Les Cahiers du GERAD ISSN: 0711–2440

A new high-order, non-stationary and transformation invariant spatial simulation approach

A. A. Haji Abolhassani, R. Dimitrakopoulos, F.P. Ferrie

G–2016–52

July 2016

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

A new high-order, non-stationary and transformation invariant spatial simulation approach

Amir Abbas Haji Abolhassani [a,b]

Roussos Dimitrakopoulos [a]

Frank P. Ferrie [b]

[a] *GERAD & COSMO - Stochastic Mine Planning, Department of Mining and Materials Engineering, McGill University, Montreal (Quebec) Canada*

[b] *Electrical amd Computer Engineering Deptartment, McGill University, Montreal (Quebec) Canada*

amir@cim.mcgill.ca
roussos.dimitrakopoulos@mcgill.ca
ferrie@cim.mcgill.ca

July 2016

Les Cahiers du GERAD

G–2016–52

Abstract: This paper presents a new high-order, non-stationary sequential simulation approach, aiming to deal with the typically complex, curvilinear structures and high-order spatial connectivity of the attributes of natural phenomena. Similarly to multipoint methods, the proposed approach employs spatial templates and a group of training images (TI). A coarse template with a fixed number of data points and a missing value in the middle is used, where the missing value is simulated conditional to a data event found in the neighborhood of the middle point of the template, under a Markovian assumption. Sliding the template over the TI, a pattern database is extracted. The parameters of the conditional distributions needed for the sequential simulation are inferred from the pattern database considering a set of weights of contribution given for the patterns in the database. Weights are calculated based on the similarity of the high-order statistics of the data event of the hard data compared to those of the training image. The high-order similarity measure introduced herein is effectively invariant under all linear spatial transformations. Following the sequential simulation paradigm, the template chosen is sequentially moved on a raster path until all missing points/nodes are simulated. The high-order similarity measure allows the approach to be fast as well as robust to all possible linear transformations of a training image. The approach respects the hard data and its spatial statistics, because it only considers TI replicate data events with similar high-order statistics. Results are promising.

Keywords: High-order spatial statistics, sequential simulation, non-stationary, transformation invariant, multi-point statistics

Acknowledgments: Funding was provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant 239019 and mining industry partners of the COSMO laboratory (AngloGold Ashanti, Barrick Gold, BHP Billiton, De Beers Canada, Kinross Gold, Newmont Mining and Vale), and the Group for Research in Decision Analysis (GERAD). Thanks are in order to Prasun Lala for his assistance.

1 Introduction

Since the early 1990's (Guardiano and Srivastava, 1993) several new approaches to geostatistical simulation have been developed to move this area of research beyond the second-order or two-point methods and their limits. These methods, well developed to date, are placed under the term multi-point statistics (MPS). The basic idea of MPS approaches is that the two-point statistical tools (variogram, covariance, correlogram) of a given attribute of interest are replaced by a so-called training image (TI). The TI is then used as a source to provide multiple point statistics and spatial relations that are used along with the hard data to generate simulated realizations for the attributes of interest. The first implemented multi-point method is SNESIM (Strebelle, 2002) and it is TI driven; thus, similar to all conventional MPS methods, the simulated realizations reproduce the high-order spatial relations in the TI. As a result, in applications with a dense set of hard data, the complex spatial relations in the data are overridden by those in the TI, and does not assist with the application of MPS methods to applications with relatively dense datasets.

Several MPS methods are well known to date, examples are discussed next. FILTERSIM (Zhang et al., 2006) is based on the classification of both data and TIs using linear filters it is efficient and also sensitive to the shape and size of the spatial template, and the number and form of the filters employed. The direct sampling method (Mariethoz et al., 2010), in the other hand, does not produce a pattern database from the TIs like FIMTERSIM. Instead, in a multigrid simulation setup, first coarser grid nodes are simulated, a template is chosen about a simulation point and the data-event is extracted. This data-event is then compared with the data-event of a randomly chosen TI pattern, in an L2-norm distance basis. If the distance is less than a threshold, then the pattern is pasted onto the simulation grid, otherwise, another TI pattern is randomly selected and compared, and so on. The direct sampling simulation method is fast and effective in simulating based on sparse data with a given TI set, and is also TI driven. Mariethoz and Kelly (2011) show the influence of data statistics on realizations generated from this method as a dataset increases. Other pattern-based MPS methods include the one suggested by Arpat and Caers (2007); first, a pattern database is generated by sliding a fixed template over the TIs. Then, the data event on the grid used for the simulation and at each location is compared with the data event of the pattern database and the one with the least L2-norm distance is chosen and pasted on the grid nodes involves. Abdollahifard and Faez (2012) first cluster the pattern database generated from the TI using a Bayesian framework. Then each cluster is modeled by a set of simple linear features and the extraction of features for each incomplete pattern on the simulated grid follows. Honarkhah Caers (2010) instead of building a raw pattern database, classifies TIs using some fixed simple features and compares them, using the L2-norm, to the same features extracted from a point of the grid being simulated. The most similar pattern is pasted onto the grid nodes, until all of them are visited.

