rangle$ is the local mean of $\mathrm{TI}(x, y)$ over the area of dev_T, shifted to (i, j):

$$\langle \mathrm{TI}_{i,j}\rangle=\frac{1}{\ell_x\ell_y}\sum_{x=i}^{i+\ell_x-1}\sum_{y=j}^{j+\ell_y-1}\mathrm{TI}(x,y)\,. \tag{5}$$

In a similar fashion $\langle \mathrm{dev}_T\rangle$, the local mean of dev_T is defined. Equations 1 and 4 both indicate that the desired position of (i, j)—the best match with the TI—is the one that maximizes $\mathcal{C}(i, j)$ or $\mathcal{C}_N(i, j)$. The two equations are essentially the same, except that the normalized CC function is less sensitive to changes of illumination that is not related to a reservoir's images. In the present study, we used Eq. 1, due to its simplicity and smaller computational cost.

If an image is too complex, such as those for a heterogeneous reservoir that contains facies and fractures, the traditional Euclidean distance function will not be able to capture the correct variations in the image and tend to generate a pattern that is quite different from the desired one. It is precisely for such cases that the use of $\mathcal{C}(i, j)$ is appropriate [36, 73] because it yields results that preserve the global variations and continuity of the image. Figure 1 summarizes the way CC function is utilized in the simulations.

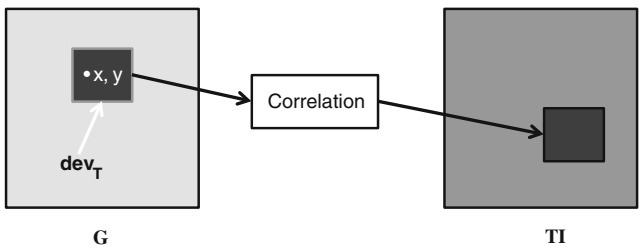

Fig. 1 The schematic of determining a matching pattern by the cross-correlation method

3 Algorithms

We have developed two algorithms for conditional and unconditional simulations, which are shown below.

3.1 Unconditional simulation

The CCSIM algorithm that we develop in this paper is based on a raster (1D) path. Raster paths have been used in the past [14, 15, 23, 51, 58, 72] and yielded accurate results. A problem with the previous applications was, however, preserving the continuity of the image of the reservoir in the sectors for which hard data (HD) are available. In this paper, we present a method to address the problem.

The algorithm is summarized in Fig. 2. It begins at the origin of the defined **G** (part I in Fig. 2). It then proceeds along a raster path on **G**. Since the simulation is unconditional, at the beginning **G** is completely unknown and contains no hard data. A random patch equal to **T** is selected and inserted into the first dev_T. Next, the dev_T is used along an axis (vertical axis in Fig. 2) and according to the raster path. As mentioned earlier, OL represents the overlap region between dev_T and the previously simulated pattern. Thus, one attempts to locate a patch that matches its bottom overlap, OL_b. The OL region is defined to preserve the continuity near the boundaries and to generate a homogeneous realization. It is shown in Fig. 3, which is the only section comparable with the TI, and thus, we ignore the rest of dev_T. Recall that the aim here is to determine the best matched pattern whose bottom section is correlated with OL_b; see Fig. 3a. If several patterns have the same degree of correlations with OL_b, one of them is selected at random. The procedure continues until the simulation finishes inspecting the first line of the raster path, i.e., the algorithm reaches Section 3 in Fig. 2. Then, along and beside the first replaced patch, the algorithm moves forward (segment IV in Fig. 2). At this point, either the first pattern is

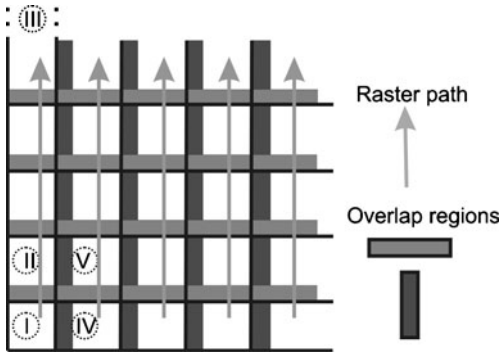

Fig. 2 The raster path with various overlap regions

not selected randomly or it has the OL_s region on its left side by which the patch is identified; see Fig. 3b. Next, the algorithm enters a more complex section—segment V in Fig. 2—which is a section in which dev_T has two overlap regions OL_d at the bottom and on the left side. Therefore, these regions are considered in order to identify the matched pattern; see Fig. 3c.

Thus, the algorithm for unconditional simulation may be summarized as follows:

- Define a raster path on the initial grid **G** to visit its nodes according to a given **T** and OL.
- At each visited node, extract the dev_T that should consist of new nodes and the OL:
  - Separate the OL from dev_T and use it to calculate the CC function.
  - Determine a new image by which one is able to compare, based on both the CC function and visualization, the similarities between the OL region and the TI.
- Based on Eq. 1 and a given acceptance threshold δ for the OL, determine the matched pattern. An acceptance threshold $\delta = 0$ implies that the error in the similarity of the OL region of the current pattern with the original one is zero. If $\delta > 0$, one may find an ensemble of patterns that do not exactly match the TI.
- If several patterns can be listed as the candidates for the matched pattern, select one at random.
- Assign the selected matched pattern to dev_T at all of its nodes.
- Move to another defined node based on the given parameters on the raster path and repeat the above procedures until all the nodes of the grid **G** have a value.

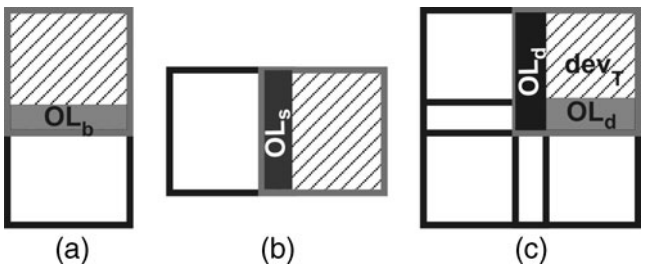

Fig. 3 Three overlap regions that may be created in the raster path simulation, with the overlap region that corresponds to: **a** the bottom region, **b** the left-side region, and **c** to the simulation of the internal nodes

3.2 Conditional simulation

As is well-known, in conditional simulation, some data are known about the initial grid **G**, which help one to generate a realization of the reservoir with smaller uncertainty and a more accurate structure. Conditioning to HD and, consequently, the HD event $hdev_T$ require some changes in the algorithm for unconditional simulation that we already described because in this case the matched pattern must honor the HD, in addition to preserving the continuity of the image. Therefore, one difference between the algorithms for unconditional and conditional simulations is that the latter involves a two-step searching. The first lookup is for identifying the patterns that honor HD, while the second step is to match the dev_T. Thus, the algorithm first computes the cross-correlations in a manner similar to that for unconditional CCSIM. Then, according to a pre-defined acceptance threshold δ, the corresponding CC functions and, thus, their patterns will be re-examined, those that honor the HD are identified, and then one of them is selected at random and inserted into the $hdev_T$.

A few points are worth mentioning here. If $hdev_T$ is empty, then the algorithm proceeds as in the case of unconditional CCSIM. If $hdev_T$ is not totally empty, at least one pattern may be identified that honors the HD and matches the given TI. A problem may arise when the template **T** is very large, or when the TI is not informative enough to support the HD, which can lead to discontinuities and prevent the algorithm from identifying a pattern in the ensemble of the possible patterns that honor the HD. Such a situation, which arises very rarely, can be addressed rather straightforwardly, due to the intrinsic features of the CCSIM algorithm.

One solution is to increase the threshold δ when the algorithm is unable to identify a pattern that honors the HD. In this case, new patterns that have more significant differences with the OL region will be a candidate for inclusion in the ensemble of the possible patterns. The process of increasing δ continues until a proper pattern is identified and inserted in the $hdev_T$.

But by increasing the threshold δ, we also allow the inappropriate patterns to enter the ensemble, which may subsequently lead to the generation of a poor realization of the reservoir that contains discontinuity. To overcome this problem, we may use another raster path in a different direction to improve the quality of the realizations. An alternative approach is to use an adaptive simulation that is shown schematically in Fig. 4. First, the patterns that honor the HD with the initial threshold δ are identified. If, however, no such pattern is found, the sector dev_T is split into smaller data events dev_T^s, and the simulation is continued with the dev_T^s. The splitting into smaller regions continues until the ensemble of the patterns that honor the HD has at least one member.

An important aspect of a reservoir model is the anisotropy. The CCSIM method proposed here can also be used when anisotropy is present. One must first analyze the TI in order to determine the anisotropy direction. Once this is done, the template can be collapsed or expanded in that direction. Other methods can also handle anisotropic patterns; see, for example, the Kappa model of Vargas-Guzmán [77]. Note that the method that we propose in this paper can also be used with secondary or soft-data conditioning. We will report on this aspect of the model in a future paper.

