Les Cahiers du GERAD ISSN: 0711–2440

Multi-Point Geostatistical Simulation Based on a Quadratic Optimization Algorithm

S. Chatterjee
R. Dimitrakopoulos

G–2013–58

September 2013

Multi-Point Geostatistical Simulation Based on a Quadratic Optimization Algorithm

Snehamoy Chatterjee

Department of Mining Engineering
National Institute of Technology Rourkela
Orissa 769008, India

snehamoy@gmail.com

Roussos Dimitrakopoulos

GERAD & COSMO – Stochastic Mine Planning Laboratory
Department of Mining and Materials Engineering
McGill University
Montreal (Quebec) Canada, H3A 2A7

roussos.dimitrakopoulos@mcgill.ca

September 2013

Les Cahiers du GERAD

G–2013–58

Abstract: The spatial continuity of lithology and ore grade is one of the key factor for proper mine planning. Traditional geostatistical methods are used for spatial modeling of lithology and ore grades. These methods are based on only two-point statistics, which are insufficient to capture geological heterogeneity. The recently developed multi-point methods are performing well in reproducing the spatial continuity; however, most of these algorithms (mostly pattern-based) are not guaranteed the reproduction of the data statistics. In this paper, we propose a support vector machine (SVM)-based multi-point algorithm which ensures the reproduction of data statistics. The SVM-based algorithms are solved after mapping the data at high dimensional space and then by a quadratic optimization technique which provides a global optimum solution for any problem. The proposed method estimates the conditional cumulative density function (*ccdf*) using SVM and the quadratic optimization algorithm. The *ccdf* is generated by thresholding the pattern data base, which is generated from a training image, and calculating the probability of each threshold class by solving regression problem by support vector machine and quadratic optimization algorithm. The method is validated by simulating conditional and unconditional simulation of categorical and continuous training images. We also compare the method with the *snesim* and *filtersim* methods. The results show that our method is performing better than both methods in reproducing the shape of the complex channels. The first- and second-order statistics are well reproduced by the proposed method for all examples.

Key Words: Multi-point statistics, quadratic optimization, support vector machine, training image.

1 Introduction

The spatial continuity of geology and ore grades are key factors for proper mine planning. Geostatistics is a quantitative tool used to generate geological models conditioned to different types of measured data. Traditional geostatistics algorithms are popular due to their computational simplicity (Journel 1983; Goovaerts 1997; Chiles and Delfiner 1999, Haldorsen and Lake 1984; Stoyan et al. 1987; Deutsch and Wang 1996; Holden et al. 1998). However, the traditional geostatistical methods use only two-point statistical algorithms which are insufficient to capture geological heterogeneity.

The multiple-point (mp) simulation techniques (Guardiano and Srivastava 1993; Strebelle 2000) overcome the shortcomings of two-point variogram-based techniques and reproduce the complex spatial continuity. In multi-point statistics, a pattern is defined as multiple-point statistics from a conceptual geological model over a given template of spatial locations (Arpat and Cares 2007). Different multiple-point simulation algorithms include the *snesim* (Strebelle 2002), *filtersim* (Zhang et al. 2006; Wu et al. 2008), *simpat* (Arpat and Caers 2007), Markov random field (Daly 2004; Tjelmeland and Eidsvik 2004), kernel-based simulation (Sarma et al. 2008; Scheidt and Caers 2008), *hosim* (Mustapha and Dimitrakopoulos 2010), *wavesim* (Gloaguen and Dimitrakopoulos 2009; Chatterjee et al. 2012), and the *cdf*-based simulation (Mustapha et al. 2012). Out of these multi-point algorithms, some are statistical-driven algorithm (*snesim*, markov random field, *hosim*, etc) and some are pattern-based algorithm (*simpat, filtersim, wavesim*, etc.) approaches. The advantage of the statistical-driven algorithm over pattern based algorithm is that the statistical-driven algorithm can reproduce the data statistic; however, the pattern-based algorithm doesn't ensure the reproduction of data statistics.