MPS simulations based on Markov methods are also available. An example in this category is the Markov Mesh model by Stien and Kolbjørnsen (2011). A unilateral raster path is chosen for the data, visiting all of the points in a left-to-right and up-to-down fashion. On a chosen spatial template a joint distribution is considered for the random field. The parameters of this distribution are then estimated from the TIs. Sampling of the local distribution function generates the realizations. These models in general are biased towards the chosen path, but are vastly used for simulating attributes of petroleum reservoirs.

In a relatively recent approach to stochastic simulation, the high-order sequential simulation extends the conventional second-order sequential simulation methods to higher-orders (Mustapha and Dimitrakopoulos, 2010a, 2011). The HOSIM approach first chooses a simulation point at random and considers N nearest conditioning data as data-event. A special template is built by connecting the data-event to the simulation point. The conditional probability distribution function (CPDF) of the simulated grid node given the data-event is then modeled by a series of weighted orthogonal functions called Legendre polynomials. The weight of each Legendre term is calculated by matching a set of particular spatial statistics, the so-called spatial cumulants that are generated from the available data. The TI is only used to complement the spatial cumulants of the available data. Note that SNESIM is similar to HOSIM with a main difference that the model used for the CPDF is much simpler and the method is TI driven.

The present manuscript presents a new patch-based high-order method, which utilizes high-order spatial statistics in the pattern's structure. Two notable differences from past approaches are that (a) it is non-stationary, and that (b) utilizes a set of TI's, rather than one, while it is data-driven. Both the above address significant topics. The proposed method follows a multigrid simulation process (coarser simulation grid nodes are simulated first and become the conditioning data for finer simulation grid points. A simulation point is randomly selected from the grid to be simulated

and an order $N+1$ template is selected, based on N nearest hard data points, *i.e.* the data-event. In addition, a N-dimensional high-order statistical feature vector, introduced herein, is calculated from the data-event. By sliding the template over the TI a pattern database is produced and then mapped into a N-dimensional high-order statistical feature space. The similarity measure of two feature vectors is defined as the weighted Euclidean distance between two vectors. The distribution of the simulated point is then estimated from the TIs using the maximum likelihood estimate (MLE) considering the similarity of each pattern. A sample is then drawn from the distribution as the realization of the simulation point. This process is continued until all grid nodes are simulated. The feature vector introduced in this paper is isotropic, that is, it is invariant to any linear transformation of the training image including rotation and transposition. This feature is also fast to calculate enabling the simulator to incorporate a large amount of TIs. The simulator is non-stationary and respects the hard data and its high order statistics, that is, only TI patterns with similar statistics are used for simulation. As a result, the simulations are data-driven.

The following sections present the proposed simulation approach; then the results of initial tests using known datasets follow. Conclusions and future work complete the presentation.

2 The proposed method

2.1 Overview

The goal of the proposed method is to simulate a random field $Z(\boldsymbol{x})$, in a sequential multi-grid process, given a grid with nodes $\boldsymbol{x}$, a training image (TI), $z(\boldsymbol{Y}) = \{z(\boldsymbol{y}_1), \dots, z(\boldsymbol{y}_M)\}$, with nodes $\boldsymbol{Y} = \{\boldsymbol{y}_1, \dots, \boldsymbol{y}_M\}$ in the training image, and a sparse set of N hard data $z(x_i)$, $i \in [1, \dots, N]$ on a regular grid $x_i, i \in [1, \dots, N]$. The hierarchy of the sequential simulation is illustrated in Figure 1. The blue nodes represent the hard data (a). In the first sequence, each red node is conditioned on the four closest blue nodes and simulated. The lines represent the conditions in (b). This continues sequentially until all the nodes the grid are simulated, Figure 1 (c, d, e). Unicolor lines in each figure represent the spatial templates, connecting the conditioning data and a single node to be simulated. The size of the templates reduces after each sequence to maintain the same number of conditioning nodes. This is the natural representation of the multi-grid approach for the simulation.

2.2 High-order transformation invariant simulation (HOSTSIM)

At each sequence, the path is chosen randomly and saved into a vector containing the indices of the visiting nodes. Each successive random variable $Z(\boldsymbol{x})$ at node $\boldsymbol{x}$, is conditioned to n-nearest neighbours, selected from the set of previously simulated nodes and the hard data $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$ (Goovaerts, 1998). A template is formed spatially by connecting each conditioning data, $\boldsymbol{x}_i$, to node $\boldsymbol{x}$, presented by a lag vector $L_{\boldsymbol{x}} = \{\boldsymbol{h}_1, \dots, \boldsymbol{h}_n\} = \{\boldsymbol{x}_1 - \boldsymbol{x}, \dots, \boldsymbol{x}_n - \boldsymbol{x}\}$. Consequently, the neighbourhood of $\boldsymbol{x}$ is denoted by $\boldsymbol{N}_{\boldsymbol{x}} = \{\boldsymbol{x} + \boldsymbol{h}_1, \dots, \boldsymbol{x} + \boldsymbol{h}_n\}$ and the data event is denoted by $\boldsymbol{d}_{\boldsymbol{N}_{\boldsymbol{x}}} = \{z(\boldsymbol{x}_1), \dots, z(\boldsymbol{x}_n)\}$. The goal is to estimate and draw a sample from the probability, $P\left(Z(\boldsymbol{x}) \middle| \boldsymbol{d}_{\boldsymbol{N}_{\boldsymbol{x}}}, z(\boldsymbol{Y})\right)$, of each successive $Z(\boldsymbol{x})$ in the next visiting node on the path given its data event, $\boldsymbol{d}_{\boldsymbol{N}_{\boldsymbol{x}}}$, and the TI, $z(\boldsymbol{Y})$. This probability is intractable for continuous variables. Alternatively, a model with a set of parameters, $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ may be chosen, to represent this probability independent from the data event and TI. The parameter θ is optimized to express the data event and TI. Thus, this probability can be decomposed using the Bayes product and sum rule:

$$P\left(Z(\boldsymbol{x}) \middle| \boldsymbol{d}_{\boldsymbol{N}_{\boldsymbol{x}}}, z(\boldsymbol{Y})\right) = \int_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} P(Z(\boldsymbol{x}) | \boldsymbol{\theta}) P\left(\boldsymbol{\theta} \middle| \boldsymbol{d}_{\boldsymbol{N}_{\boldsymbol{x}}}, z(\boldsymbol{Y})\right) d\boldsymbol{\theta}. \quad (1)$$

Estimation of the integrand further simplifies Equation 1. Figure 2 represents the term $P\left(\boldsymbol{\theta} \middle| \boldsymbol{d}_{\boldsymbol{N}_{\boldsymbol{x}}}, z(\boldsymbol{Y})\right)$ as a function of $\boldsymbol{\theta}$.

The contribution of this function is negligible except for a narrow band near an optimal value for the parameter's Maximum Likelihood Estimate, $\boldsymbol{\theta}_{MLE}$

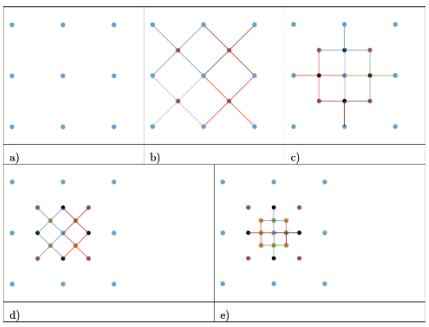

Figure 1. The hierarchy of the sequential-multi-grid simulation for $n = 5$, the connecting lines present the conditioning dependencies. a) The hard data. b) Red nodes are simulated conditioned on their 4 nearest neighbours. c) Black nodes are simulated conditioned on both blue and red nodes from previous sequence. d) Oranges are simulated conditioned on previous simulated nodes and hard data. e) Greens are simulated in the last sequence. At each sequence the resolution of the grid doubles.

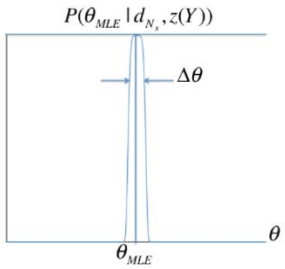

Figure 2. Presentation of $P\left(\boldsymbol{\theta}\middle|\boldsymbol{d}_{\boldsymbol{N_x}}, z(Y)\right)$ as a function of $\boldsymbol{\theta}$. In practice this probability is negligible except at a narrow $\Delta\boldsymbol{\theta}$ band near $\boldsymbol{\theta}_{MLE}$.

Consequently, Equation 1 could be estimated as in Equation 2:

$$P\left(Z(\boldsymbol{x})\middle|\boldsymbol{d}_{\boldsymbol{N_x}}, z(\boldsymbol{Y})\right) \approx P(Z(\boldsymbol{x})|\boldsymbol{\theta}_{MLE}) \overbrace{P\left(\boldsymbol{\theta}_{MLE}\middle|\boldsymbol{d}_{\boldsymbol{N_x}}, z(\boldsymbol{Y})\right)\Delta\boldsymbol{\theta}}^{constant}$$

$$= \frac{1}{C_{MLE}} P(Z(\boldsymbol{x})|\boldsymbol{\theta}_{MLE}). \tag{2}$$

In Equation 2, C_{MLE} can be regarded as the normalization factor to produce a valid probability, $C_{MLE} = \int_{z\in Z} P(z|\boldsymbol{\theta}_{MLE})dz$.