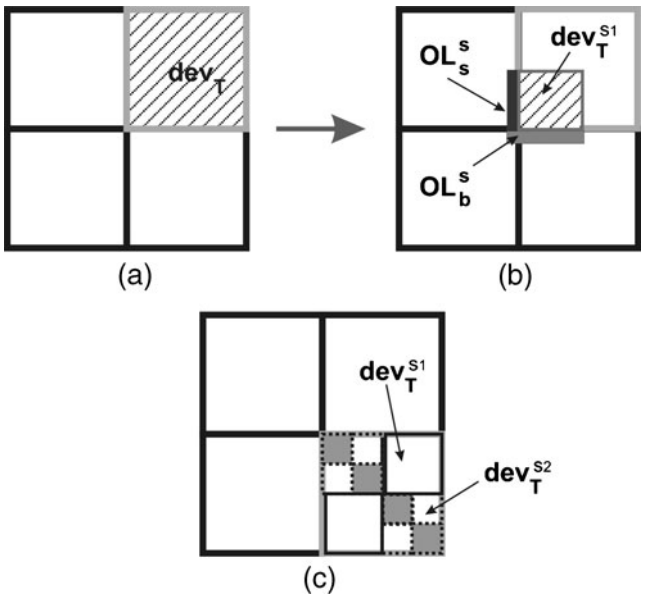

Fig. 4 Adaptive recursive template splitting for CCSIM. The overlap regions are shown by the *shaded gray* for three cases. **a** The prototype of CCSIM without any splitting, but if any patterns in the TI cannot be found, the dev_T will be divided into four sections dev_T^s1, each of which is simulated separately in **b**. If under the new condition no matching pattern can be found, dev_T^s1 is divided into four equal sections dev_T^s2, as shown in **c**. The process continues until a matching pattern is identified

The advantage of the CCSIM is in a *combination* of ideas and concepts, including the raster path, the overlap region, template splitting, and the computational cost of the CC function that is essentially trivial. Using Eq. 1 is less CPU-intensive than Eq. 3. The second advantage is due to using the overlap regions by which only a small part of the data event is used for the similarity computations. Moreover, reservoir patterns appear dissimilar under the Euclidean distance because the comparison is made pointwise and in some complex cases one should use the proximity transform distance function, whereas a human would perceive both patterns as similar [2]. As the human perception is tolerant to translational effects, using the CC should be a better choice than the Euclidean distance.

Furthermore, calculation of the CC function can be carried out in both the spatial and Fourier domains. If the size of the TI is not too large, all the computations can be carried out in the spatial domain, as is done in the present paper. We will demonstrate in a future paper that very significant performance benefits are achieved, however, by computing the CC function in the Fourier space, when the TI is very large and contains millions of grid blocks. Finally, although the differences between the quality of the results obtained with the CC function and the Euclidean distance for relatively TIs are not large, the differences emerge as highly significant when one has TIs that include fractures, multiple facies, etc. In that case, the results obtained with the CC function are much more accurate. We will come back to this point shortly.

4 Results and discussion

We now demonstrate the accuracy and efficiency of the proposed CCSIM algorithm by comparing its performance with that of FILTERSIM, a distance-based algorithm with a number of similarities with the CCSIM, and for which the distance-based calculations are based on filter scores. FILTERSIM is also much faster than the methods based on template pixels (see, for example, [78]). All the simulations were carried out on a Dell notebook computer with a 2.66-GHz CPU processor.

In the discussions that follow, matching refers to quantitative comparison between the generated realization and the TI, but most importantly, it also implies *visual* matching. The reason is that in many of the previous methods, one often obtains an apparent matching between, for example, the variograms of the TI and the realization, as well as other quantitative measures, but there are still striking differences between the image of the generated realization and the TI.

4.1 Unconditional simulations

Several examples are presented to demonstrate the usefulness of the algorithm for unconditional simulations. In what follows, we describe the examples and discuss the results.

Random distribution of circles The TI consists of a random distribution of non-overlapping circles, which is a 2D model of unconsolidated porous media. The size of the TI is 100×100. Figure 5 presents the results obtained with CCSIM algorithm. Three independent realizations were generated with a template **T** of size 15×15 and an overlap region OL of size 4. To make comparison with FILTERSIM on equal footing, we used the same conditions with the FILTERSIM. Therefore, we selected a template **T** of size 15×15 and an inner template **IT** of size 11×11 for FILTERSIM. Since the TI is not very complex, we used $k = 100$ as the number of clusters in FILTERSIM. The size of the data event dev_T for both algorithms was the same.

As shown in Fig. 5, FILTERSIM could not reproduce the given TI correctly, whereas the CCSIM algorithm is able to mimic the patterns in the TI and preserve the randomness of the distribution of the circles. Furthermore, the CCSIM and FILTERSIM algorithms took about 0.14 and 5.4 CPU s, respectively, hence indicating a speed-up factor of almost 40 in the computations. While it is possible to increase the quality of the realizations obtained with the FILTERSIM algorithm by increasing the number of the clusters, the CPU will also increase.

Mixture of shale and sand The TI shown in Fig. 6, which represents a "binary" image of a reservoir consisting of shales and sands, is perhaps a better representative of an actual reservoir with a complex channel structure. The model has been used extensively in the past with algorithms SNESIM, FILTERSIM, and DS. Its size is 250×250. The results obtained with CCSIM are shown in Fig. 6, where they are compared with those obtained with the FILTERSIM algorithm. It is clear that FILTERSIM could capture neither the variability

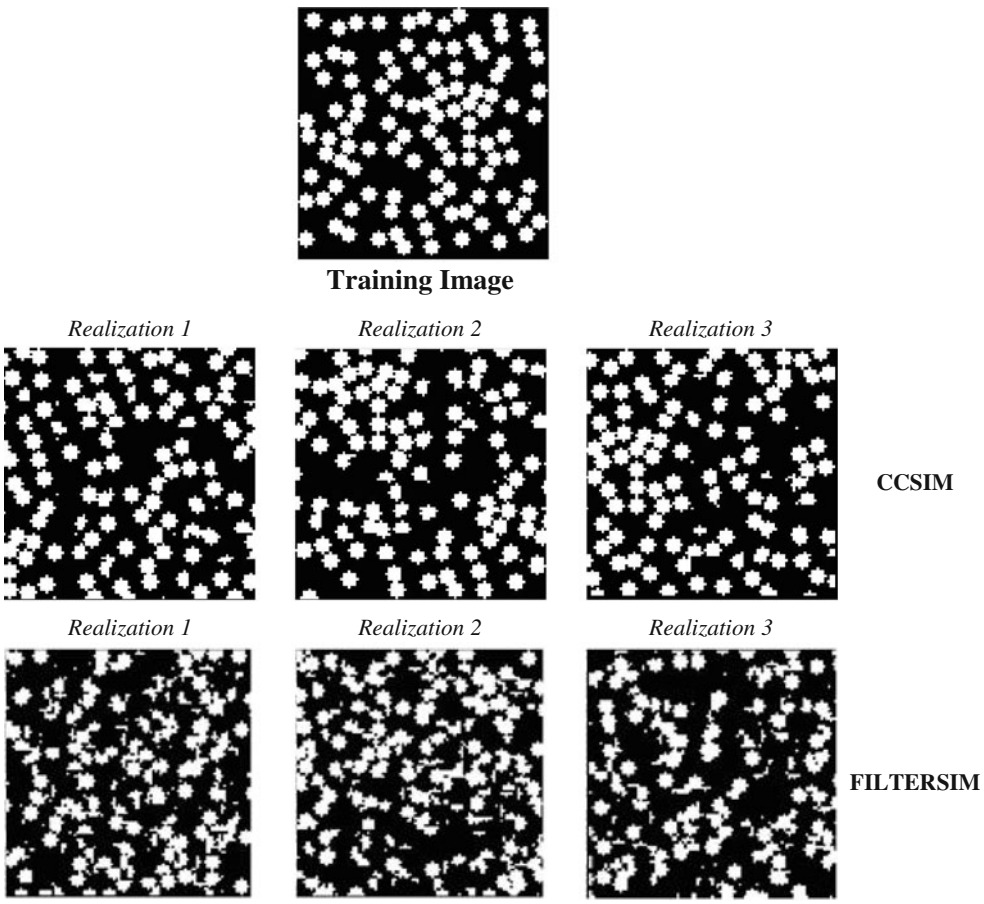

Fig. 5 Comparison of the performance of CCSIM with FILTERSIM for a TI consisting of a random distribution of *circles*