In *snesim* (Strebelle 2002), a statistical-driven approach, the conditional probability is sampled from the training image by searching replicates of the data event. The main limitation of the *snesim* algorithm is that it searches for exact replicates of conditioning data event. Since exact replicates may not always be possible to obtain from the pattern database some conditioning data points from the conditioning data event are deleted. To overcome this limitation of *snesim*, researchers (Tjelmeland 1996; Daly 2004; Tjelmeland and Eidsvik 2004) have proposed another statistical-driven algorithm for mp modeling called Markov random fields (MRF)-based simulation. They have used iteratively the Markov chain Monte Carlo (McMC), and directional Metropolis–Hastings for nonlinear likelihood posterior updating. Kjønsberg and Kolbjørnsen (2008) have applied Markov mesh models for multi-point simulation. However, their algorithm is also restricted to categorical data and stochasticity is limited due to following unilateral path during sequential simulation. Caers (2001) proposed an iterative algorithm for calculating the posterior conditional probability using neural network. However, this iterative algorithm for calculating the probability is performed through a non-linear optimisation problem; therefore, it is impossible to get globally an optimal solution. Dimitrakopoulos et al. (2010) and Mustapha and Dimitrakopoulos (2010) developed spatial cumulants characterization of non-Gaussian non-linear random variables. The limit of this approach is that, at present, the framework is limited to simulating continuous variables. Arpat and Cares (2007) first proposed a pattern-based algorithm for multi-point simulation termed *simpat* (Simulation with Patterns) which is not exactly based on the probabilistic approach. *Simpat* considers the training image as a collection of patterns, from which a pattern can be selected to locally match as close as possible to the conditioning data event. The major limitation of this algorithm is that the entire pattern database will be searched to find the best match at each simulating node; therefore computational time will be extensively high and does not guarantee the reproduction of statistics of the conditioning data. The *filtersim* algorithm (Zhang et al. 2006; Wu et al. 2008) reduces the computational limitations of *simpat* by classifying the patterns into different clusters. However, studies show (Honarkhah and Caers 2010; Chatterjee et al. 2012) that only few filter scores are not sufficient to capture the complex channel patterns. Gloaguen and Dimitrakopoulos (2009) and Chatterjee and Dimitrakopoulos (2011) present a different technique of conditional simulation using the inter-scale dependency at the wavelet domain. The advantage of this approach is that the direct conditioning is easy, but fitting the conditioning data in the wavelet domain is difficult. Chatterjee et al. (2012) proposed a wavelet-based dimensional reduction technique for classification of the pattern database. Honarkhah and Caers (2010) proposed a distance-based simulation algorithm with multi-dimensional scaling (MDS) for efficiently classifying pattern databases. MDS techniques are generally point-to-point mappings and do not provide a generalising mapping function or manifold (Yin 2008). Chatterjee and Dimitrakopoulos (2011) proposed self-organized maps (SOM) based dimension reduction algorithm for pattern-based simulation. However, these algorithms (MDS-based and SOM-based) are iterative algorithms with non-linear optimization; thus don't guarantee any optimum solution. A serious problem with all pattern-based simulation algorithms is that it has a high dependence on pattern frequencies in the training image. The problem is not with the patterns seen in the training image but, rather, with how the method treats patterns that are not present in the training image. In this context, methods using statistical models have an advantage over pattern-based methods, since they can interpolate between observed patterns to compute the probability of patterns that are not present in the training image.

In this paper, we propose an algorithm which is not entirely dependent on the patterns seen in the training image; rather the algorithm develops cumulative conditional distribution function (*ccdf*) by support vector machine algorithm, which is based on statistical learning theory (Vapnik 1995). The random field is simulated sequentially the same as other statistics-driven algorithms. The conditional cumulative density function (*ccdf*) at unknown spatial location is calculated from a pattern database which is generated from the training image. Since the regression is considered as the expected value of an unknown variable conditioning to known data, the *ccdf* is calculated by solving the regression equations of indicator transformed pattern database at different threshold values. The regression value of any indicator transformed data provides the probability of presence of that specific thresholded value. Therefore, by solving different regression problems at different threshold values, the *ccdf* can be estimated. The regression equations for different threshold values are solved by support vector machine algorithm in a high dimensional space because complex patterns may be non-linear in the original domain. The pattern database is mapped into a high dimensional space using a kernel function to make the patterns linear in nature. The quadratic optimization algorithm is used to solve the SVM regression formulation. Quadratic optimization is a special type of mathematical optimization problem which optimizes a quadratic objective function of several variables subject to linear constraints on these variables (Gill et al. 1981). Since the constraints are linear it will ensure the global optimum solution for any regression problem can be obtained (Gill et al. 1981).