Equation 2 implies the probability distribution function of the node $\boldsymbol{x}$ can be calculated if $\boldsymbol{\theta}_{MLE}$ is known. To estimate $\boldsymbol{\theta}_{MLE}$ based on Figure 2, $\boldsymbol{\theta} = \boldsymbol{\theta}_{MLE}$ when $P(\boldsymbol{\theta}|\boldsymbol{d}_{\boldsymbol{N_x}}, z(\boldsymbol{Y}))$ is maximum for $\boldsymbol{\theta} \in \boldsymbol{\Theta}$.

$$\boldsymbol{\theta}_{MLE} = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{argmax}\, P\left(\boldsymbol{\theta}\middle|\boldsymbol{d}_{\boldsymbol{N_x}}, z(\boldsymbol{Y})\right). \tag{3}$$

Using Bayes theorem:

$$\boldsymbol{\theta}_{MLE} = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{argmax}\, \frac{P\left(z(\boldsymbol{Y})\middle|\boldsymbol{\theta}, \boldsymbol{d}_{\boldsymbol{N_x}}\right) \overbrace{\widetilde{P(\boldsymbol{\theta})}}^{uniform\ prior}}{\underbrace{P(z(\boldsymbol{Y}))}_{independant\ from\ \theta}}. \tag{4}$$

One assumes a uniform prior in parameters space $\boldsymbol{\Theta}$, and note that the marginal probability in TI, $P(z(\boldsymbol{Y}))$, is independent from $\boldsymbol{\theta}$. Hence, Equation 4 becomes:

$$\boldsymbol{\theta}_{MLE} = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{argmax}\, P\left(z(\boldsymbol{Y})\middle|\boldsymbol{\theta}, \boldsymbol{d}_{\boldsymbol{N_x}}\right). \tag{5}$$

Each node in TI is only conditioned on its neighbours and the parameters set $\boldsymbol{\theta}$. Hence, the joint distribution in Equation 5 can be decomposed further.

$$\boldsymbol{\theta}_{MLE} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{argmax} \prod_{i=1}^{M} P\left(z(\boldsymbol{y}_i)\middle|\boldsymbol{\theta}, \boldsymbol{d}_{\boldsymbol{N}_x}, \boldsymbol{d}_{\boldsymbol{N}_{y_i}}\right). \tag{6}$$

A probability is maximized if the logarithm of that probability is maximized:

$$\boldsymbol{\theta}_{MLE} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{argmax} \sum_{i=1}^{M} logP\left(z(\boldsymbol{y}_i)\middle|\boldsymbol{\theta}, \boldsymbol{d}_{\boldsymbol{N}_x}, \boldsymbol{d}_{\boldsymbol{N}_{y_i}}\right). \tag{7}$$

At maximum the derivative with respect to $\boldsymbol{\theta}$ should be zero:

$$\frac{\partial}{\partial \boldsymbol{\theta}}\left(\sum_{i=1}^{M} logP\left(z(\boldsymbol{y}_i)\middle|\boldsymbol{\theta}, \boldsymbol{d}_{\boldsymbol{N}_x}, \boldsymbol{d}_{\boldsymbol{N}_{y_i}}\right)\right) = 0\,. \tag{8}$$

Equation 8 should be solved for $\boldsymbol{\theta}_{MLE}$. By choosing a model with a set of parameters $\boldsymbol{\theta}$, solving Equation 8 for $\boldsymbol{\theta}_{MLE}$ is straightforward.

2.3 Simulation model

The exponential family is used herein to model the likelihood function in Equation 8 with the parameter set $\boldsymbol{\theta} = \{\theta_1, \theta_2\}$.

$$P\left(z(\boldsymbol{y}_i)\middle|\boldsymbol{\theta}, \boldsymbol{d}_{\boldsymbol{N}_x}, \boldsymbol{d}_{\boldsymbol{N}_{y_i}}\right) = \frac{1}{c}\exp\left(-\frac{1}{2}\omega\left(\boldsymbol{d}_{\boldsymbol{N}_x}, \boldsymbol{d}_{\boldsymbol{N}_{y_i}}, \sigma_0^2\right)\frac{(z(\boldsymbol{y}_i)-\theta_1)^2}{\theta_2}\right). \tag{9}$$

$\omega\left(\boldsymbol{d}_{\boldsymbol{N}_x}, \boldsymbol{d}_{\boldsymbol{N}_{y_i}}, \sigma_0^2\right)$ is introduced as the similarity measure (SM) of the data event $\boldsymbol{d}_{\boldsymbol{N}_x}$ and $\boldsymbol{d}_{\boldsymbol{N}_{y_i}}$. It ensures that the TI patterns with similar data events contribute more toward building the likelihood function in Equation 9. The SM is defined as:

$$\omega\left(\boldsymbol{d}_{\boldsymbol{N}_x}, \boldsymbol{d}_{\boldsymbol{N}_{y_i}}, \sigma_0^2\right) = \exp\left(-\frac{\frac{1}{2}\boldsymbol{D}\left(\boldsymbol{d}_{\boldsymbol{N}_x}, \boldsymbol{d}_{\boldsymbol{N}_{y_i}}\right)^T \boldsymbol{D}\left(\boldsymbol{d}_{\boldsymbol{N}_x}, \boldsymbol{d}_{\boldsymbol{N}_{y_i}}\right)}{\boldsymbol{\Sigma}\left(\boldsymbol{d}_{\boldsymbol{N}_x}, \sigma_0^2\right)}\right). \tag{10}$$

where $\boldsymbol{D}\left(\boldsymbol{d}_{\boldsymbol{N}_x}, \boldsymbol{d}_{\boldsymbol{N}_{y_i}}\right)$ is introduced as the high-order-statistics disparity vector. $\boldsymbol{\Sigma}\left(\boldsymbol{d}_{\boldsymbol{N}_x}, \sigma_0^2\right)$ is the covariance matrix of the disparity vector and is calculated using the calculus of variations.