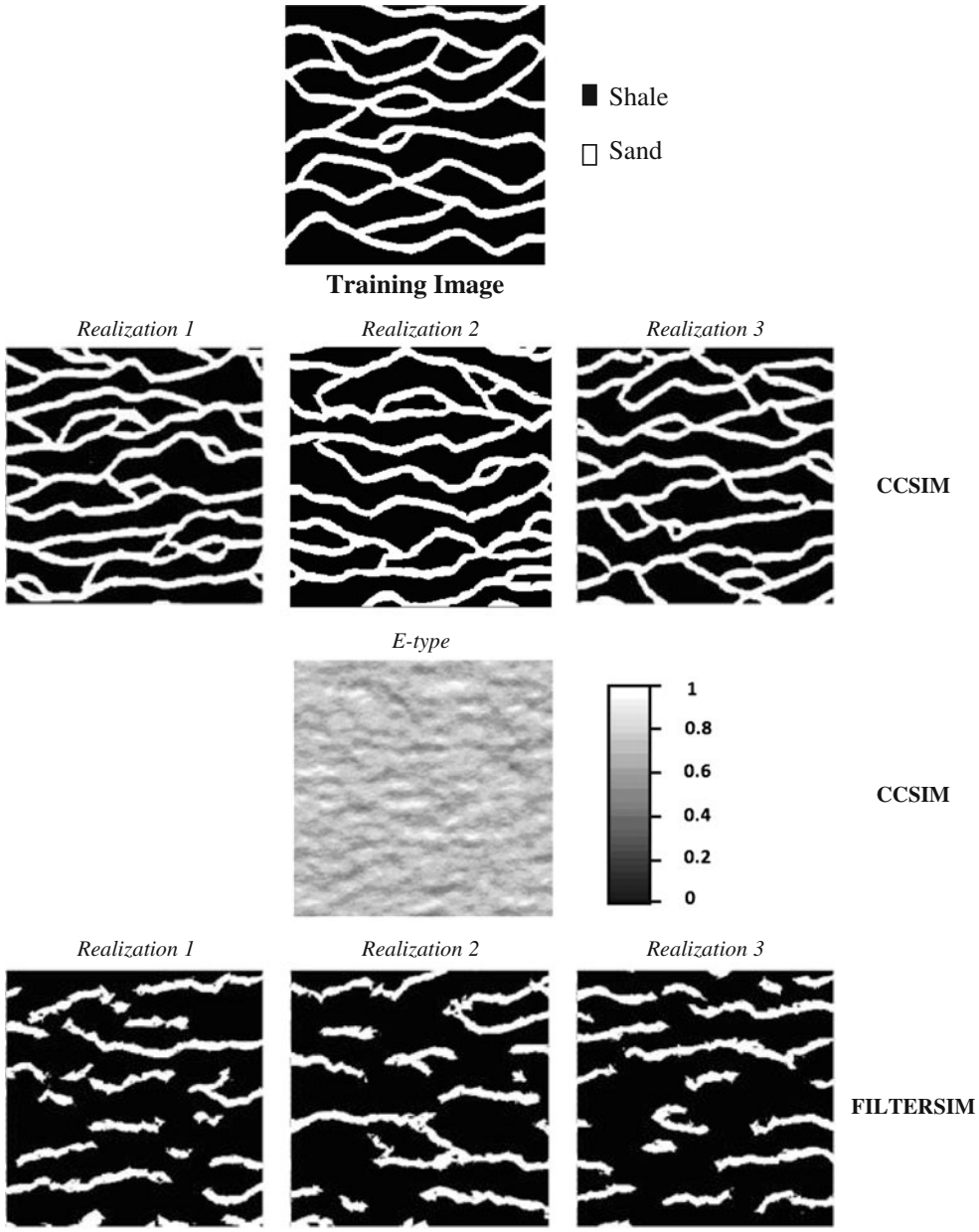

Fig. 6 Comparison of the performance of CCSIM with FILTERSIM for a complex TI with binary channels

nor the connectivity nor the large convoluted features of the TI, whereas the CCSIM algorithm produces very accurate images. Furthermore, in order to make a more precise statistical comparison of the generated realizations with the TI, a map of the spatial distribution of the channels—the so-called *E*-type map—based on single-pixel averaging of 25 realizations was generated. Since the realizations are generated randomly and independently, the *E*-type map dose not possess any structure (correlation). If we assume, for example, that the permeabilities of the background and the channels are, respectively, 0.01 and 100 md, then the mean permeability of the *E*-type map of the generated image will be 0.29 md, which is close to that of the TI that is about 0.276, hence indicating the satisfactory performance of the CCSIM algorithm for reproducing the correct image. Moreover, the CCSIM and FILTERSIM algorithms took about 3 and 342 CPU s, respectively.

We also computed the variograms of the TI and compared them with those of the generated realizations. They were computed in the north–south (NS) and east–west (EW) directions. In addition, multiple-point connectivity was tested statistically. The probability of multiple-point connectivity was computed in a specific direction, and a lag distance was defined as the proportion of all the sequential pixels of size *n* that

belong to a channel in that direction [42]. Thus, one moves in a given direction and counts the number of points that are connected in pixels of size 1, 2, ⋯, and forms their ratios with the total number of points that are within a channel. The results are shown in Fig. 7. It is clear that the CCSIM algorithm is able to reproduce both the variograms and connectivity of the TI. These results should also be compared with those obtained with other algorithms, such as those presented by Honarkhah and Caers [35] who improved SIMPAT (and referred to it as DISPAT), by Mariethoz et al. [51] with the DS, by Wu et al. [78] with the FAST-FILTERSIM, and by Strebelle [68] with the SNESIM. Although most of previous methods are supposed to reproduce the variograms and histogram of the TI, they often suffer from incorrect connectivity. In addition, even at the visual level, the realizations produced by such methods do not often match the TI.

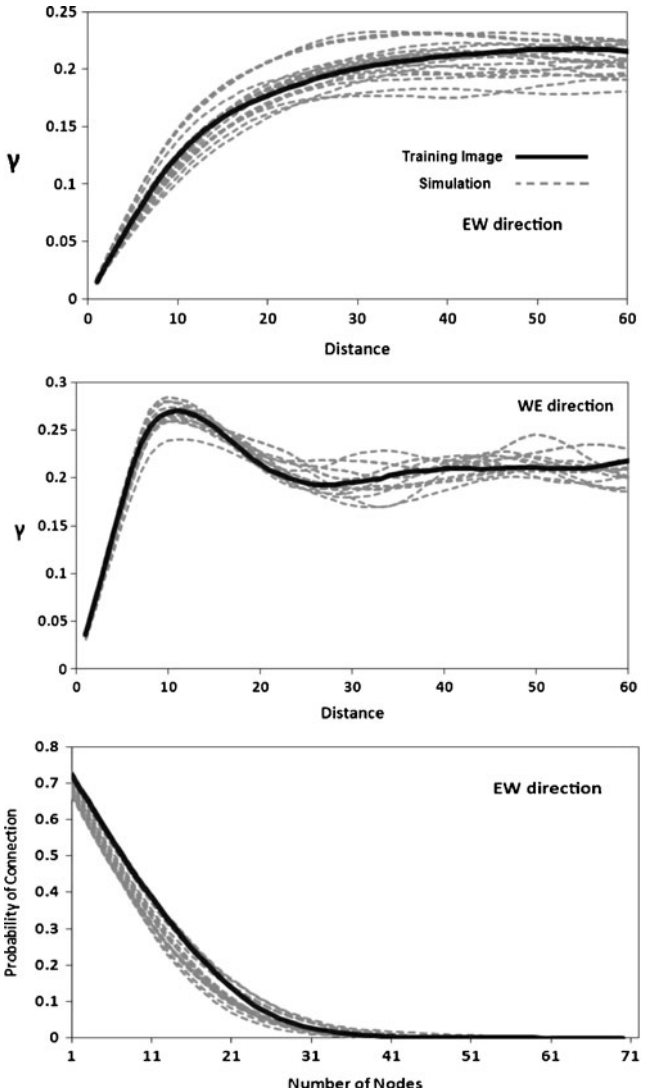

Fig. 7 Comparison of the variograms of several realizations with those of the TI in the EW (*top*) and NS (*middle*) directions and the probability of multiple-point connections of *n* points (*bottom*)

Training images with four distinct facies The performance of the algorithm was also tested by a TI that contained four types of facies. The size of the grid **G**, 150 × 150, is as same as that of the TI. The proportions in the TI of the shale background, sand channel, levee, and crevasse are 0.45, 0.2, 0.2, and 0.15, respectively. In order to compare the results with those obtained with FILTERSIM algorithm, we used a template size of 11 × 11 with an OL size of 4 with the CCSIM algorithm, and a template size of 11 × 11 with the OL size of 7 with FILTERSIM.

The results of the simulations are shown in Fig. 8. Similar to the previous examples, CCSIM algorithm reproduces the essential features of the TI much better than FILTERSIM. The simulations took 0.95 and 125 CPU s with CCSIM and FILTERSIM algorithms, respectively. These results should be compared with those obtained with other methods for similar TIs, given by Wu et al. [78] and Honarkhah and Caers [35]. A comparison with their results indicate that CCSIM algorithm produces much more accurate results, as well as being much more efficient.

Continuous training image We now test the performance of the algorithm with a TI that is much more complex than the previous ones. In this case, the TI was borrowed from Zhang et al. [79] and represents a complex structure of bricks with the size of 200 × 200. Since the TI is very complex, we used a large number of clusters, 300, with FILTERSIM in order to increase its accuracy.

The results are shown Fig. 9. It is clear that CCSIM algorithm is able to reproduce the connectivity and complex features of the TI, whereas FILTERSIM fails almost completely. The simulation times for CCSIM and FILTERSIM were 2.3 and 182 CPU s, respectively. Furthermore, we also computed the histograms (distribution) of the pixels for the TI and the generated realizations. The results, shown in Fig. 10, illustrate that CCSIM is able to produce realizations with a distribution similar to that of the TI. These results should also be compared with similar attempts by Zhang et al. [79] and Mariethoz et al. [51].

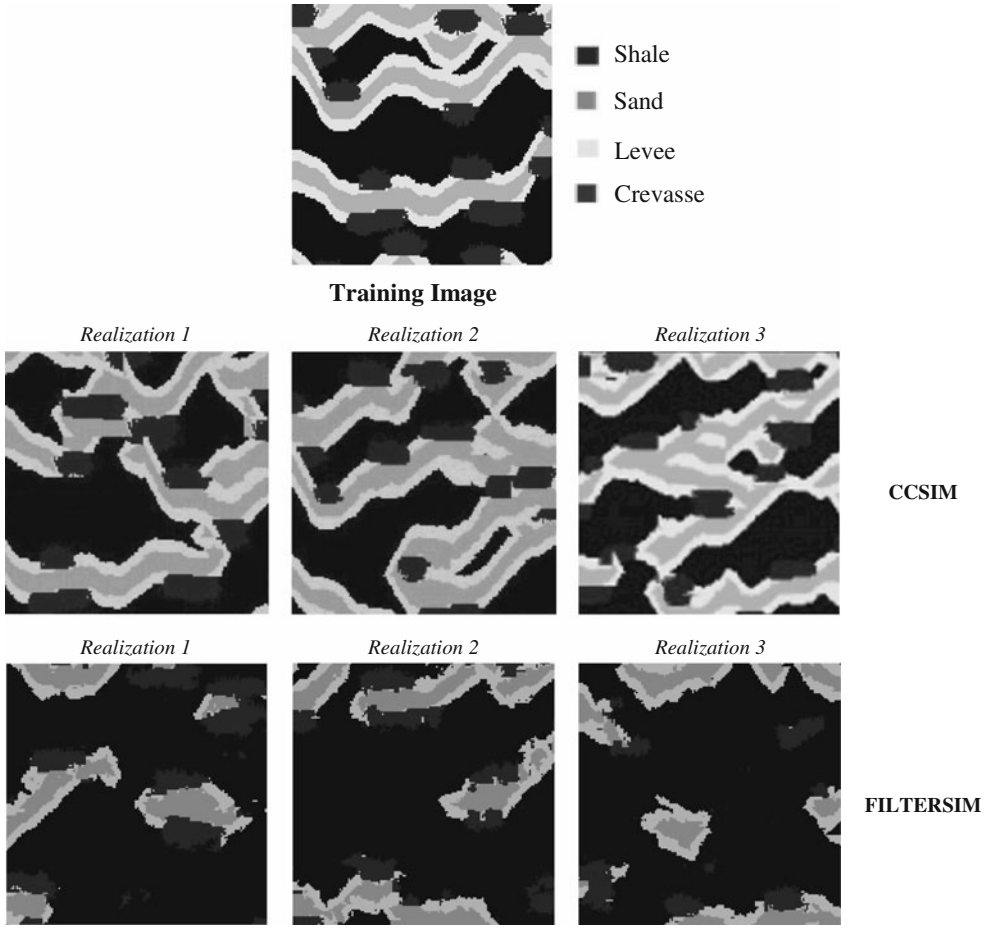

Fig. 8 Comparison of the results, obtained with the CCSIM and FILTERSIM, for a complex TI with four facies

Three-dimensional mixture of shale and sand So far we have tested the performance of CCSIM algorithm using 2D training images. We now present the results for some 3D systems and compare them with those computed with FILTERSIM for a 3D TI of shale and sand with a size of $100 \times 100 \times 15$. The results are shown in Fig. 11, where the realizations, one each obtained with CCSIM and FILTERSIM algorithms, are shown, using a template size of $20 \times 20 \times 7$. To employ CCSIM algorithm, we used an OL region of size $5 \times 5 \times 3$, while in the FILTERSIM algorithm, an inner patch of size $15 \times 15 \times 4$ was set. Figure 11 demonstrates once again that CCSIM algorithm can capture the geological features of a 3D model with extreme channeling and generate a realization that is similar to the TI much more closely than what FILTERSIM generates. Let us also mention that the CCSIM and FILTERSIM realizations took about 5 and 21 CPU min, respectively.

4.2 Conditional simulations

We have also carried out extensive simulations using the CCSIM algorithm, in order to test the accuracy of the method. In what follows, we present and discuss the results.

4.3 Image conditioned with sparse hard data

As the first example, we used 10 data points as the HD in the simulation. Because in this example the HD are sparsely populated and are separated by relatively large distances, we may expect an *E*-type map with more uncertainty. Moreover, in the regions near the HD, the produced realization may have some undesired features, but on the other hand, the uncertainty near the HD should also be smaller with a more accurate structure in the corresponding locations in the *E*-type

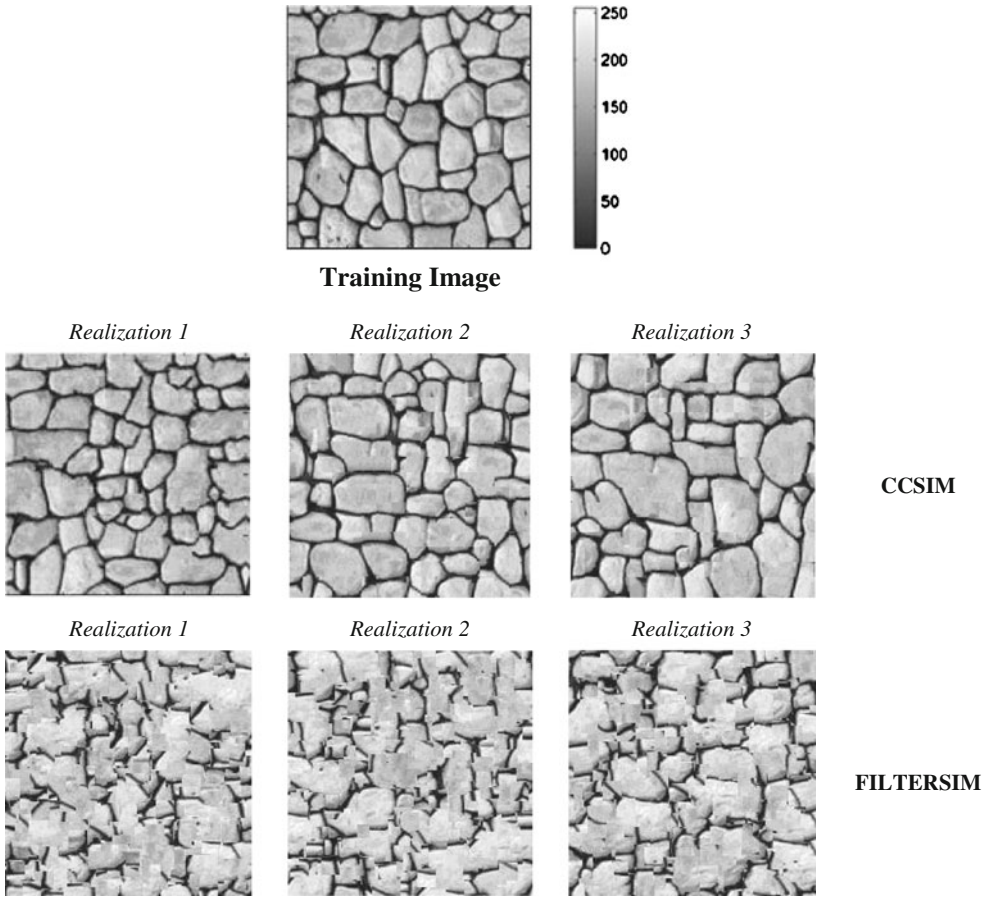

Fig. 9 Comparison of the results, obtained with the CCSIM and FILTERSIM, for a continuous and complex TI

map. The size of the system was 100 × 100. The results are shown in Fig. 12. They indicate that there is some specific structure near the HD and that far from the HD, the uncertainty increases. But, on the other hand, the FILTERSIM algorithm cannot generate any clear structure near the HD and only honors the HD. Since we did not use matching with the TI proportions of sand and shales, the realizations tend to reproduce more shale regions. These results should also be compared with similar attempts by Arpat and Caers [3].