2.4 High-order-statistics disparity vector

In this paper, a particular form of disparity vector is presented, which is isotropic and compares the high-order statistics of two data events. Most of the MP simulation methods choose an L2-norm for the disparity measure (Arpat and Caers, 2007; Chatterjee et al., 2012, Honarkhah and Caers, 2010; Mariethoz et al., 2010, and Mustapha et al., 2013). When considering two sets of data events $\boldsymbol{d}_{\boldsymbol{N}_x} = \{x_1, \dots, x_n\}$, $\boldsymbol{d}_{\boldsymbol{N}_y} = \{y_1, \dots, y_n\}$ of order n, to develop an isotropic L2-norm disparity measure, one must compare all possible ordering of these two data events, which results in $n \times n!$ number of operations. *F*or $n = 5$ the number of operations are 600. This is computationally expensive and can only operate on small size TIs. The following method is employed to reduce the computing time.

First, Vieta's formula (Funkhouser, 1930) is used to calculate the coefficients of two polynomials $p_s(X)$ and $p_t(Y)$ with the roots equal to the data events $\boldsymbol{d}_{\boldsymbol{N}_x}$ and $\boldsymbol{d}_{\boldsymbol{N}_y}$, respectively.

$$p_{\boldsymbol{s}}(X) = X^n + s_1 X^{n-1} + \cdots + s_n \,.$$

$$\begin{cases} s_1 = x_1 + x_2 + \cdots + x_n \\ s_2 = x_1(x_2 + x_3 + \cdots + x_n) + x_2(x_3 + \cdots + x_n) + x_{n-1}x_n \\ \vdots \\ s_n = x_1 x_2 \ldots x_n \end{cases}. \tag{11}$$

$$p_{\boldsymbol{t}}(Y) = Y^n + t_1 Y^{n-1} + \cdots + s_n \,.$$

$$\begin{cases} t_1 = y_1 + y_2 + \cdots + y_n \\ t_2 = y_1(y_2 + y_3 + \cdots + y_n) + y_2(y_3 + \cdots + y_n) + y_{n-1}y_n \\ \vdots \\ t_n = y_1 y_2 \ldots y_n \end{cases}. \tag{12}$$

These could be regarded as two mappings $\boldsymbol{d}_{\boldsymbol{N_x}} \to \boldsymbol{s} = \{s_1, \ldots, s_n\}$ and $\boldsymbol{d}_{\boldsymbol{N_y}} \to \boldsymbol{t} = \{t_1, \ldots, t_n\}$. The advantage of these mappings is that they are invariant to the ordering of the domain. This invariance results from the coefficients of a polynomial being invariant to the order of the roots in a set due to equations 11 and 12. $\boldsymbol{s}$ and $\boldsymbol{t}$ are in a particular form of high-order moments:

$$s_m = \sum_{k \in K} N(k) m^{(u)}(k),$$

$$m^{(u)}(k) = \frac{1}{N(k)} \sum_{j=1}^{N(k)} x_j \prod_{l \in \boldsymbol{k}} x_{j+l} \,, \tag{13}$$

$$t_m = \sum_{k \in K} N(k) m^{(u)}(k) \,,$$

$$m^{(u)}(k) = \frac{1}{N(k)} \sum_{j=1}^{N(k)} y_j \prod_{l \in \boldsymbol{k}} y_{j+l} \,, \tag{14}$$

where, $\boldsymbol{k} = \{k_1, \ldots, k_{u-1}\}$ and $N(k)$ is the support for estimating the moment. The high-order statistics disparity vector is defined as follows:

$$\boldsymbol{D}\left(\boldsymbol{d}_{\boldsymbol{N_x}}, \boldsymbol{d}_{\boldsymbol{N_{y_i}}}\right) = \boldsymbol{s}^T - \boldsymbol{t}^T \,. \tag{15}$$

It worth mentioning that the number of operations for this new disparity measure is reduced dramatically to $n \times 2^{n-1}$ per simulation node versus $n \times n!$ for the L2-norm. *F*or $n = 5$, $op(L2 - norm) = 600$ and $op(high - ord) = 80$.

3 Results from HOSTSIM and comparisons

The dataset used in this section is the Stanford V Reservoir dataset (Mao and Journel, 1999). This exhaustive dataset consists of a 3D grid with porosity values. The grid consisting of $130 \times 100 \times 30$ nodes, *i.e.* $X \times Y \times Z$. Here this dataset is cropped into a grid of $100 \times 100 \times 30$ in order to perform some linear transformation on the data, *e.g.* rotation. For each simulation, a layer, $Z \in \{1, \ldots, 30\}$, is selected as the ground truth of the simulation, referred to as the original image. This image is then down-sampled to produce the hard data set, containing N points. All layers except layer Z are considered as TI for each simulation. The results of the simulation produced by HOSTISIM are compared, with an order 5 template, with the ones produced by FILTERSIM, with a search grid size 11×11 and inner patch size 7×7. Note that the TI's provided for HOSTISIM are rotated 90 degrees clock-wise. The simulations are generated for layers $z = \{1, \ldots, 4\}$, from top to bottom in Figures 3 and 4. First with $N = 625$ number of hard data points, 6.25% of the original data, in Figure 3 and second with $N = 169$ number of hard data points, 1.69% of the original data, in Figure 4. For each simulation on Figures 3 and 4, from left to right, the original image, HOSTISIM and FILTERSIM typical realizations are all presented. For each set of results, the histogram of the original image and two simulations are also plotted. As a robust quantitative comparison, for each simulation 10 realizations are generated