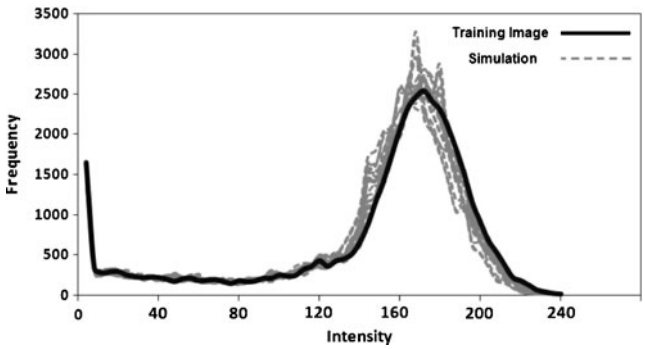

Fig. 10 Comparison of the histograms of continuous TI, shown in Fig. 9, and those of the generated realizations

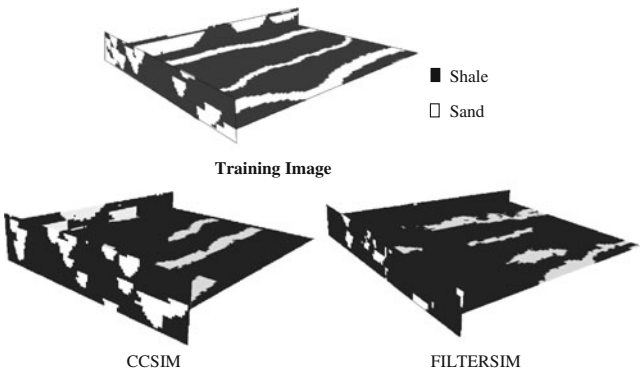

Fig. 11 3D unconditional simulation of a binary TI

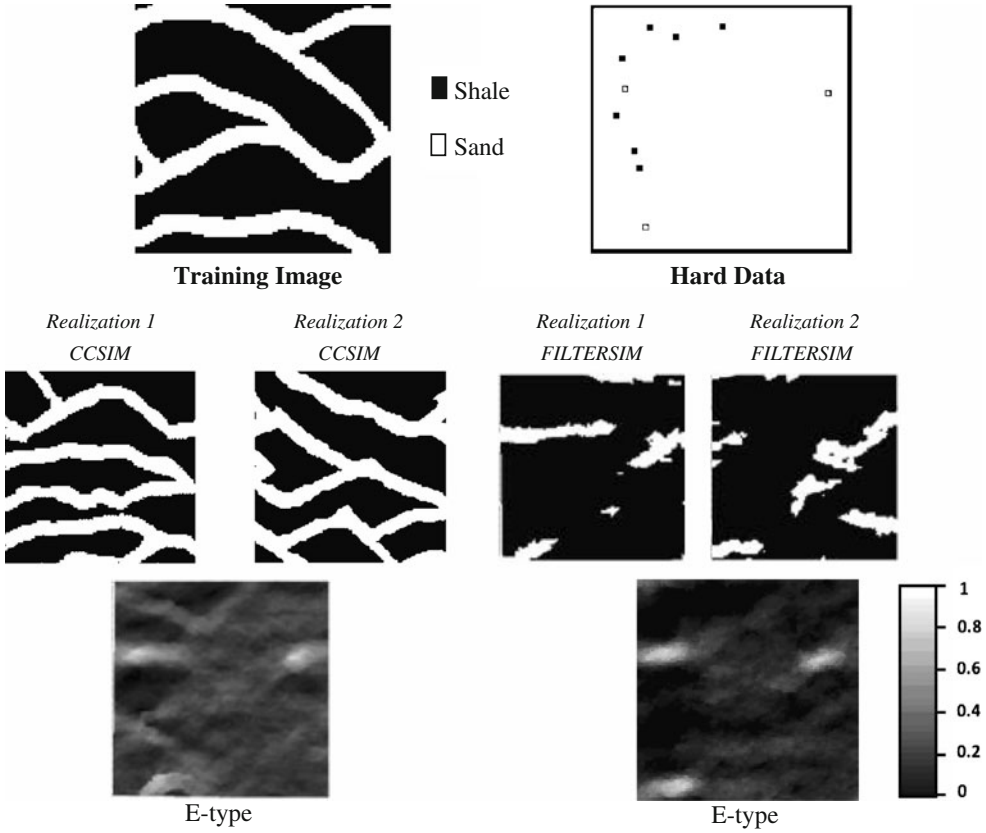

Fig. 12 Comparison of the results with hard-data conditioning with 10 data points, the locations of which are shown

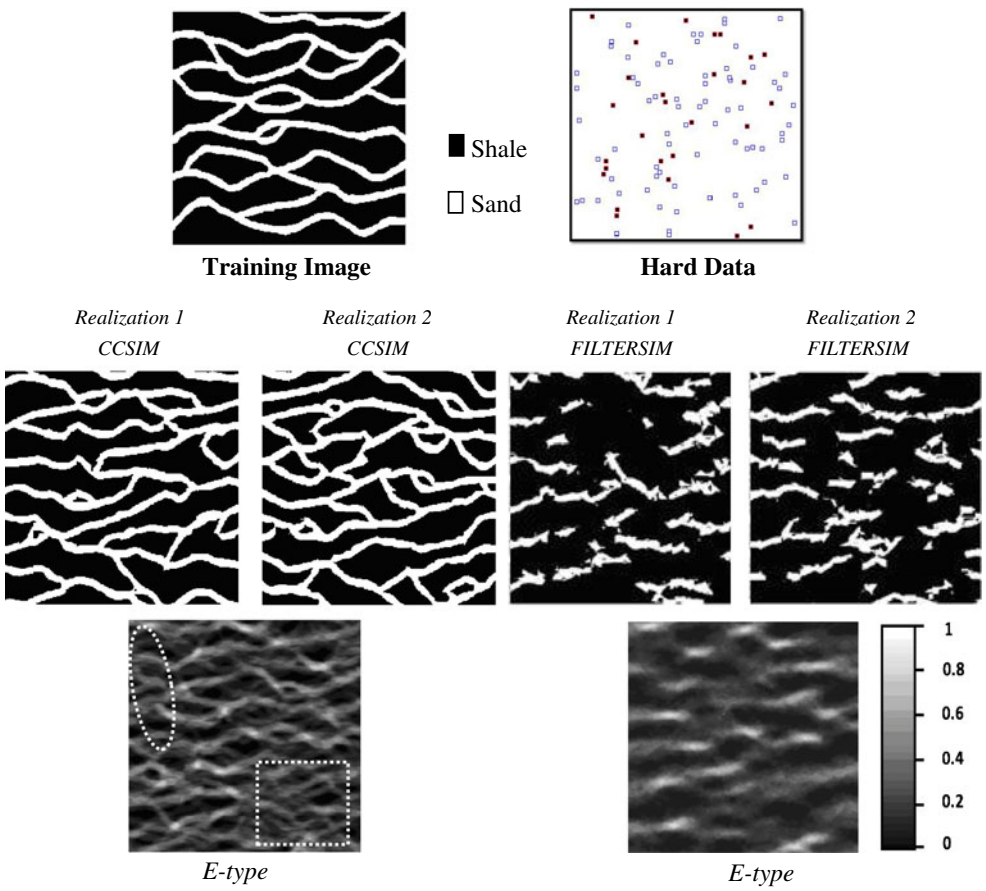

Fig. 13 Same as in Fig. 12, but with 100 hard data points

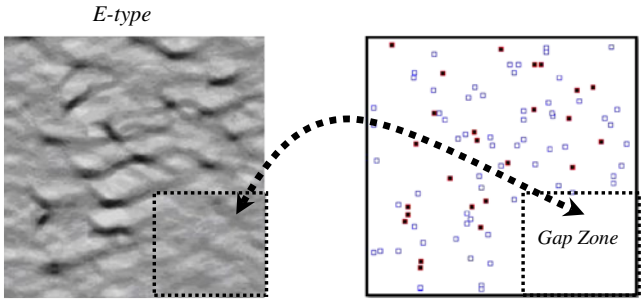

Fig. 14 Sensitivity analysis of CCSIM to a gap zone of the hard data. The TI is the same as in Fig. 13

Training images conditioned with dense hard data We used 100 data points as the HD for conditioning, which were taken from Honarkhah and Caers [35]. The size of the system was 250×250. The results are shown in Fig. 13. It is once again clear that FILTERSIM cannot reproduce the continuity of the features and the geological structure in both the realizations and the E-type maps. It is, however, possible to improve the results of FILTERSIM by increasing the number of the clusters, but it will lead to the algorithm being closer to the SIMPAT and its CPU requirement will increase dramatically.

As one may expect, some uncertainty exists at the bottom right and upper left sections of the realizations, due to the locations of the HD and lack of HD within these zones. For a more accurate sensitivity analysis, we create a gap zone at the bottom right section without any HD, in order to see how the CCSIM may change the E-type map, particularly near and within the gap zone. The results are presented in Fig. 14. It is clear that CCSIM can reproduce an unstructured segment in the gap zone. Therefore, CCSIM generates both specific structure and uncertainty in the presence and absence of the HD.

Another test is checking the effective permeabilities (K_x, K_y), as they depend on the connectivity of the image. To compute the effective permeabilities, we assumed that the sand channels have a permeability of 100 md, while the shale background has a permeability of 0.1 md. The effective permeabilities for the generated realizations were then calculated and averaged over all of them. The results are presented in Table 1. The effective permeabilities of the TI are $K_x = 28$ md and $K_y = 0.17$ md. The computed median values of K_x and K_y for the E-type map generated by CCSIM in Fig. 13 are 27.64 and 0.19 md, respectively, very close to the actual values.

Table 1 The compute effective permeabilities for the E-type map of Fig. 13, generated by the CCSIM

	K_e (x)	K_e (y)
Maximum	32.26	0.73
Upper quartile	29.36	0.29
Median	27.64	0.19
Lower quartile	14.18	0.17
Minimum	1.52	0.15
Mean	27.85	0.32
Variance	3.23	0.07

The results represent averages over many realizations

Geobody conditioning Geobody detection is an important issue in geophysics and underground feature detection. It uses, for example, geophysical attributes to visualize the channels, mounds, clinoforms, etc.; to highlight the differences; and to improve reservoir modeling. An example is shown in Fig. 15, which used the same TI as in Fig. 13. Figure 15 indicates that geobody conditioning is, in fact, a case of conditioning with the HD, which means that this type of conditioning should have a significant effect on the E-type map, in the sense that it reduces the uncertainty. This can be seen in Fig. 15. The CCSIM honors the geobody conditioning perfectly and can also produce a channel by the given geobody. Furthermore, near the geobody the uncertainty is minimum.

3D systems with conditioning To further demonstrate the power of our method, we now describe its application to a 3D system. We first generate a reference image and a TI, shown in Fig. 16. Then, a set of data is sampled from the reference image, also shown in Fig. 16, and is considered as the HD. The result computed with CCSIM is also shown there, which is in reasonable agreement with the reference image, even though the selected TI is not really close to the reference image. For comparison, we also show the result obtained with FILTERSIM.

Fractured systems As pointed out earlier, when a reservoir contains fractures, multiple facies, and other complex features, use of the Euclidean distance in the computations will lead results that are not accurate, whereas if the computations are based on the CC function, one obtains accurate results. To demonstrate this,

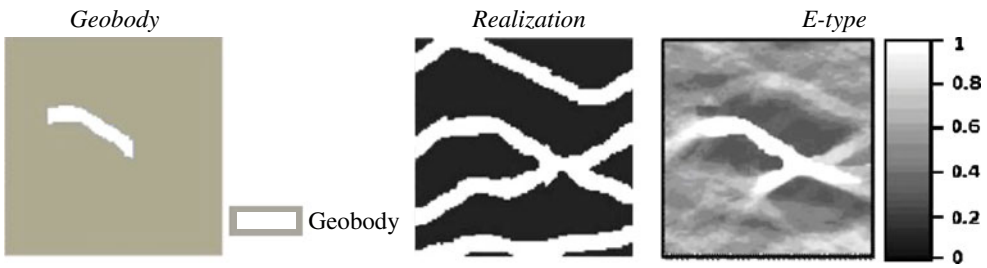

Fig. 15 Results of simulation of conditional CCSIM with a geobody. *Middle panel* shows a single realization. The TI is same as in Fig. 13

we used a TI from Arpart [2], shown in Fig. 17, and tried to reconstruct the image using both the Euclidean distance using the SIMPAT and the CC functions. The results, shown in Fig. 17, demonstrate the superior performance of the CC function.

5 Summary and conclusions

Due to its intrinsic abilities, multiple-point geostatistics is becoming a popular tool in mining, petroleum reservoir modeling, and hydrogeology. One main issue is the

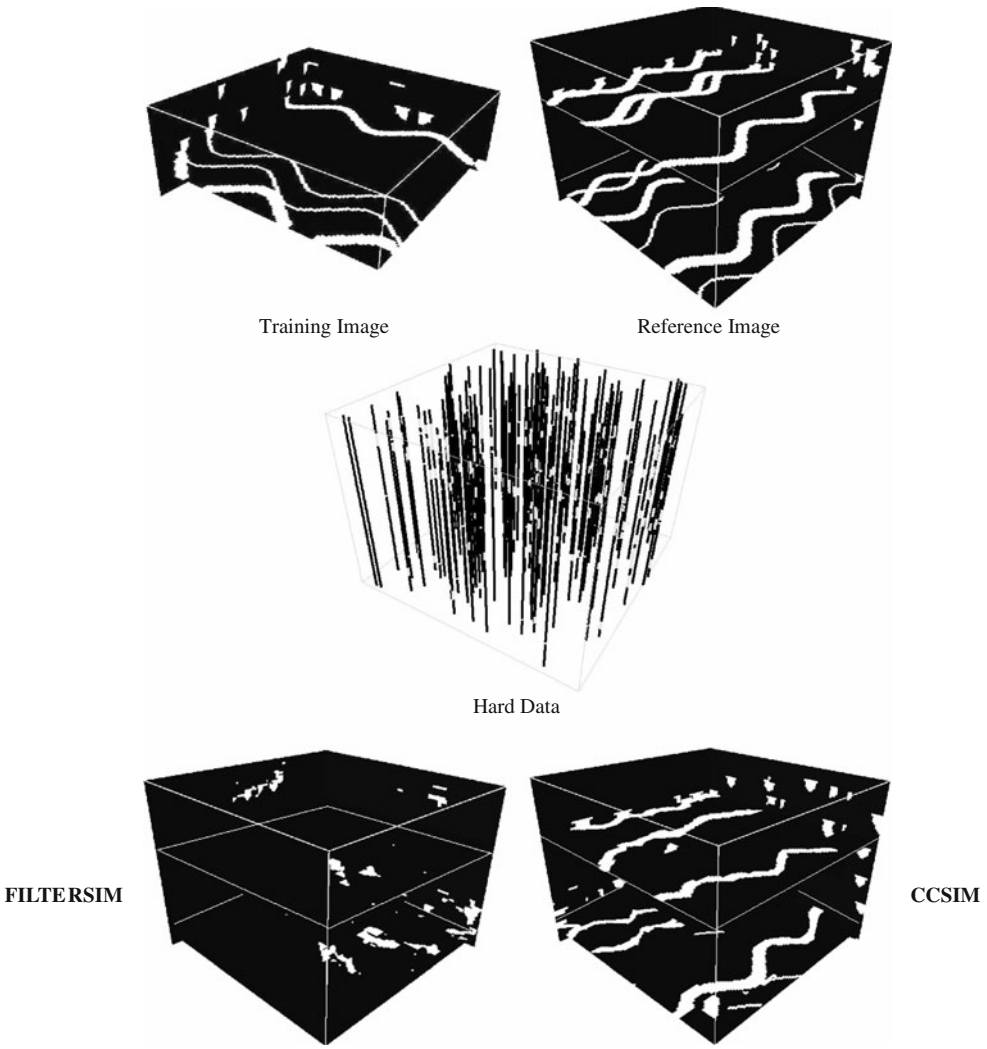

Fig. 16 Result of a simulation of a conditioned 3D system using CCSIM and its comparison with what is obtained with the FILTERSIM method

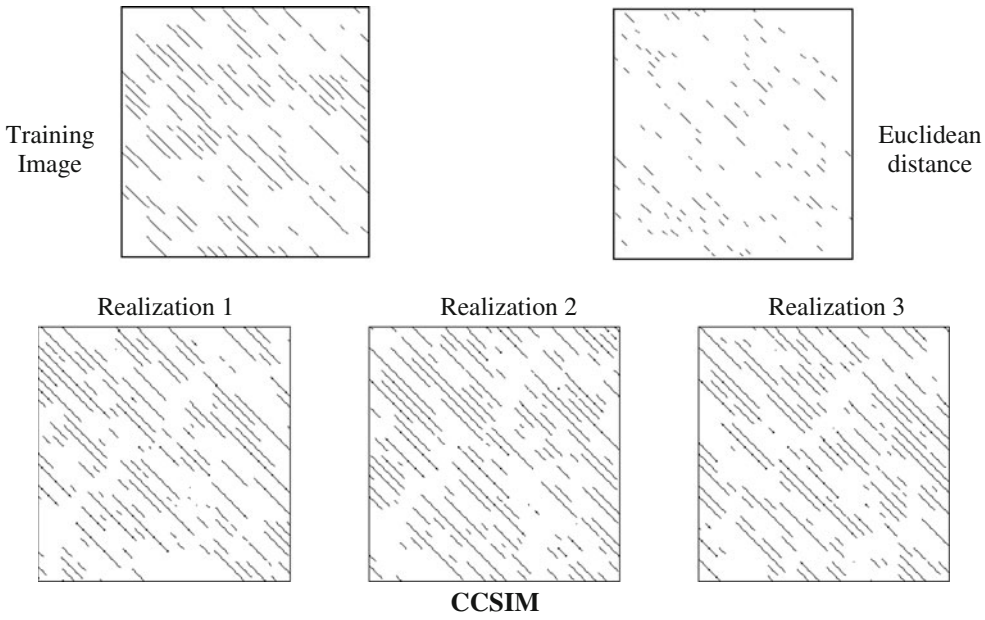

Fig. 17 Comparison of the accuracy of the reconstructed TI, obtained with Euclidean distance function and SIMPAT [2], and three realizations that were computed based on the CC function

development of the MPS methods that are computationally fast but also accurate. In this paper, we presented a CCSIM algorithm that is highly efficient and generates high-quality realizations. The performance of the new method, which can deal with both categorical and continuous training images, was compared with the FILTERSIM algorithm and was shown to be superior. Furthermore, a new conditional scheme based on recursive template splitting that improves the accuracy and continuity of the features near the conditioning data was presented.