by HOSTISIM and FILTERSIM methods and for each one the PSNR and SSIM scores (Wang et al., 2004) are calculated and averaged for each method over all 10 realizations and provided on Figures 3 and 4. For every single case, HOSTISIM outperforms FILTERSIM; visually by better representing the channels and low contrast structures of the original exhaustive image, and with higher PSNR and SSIM scores, and by better matching the histogram. Tables 1 and 2 are presenting the average PSNR and SSIM for HOSTISIM and FILTERSIM.

Table 1. Comparing average PSNR and SSIM for HOSTISIM and FILTERSIM methods with $N = 625$ number of hard data points (6.25%).

	PSNR		**SSIM**	
Z	HOSTISIM	FILTERSIM	HOSTISIM	FILTERSIM
1	24.94	23.10	0.61	0.50
2	24.31	22.77	0.60	0.47
3	23.84	22.42	0.57	0.44
4	23.10	21.29	0.56	0.42

Table 2. Comparing average PSNR and SSIM for HOSTISIM and FILTERSIM methods with $N = 169$ number of hard data points (1.69%).

	PSNR		**SSIM**	
Z	HOSTISIM	FILTERSIM	HOSTISIM	FILTERSIM
1	21.23	20.98	0.47	0.43
2	21.49	20.65	0.47	0.40
3	20.87	20.20	0.42	0.35
4	20.17	18.76	0.40	0.30

For more and higher resolution results, please visit this link: http://cim.mcgill.ca/~amir/HOSTISIM.html

4 Comparing the computing times

The current implementation of the method is in Matlab, using the GPU parallel computation library. It has not been optimized nor developed in C, nor Python, as of yet. On the other hand, FILTERSIM has been optimized and developed in Python, and is available in the SGEMS software platform. Despite this disparity in optimization, we ran some tests to compare them as is. The system used for the tests was a Unix OS server with 8 cores Xeon CPU, 3.500GHz with 8MB cache size and 64GB DDR4 memory and Nvidia Tesla k40c GPU with 12GB DDR5 memory with 2880 cores.

For each method, two sets of tests were performed. Each set of tests consisted of generating 10 separate simulations and averaging the computing time for each simulation. The hard data used for all cases were 12x12 real-valued data on a regular grid. The goal was to generate a realization on a 100x100 SG, for each case. Each TI was a 100x100 real-valued image. In the first test only 1 TI was used and in the second test 29 TI's were used. The average computing times are presented in Table 3.

Table 3. Comparing average computation time for HOSTISIM and FILTERSIM methods.

	FILTERSIM	**HOSTISIM**	*******
Test#1	8 seconds	3 seconds	+ 9 seconds GPU initialization in Matlab
Test#2	134 seconds	42 seconds	+ 12 seconds GPU initialization in Matlab

***: Almost constant, overhead GPU initialization time in Matlab.

5 Conclusion

A high-order, stochastic and transformation invariant simulation method, HOSTISIM, is introduced in this paper. This method sequentially simulates the nodes in the grid to be simulated. At each sequence the previous simulated nodes are also considered as conditioning data. Hence, the size of the template shrinks at each new sequence. A high-order statistical disparity vector is introduced to calculate the distance between the data event of a pattern in the grid with a pattern in the TI. This disparity vector is designed to be isotropic and invariant against any linear transformation of the TI patterns. The PDF of the simulating node is then estimated using a likelihood function based on the disparity vector. This method is easy to implement and fast in performance. Since the number of operations is dramatically reduced compared to isotropic L2-norm distance measures, a large TI set can be processed. This method is non-stationary, uses a disparity measure to choose a pattern from the TI, never becomes biased from the TI, and always respects the high-order statistics of the hard data's inherent structure. This method will be expanded to accommodate irregular data locations and implemented to simulate in 3D.