The results indicate that CCSIM has significant advantages over the previous MPS methods in terms of the CPU, which is reduced by a factor of 30–80, and the memory requirement of the computations, which is reduced very significantly. One reason for the improved performance is that CCSIM proceeds only point by point, rather than whole patterns, because the final pattern will be selected in the final step of the simulation. Moreover, the algorithm does not need to use a large data event to identify the most similar pattern, as it uses an overlap region to determine the patterns. The overlaps are usually smaller than the data event, and therefore, the algorithm is less CPU demanding. Because CCSIM's grid is not completed before the end of the simulation, it represents an algorithm that in some sense is more stochastic. In other words, in most of the previous pattern-based algorithms that use random paths, the grid **G** is completed gradually, and over time, some of the patterns are deleted from the competition cycle. Thus, the algorithm enables the data events to be more "informed." However, in CCSIM, the comparison or similarity criteria with the training image is based only on the overlap region. Therefore, the patterns always have a chance to be a part of the realization. This feature leads the algorithm to be less sensitive to the initial state of the system and the preceding patterns and enables it to deal with most of the training images. This ability was amply demonstrated by comparing the performance of the method for reproducing complex training images with those produced by the FILTERSIM.

Another feature that helps distinguishing CCSIM from the previous algorithms is the acceptance threshold δ, which is a relaxation threshold that leads the algorithm to generate a pool of patterns, most of which can preserve the similarity and continuity of the image in the overlap regions. In order to have better conditioning of training images, we presented a new approach based on adaptive recursive template splitting. The approach helps better conditioning and continuity because the adaptive algorithm can better honor the hard data.

The overall computational time of CCSIM depends on the template size, TI and the overlap region, and the threshold δ. Since the TI directly affects the CPU time, the computation time can be decreased by using a larger template or smaller overlap region. Thus, for large TI, one can use a larger template to decrease the CPU time. At the same time, an important feature

of CCSIM is that it is amenable to parallel processing. A raster-based algorithm can be parallelized more efficiently than those that are based on random paths. Therefore, the efficiency of the CCSIM can still be improved by using parallel machines, which is very useful when one deals with large 3D systems.

Finally, one may attempt to combine the present methodology, which is a pattern-based method, with new data integration tools, such as the Kappa and Tau models, in order to further improve the accuracy and computational efficiency of CCSIM. These issues will be taken up in our future papers.

Acknowledgements Work at USC was supported in part by the Department of Energy. We thank Andre Journel for many stimulating discussions and an anonymous referee whose constructive comments improved the manuscript.

References

1. Aitokhuehi, I., Durlofsky, L.J.: Optimizing the performance of smart wells in complex reservoirs using continuously updated geological models. J. Pet. Sci. Eng. **48**, 254–264 (2005)
2. Arpat, B.: Sequential simulation with patterns. Ph.D. thesis, Stanford University (2005)
3. Arpat, B., Caers, J.: Stochastic simulation with patterns. Math. Geol. **39**, 177–203 (2007)
4. Borgman, L., Taheri, M., Hagan, R.: Three-dimensional frequency-domain simulations of geological variables. In: Verly, G., et al. (eds.) Geostatistics for Natural Resources Characterization, pp. 517–541. Reidel, Dordrecht (1984)
5. Brunelli, R., Poggio, T.: Face recognition: features versus templates. IEEE Trans. Pattern Anal. Mach. Intell. **15**, 1042–1052 (1993)
6. Caers, J., Park, K.A.: Distance-based representation of reservoir uncertainty: the Metric EnKF. In: Proceedings of the 11th European Conference on the Mathematics of Oil Recovery, Bergen, Norway (2008)
7. Caers, J.: Petroleum Geostatistics. Society of Petroleum Engineers, Richardson (2005)
8. Caers, J., Journel, A.G.: Stochastic reservoir simulation using neural networks trained on outcrop data. SPE paper 49026 (1998)
9. Caers, J., Srinivasan, S., Journel, A.G.: Geostatistical quantification of geological information for a fluvial-type North Sea reservoir. SPE paper 56655 (1999)
10. Caers, J., Strebelle, S., Payrazyan, K.: Stochastic Integration of Seismic and Geological Scenarios: A Submarine Channel Saga, pp. 192–196. The Leading Edge (2003)
11. Chiles, J.P., Delfiner, P.: Geostatistics: Modeling Spatial Uncertainty. Wiley, New York (1999)
12. Chu, J., Journel, A.G.: Conditional fBm simulation with dual kriging. In: Dimitrakopoulos, R. (ed.) Geostatistics for the Next Century, pp. 407–421. Kluwer, Dordrecht (1993)
13. Chugunova, T., Hu, L.: Multiple-point simulations constrained by continuous auxiliary data. Math. Geosci. **40**, 133–146 (2008)
14. Daly, C.: Higher order models using entropy, Markov random fields and sequential simulation. In: Leuangthong, O., Deutsch, C.V. (eds.) Geostatistics, pp. 215–224. Springer, Berlin (2004)
15. Daly, C., Knudby, C.: Multipoint statistics in reservoir modelling and in computer vision. In: Petroleum Geostatistics 2007, Cascais, Portugal (2007)
16. Davis, M.: Production of conditional simulations via the LU decomposition of the covariance matrix. Math. Geol. **19**, 91–98 (1987)
17. Deutsch, C.V.: Annealing techniques applied to reservoir modeling and integration of geological and engineering (well test) data. Ph.D. thesis, Stanford University (1992)
18. Deutsch, C.V., Journel, A.G.: GSLIB: Geostatistical Software Library and User's Guide, 2nd edn. Oxford University Press, Oxford (1998)
19. Deutsch, C.V., Wang, L.: Hierarchical object-based geostatistical modeling of fluvial reservoirs. Paper SPE 36514 (1996)
20. Dimitrakopoulos, R., Mustapha, H., Gloaguen, E.: High-order statistics of spatial random fields: exploring spatial cumulants for modelling complex, non-Gaussian and non-linear phenomena. Math. Geosci. **42**, 65–100 (2010)
21. Dimitrakopoulos, R., Luo, X.: Generalized sequential Gaussian simulation on group size and screen-effect approximations for large field simulations. Math. Geol. **36**, 567–591 (2004)
22. Di Stefano, L., Mattoccia, S., Tombari, F.: ZNCC-based template matching using bounded partial correlation. Pattern Recogn. Lett. **26**, 2129–2134 (2005)
23. El Ouassini, A., Saucier, A., Marcotte, D., Favis, B.: A patchwork approach to stochastic simulation: a route towards the analysis of morphology in multiphase systems. Chaos, Solitons Fractals **36**, 418–436 (2008)
24. Farmer, C.: Numerical rocks. In: King, P.R. (ed.) The Mathematical Generation of Reservoir Geology, pp. 22–33. Clarendon, Oxford (1990)
25. Feyen, L., Caers, J.: Quantifying geological uncertainty for flow and transport modelling in multi-modal heterogeneous formations. Adv. Water Resour. **29**, 912–929 (2006)
26. Gloaguen, E., Dimitrakopoulos, R.: Two-dimensional conditional simulations based on the wavelet decomposition of training images. Math. Geosci. **41**, 679–701 (2009)
27. Goovaerts, P.: Geostatistics for Natural Resources Evaluation. Oxford University Press, Oxford (1997)
28. Goshtasby, A., Gage, S.H., Bartholic, J.F.: A two-stage cross-correlation approach to template matching. IEEE Trans. Pattern Anal. Mach. Intell. **6**, 374–378 (1984)
29. Guardiano, F., Srivastava, M.: Multivariate geostatistics: beyond bivariate moments. In: Geostatistics - Troia, pp. 133–144. Kluwer Academic, Dordrecht (1993)
30. Haldorsen, H.H., Lake, L.W.: A new approach to shale management in field-scale models. Soc. Pet. Eng. J. **24**, 447–452 (1984)
31. Hamzehpour, H., Rasaei, M.R., Sahimi, M.: Development of optimal models of porous media by combining static and dynamic data: the permeability and porosity distributions. Phys. Rev. E **75**, 056311/1–056311/17 (2007)
32. Harding, A., Strebelle, S., Levy, M.: Reservoir facies modeling: new advances in mps. In: Proceedings of GEOSTAT 2004, Banff, Canada (2004)
33. Hoffman, B.T., Caers, J.: History matching by jointly perturbing local facies proportions and their spatial distribution: application to a North Sea reservoir. J. Pet. Sci. Eng. **57**, 257–272 (2007)

34. Holden, L., Hauge, R., Skare, O., Skorstad, A.: Modeling of fluvial reservoirs with object models. Math. Geol. **30**, 473–496 (1998)
35. Honarkhah, M., Caers, J.: Stochastic simulation of patterns using distance-based pattern modeling. Math. Geosci. **42**, 487–517 (2010)
36. Höppner, F., Klawonn, F.: Compensation of translational displacement in time series clustering using cross correlation. In: Proceedings of the 8th International Symposium on Intelligent Data Analysis, Advances in Intelligent Data Analysis VIII (2009)
37. Huysmans, M., Dassargues, A.: Application of multiple-point geostatistics on modelling groundwater flow and transport in a crossbedded aquifer. Paper presented at GeoENV2008: 7th international conference on geostatistics for environmental applications, Southampton, 8–10 Sept 2008
38. Jensen, J.L., Lake, L.W., Corbett, P.W.M., Goggin, D.J.: Statistics for Petroleum Engineers and Geoscientists, 2nd edn. Prentice Hall, Upper Saddle River (2000)
39. Journel, A.G.: Geostatistics for conditional simulation of ore bodies. Econ. Geol. **69**, 673–687 (1974)
40. Journel, A.G.: Geostatistics: roadblocks and challenges. In: Soares, A. (ed.) Geostatistics—Troia, vol. 1, pp. 213-224. Kluwer Academic, Dordrecht (1992)
41. Journel, A.G.: Combining knowledge from diverse sources: an alternative to traditional data independence hypotheses. Math. Geol. **34**, 573 (2002)
42. Journel, A.G., Alabert, F.G.: Non-Gaussian data expansion in the earth sciences. Terra Nova **1**, 123–134 (1989)
43. Journel, A.G., Deutsch, C.V.: Entropy and spatial disorder. Math. Geol. **25**, 329–355 (1993)
44. Journel, A.G., Huijbregts, C.J.: Mining Geostatistics. Academic, New York (1978)
45. Kjønsberg, H., Kolbjørnsen, O.: Markov mesh simulations with data conditioning through indicator kriging. In: Proceedings of the 8th International Geostatistics Congress, Santiago, Chile (2008)
46. Kleingeld, W.J., Lantuejoul, C., Prins, C.F., Thurston, M.L.: The conditional simulation of a Cox process with application to deposits with discrete particles. In: Baafi, E.Y., Schofield, N.A. (eds.) Geostatistics Wollongong 96, pp. 683–694. Kluwer Academic, Dordrecht (1997)
47. Knuth, D.: Art of Computer Programming, 3rd edn. Addison-Wesley, Reading (1997)
48. Lewis, J.P.: Fast template matching. Vision Interface, pp. 120–123 (1995, Fast normalized cross-correlation, unpublished extended version of the same paper)
49. Liu, Y., Harding, A., Abriel, W., Strebelle, S.: Multiple-point simulation integrating wells, three-dimensional seismic data, and geology. AAPG Bull. **88**, 905–921 (2004)
50. Lyster, S., Deutsch, C.V.: MPS simulation in a Gibbs sampler algorithm. In: Proceedings of the 8th International Geostatistics Congress, Santiago, Chile (2008)
51. Mariethoz, G., Renard, P., Straubhaar, J.: The direct sampling method to perform multiple-point simulations. Water Resour. Res. **46**, 2008WR007621 (2010)
52. Matheron, G.: The intrinsic random functions and their applications. Adv. Appl. Probab. **5**, 439–68 (1973)
53. Mattoccia, S., Tombari, F., Di Stefano, L.: Reliable rejection of mismatching candidates for efficient ZNCC template matching. In: Image Processing. ICIP 2008. 15th IEEE International, pp. 849–852 (2008)
54. Ooi, J., Rao, K.: New insights into correlation-based template matching. In: Proceedings of SPIE, V., Applications of Artificial Intelligence IX, p. 1468. Orlando (1991)
55. Ortiz, J.M., Deutsch, C.V.: Indicator simulation accounting for multiple-point statistics. Math. Geol. **36**, 545–565 (2004)
56. Ortiz, J.M., Emery, X.: Integrating multiple point statistics into sequential simulation algorithms. In: Leuangthong, O., Deutsch, C.V. (eds.) Geostatistics Banff 2004, pp. 969–978. Springer, Berlin (2005)
57. Park, K.A., Schiedt, C., Caers, J.: Simultaneous conditioning of multiple non-Gaussian geostatistical models to highly nonlinear data using distances in kernel space. In: Proceedings of the 8th International Geostatistical Congress, Santiago, Chile (2008)
58. Parra, A., Ortiz, J.M.: Conditional multiple-point simulation with a texture synthesis algorithm. In: IAMG 2009 Conference, Stanford University (2009)
59. Remy, N., Boucher, A., Wu, J.: Applied Geostatistics with SGeMS: A User's Guide. Cambridge University Press, Cambridge (2008)
60. Renard, P.: Stochastic hydrogeology: what professionals really need? Ground Water **45**, 531–541 (2007)
61. Ronayne, M., Gorelick, S., Caers, J.: Identifying discrete geologic structures that produce anomalous hydraulic response: an inverse modeling approach. Water Resour. Res. **44**, W08426 (2008)
62. Sahimi, M.: Large-scale porous media and wavelet transformations. Comput. Sci. Eng. **5** (4), 75–87 (2003)
63. Sahimi, M.: Flow and Transport in Porous Media and Fractured Rock, 2nd edn. Wiley-VCH, Weinheim (2011)
64. Skorstad, A., Hauge, R., Holden, L.: Well conditioning in a fluvial reservoir model. Math. Geol. **31**, 857–872 (1999)
65. Srivastava, R.: An overview of stochastic methods for reservoir characterization. In: Chambers, R.L. (ed.) Stochastic Modeling and Geostatistics: Principles, Methods, and Case Studies, Computer applications, vol. 3, p. 380. AAPG, Tulsa (1994)
66. Straubhaar, J., Renard, P., Mariethoz, G., Froidevaux, R., Besson, O.: An improved parallel multiple-point algorithm using a list approach. Math. Geosci. **43**, 305–328 (2011)
67. Straubhaar, J., Walgenwitz, A., Renard, P., Froidevaux, R.: Optimization issues in 3D multipoint statistics simulation. Paper presented at Geostats 2008, Santiago, Chile (2008)
68. Strebelle, S.: Conditional simulation of complex geological structures using multiple-point geostatistics. Math. Geol. **34**, 1–22 (2002)
69. Strebelle, S., Payrazyan, K., Caers, J.: Modeling of a deepwater turbidite reservoir conditional to seismic data using multiple-point geostatistics. SPE paper 77425 (2002)
70. Suzuki, S., Caers, J.: A distance-based prior model parameterization for constraining solutions of spatial inverse problems. Math. Geosci. **40**, 445–469 (2008)
71. Suzuki, S., Caumon, G., Caers, J.: Dynamic data integration for structural modeling: model screening approach using a distance-based model parameterization. Comput. Geosci. **12**, 105–119 (2008)
72. Suzuki, S., Strebelle, S.: Real-time post-processing method to enhance multiple-point statistics simulation. Paper presented at Petroleum Geostatistics 2007, Cascais, Portugal (2007)
73. Theodoridis, S., Koutroumbas, K.: Pattern Recognition & Matlab Intro: Pattern Recognition. Academic, New York (2007)
74. Tjelmeland, H.: Stochastic models in reservoir characterization and Markov random fields for compact objects. Doctoral dissertation, Norwegian University of Science and Technology, Trondheim (1996)
75. Tjelmeland, H., Eidsvik, J.: Directional Metropolis–Hastings updates for posteriors with non linear likelihood. In: Leuangthong, O., Deutsch, C.V. (eds.) Geostatistics, pp. 95–104. Springer, Banff (2004)

76. Vargas-Guzmán, J.A.: Unbiased resource evaluations with kriging and stochastic models of heterogeneous rock properties. Nat. Resour. Res. **17**, 245–254 (2008)
77. Vargas-Guzmán, J.A.: The Kappa model of probability and higher-order rock sequences. Comput. Geosci **15**, 661–671 (2011)
78. Wu, J., Zhang, T., Journel, A.G.: A fast FILTERSIM simulation with score-based distance. Math. Geosci. **47**, 773–788 (2008)
79. Zhang, T., Switzer, P., Journel, A.G.: Filter-based classification of training image patterns for spatial simulation. Math. Geol. **38**, 63–80 (2006)