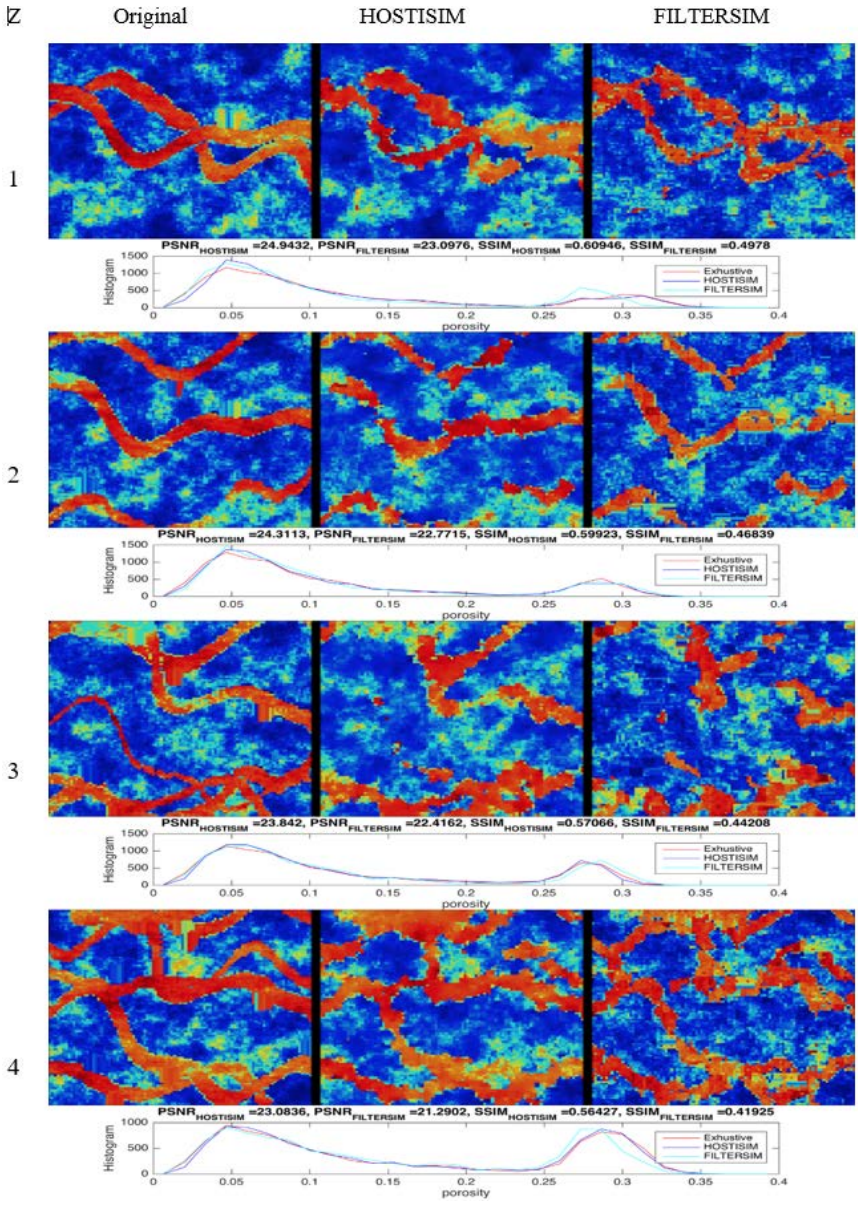

Figure 3. The results generated given 625 hard data (6.25%). From top to bottom, layers Z=1,…4. From left to right, Original image, HOSTISIM and FILTERSIM simulations. On the bottom of each simulation the histogram of each image is presented and the PSNR and SSIM scores of the two methods are also provided.

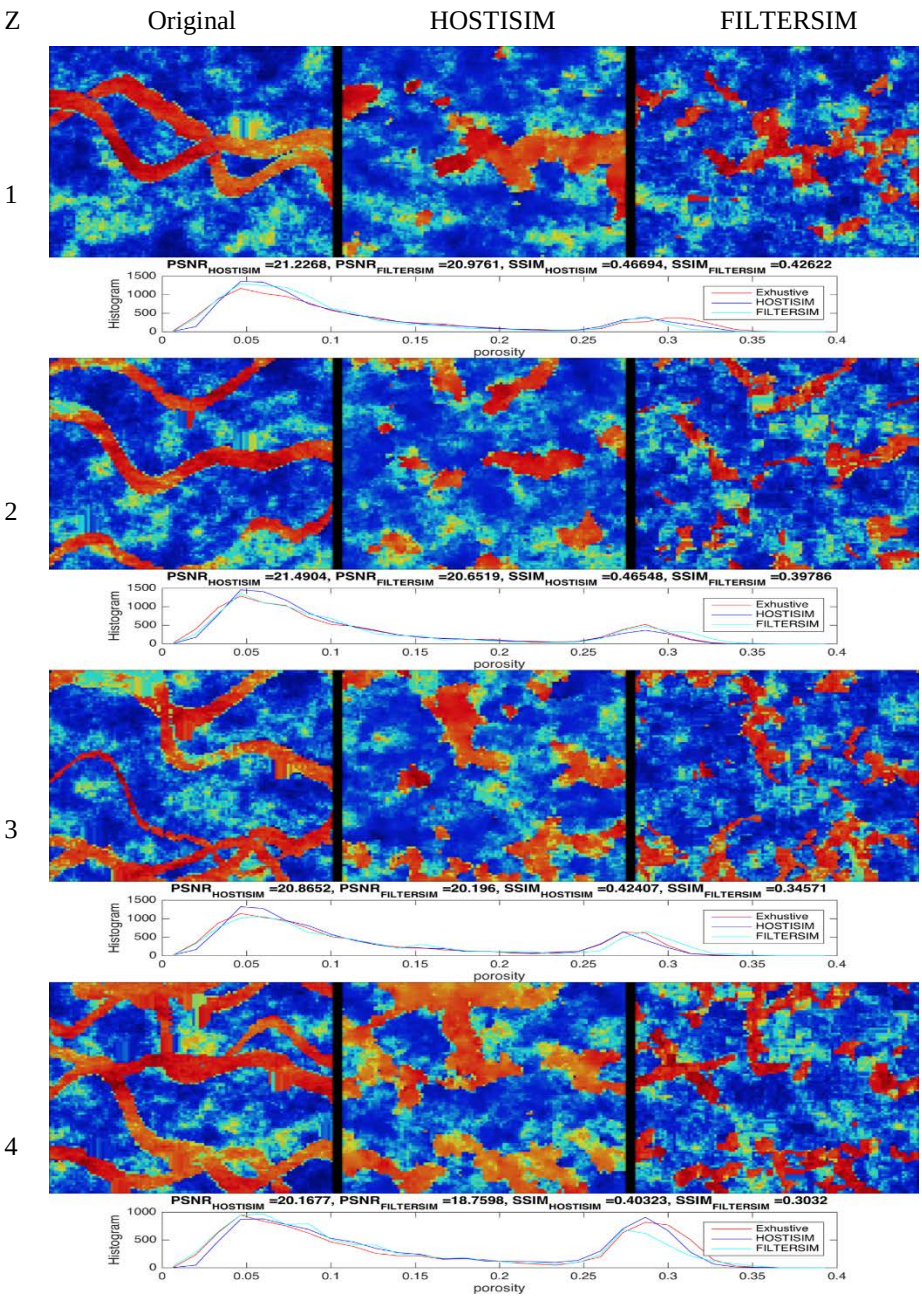

Figure 4. The results generated given 169 hard data (1.69%). From top to bottom, layers Z=1,…4. From left to right, Original image, HOSTISIM and FILTERSIM simulations. On the bottom of each simulation the histogram of each image is presented and the PSNR and SSIM scores of the two methods are also provided.

References

Abdollahifard, M. J. & Faez, K. (2012). Stochastic simulation of patterns using Bayesian pattern modeling. Computational Geosciences, 17(1), 99—116.

Arpat, G. B. & Caers, J. (2007). Conditional simulation with patterns. Math Geology, 39(2), 177—203.

Chatterjee, S., Dimitrakopoulos, R. (2012) Multi-scale stochastic simulation with wavelet-based approach. Computers & Geosciences, 45(3) 177—189

Dimitrakopoulos, R., Mustapha, H., & Gloaguen, E. (2010). High-order statistics of spatial random fields: exploring spatial cumulants for modeling complex non-Gaussian and non-linear phenomena. Math Geosci, 42(1), 65—99.

Funkhouser, H. G. (1930). A short account of the history of symmetric functions of roots of equations. The American Mathematical Monthly, 37(7), 357—365.

Goovaerts, P. (1998). Geostatistics for natural resources evaluation. Cambridge University Press: Cambridge.

Guardiano, F. B. & Srivastava, R. M. (1993). Multivariate Geostatistics: Beyond bivariate moments. Geosatistics Tróia '92, Springer, 133—144.

Honarkhah, M. & Caers, J. (2010). Stochastic simulation of patterns using distance-based pattern modeling. Math Geosciences, 42(5), 487—517.

Mariethoz, G., Renard, P., & Straubhaar, J. (2010). The direct sampling method to perform multiple-point geostatistical simulations. Water Resources Research, 46(11), DOI: 10.1029/2008WR007621.

Minniakhmetov, I. & Dimitrakopoulos, R. (2016). Joint high-order simulation of spatially correlated variables using high-order spatial statistics. Math Geosci, in press.

Minniakhmetov, I. & Dimitrakopoulos, R. (2016) A high-order, data-driven framework for joint simulation of categorical variables. GEOSTAT2016, in this volume.

Mustapha, H. & Dimitrakopoulos, R. (2010a). A new approach for geological pattern recognition using high-order spatial cumulants. Computers & Geosciences, 36(3), 313—334.

Mustapha, H. & Dimitrakopoulos, R. (2010b). High-order stochastic simulation of complex spatially distributed natural phenomena. Math Geosci, 42(5), 457—485.

Mustapha, H. & Dimitrakopoulos, R. (2011). HOSIM: A high-order stochastic simulation algorithm for generating three-dimensional complex geological patterns. Computers & Geosciences, 37(9), 1242—1253.

Mustapha, H., Chatterjee, S., & Dimitrakopoulos, R. (2013). CDFSIM: Efficient stochastic simulation through decomposition of cumulative distribution functions of transformed spatial patterns. Math Geosci, 46(1), 95—123.

Stien, M. & Kolbjørnsen, O. (2011). Facies modeling using a Markov mesh model specification. Math Geosci, 43(6), 611—624.

Strebelle, S. (2002). Conditional simulation of complex geological structures using multiple-point statistics. Math Geology, 34(1), 1—21.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4), 600—612.

Zhang, T., Switzer, P., & Journel, A. (2006). Filter-based classification of training image patterns for spatial simulation. Math Geology, 38(1) 63—80.