The paper is organized as follows. Section 2 describes the proposed simulation method. A brief overview of the problem is presented in Section 2.1. The generation of the pattern database is described in Section 2.1.1. The pattern database transformation into an indicator data is presented in Section 2.1.2. An overview of the support vector machine for regression and the quadratic optimization algorithms formulation for the problem are presented in Section 2.1.3. The *ccdf* calculation of an unknown spatial location is presented in Section 2.1.4. Section 3 presents examples of conditional and unconditional simulation using the proposed algorithm. Conclusion and discussion follow.

2 Method

2.1 Overview of the problem

For geostatistical simulations, the random field is considered as an outcome of a multivariate process $\boldsymbol{Z} = \{Z(u_1), Z(u_2), \ldots, Z(u_N)$ with joint probability distribution function $f_Z(z(u_1), z(u_2), \ldots, z(u_N))$ where $Z(u_i)$, $u_i \in R^n (n = 1, 2, 3)$ $i = 1, 2, \ldots, N$, is a stationary and ergodic random field, and N is the number of discrete points in a random field. The joint probability function f_Z can be written as

$$\begin{aligned} & f_Z(z(u_1), z(u_2), \ldots, z(u_N)) \\ & \quad = f_{Z_1}(z(u_1)) f_{Z_2}(z(u_2) \,|z(u_1)\,) f_{Z_3}(z(u_3) \,|z(u_1), z(u_2)\,) \ldots f_{Z_N}(z(u_N) \,|z(u_1), z(u_2), \ldots, z(u_{N-1})\,) \end{aligned} \quad (1)$$

Assume that the conditional probability for $Z(u_i)$ depends only on a subset Γ_i of all cells $j < i$, such that (Dimitrakopoulos and Luo, 2004)

$$f_{Z_i}(z(u_i) \,|z(u_1), z(u_2), \ldots, z(u_{i-1})\,) = f_{Z_i}(z(u_i) \,|\boldsymbol{z}_{\Gamma_i}\,), \quad (2)$$

where, $\boldsymbol{z}_{\Gamma_i}$ is the geometrical configuration of the set of cells in Γ_i. The set Γ_i denotes the sequential spatial neighbourhood of cell i. The joint probability is defined as

$$f_Z(z(u_1), z(u_2), \ldots, z(u_N)) = \prod_{i=1,2,\ldots,N} f_{Z_i}(z(u_i)\,|\boldsymbol{z}_{\Gamma_i}). \tag{3}$$

Equation (3) is an estimation of joint probability function f_Z by sequentially estimating conditional distribution function with respect to the set of sequential neighbourhoods $\{\Gamma_i\}_{i=1,2,\ldots,N}$.

In this paper, the conditional probability functions $f_{Z_i}(z(u_i)\,|\boldsymbol{z}_{\Gamma_i})$ are estimated in four steps: (a) generating the pattern database from training image using geometrical spatial template, (b) transforming the patterns database into indicator transformed data by selecting M different threshold values; (c) generating M different regression models using support vector machine algorithm; and (d) developing *ccdf* curve from the output of M regression models. The steps involved for *ccdf* calculation are presented in the subsequent sections.

2.1.1 Generation of a pattern database from training image

The geometrical replicates of conditioning data for location u were obtained by scanning the training image and then saved in a pattern database.

Define $ti(u)$ as a value of the training image $\boldsymbol{ti}$ where $u \in G_{ti}$ and G_{ti} is the regular Cartesian grid discretizing the training image, $ti_t(u)$ indicates a specific multiple-point vector of $ti(u)$ within a spatial geometrical template Γ centered at node u, that is

$$ti_\Gamma(u) = \{ti(u+h_1), ti(u+h_2), \ldots, ti(u+h_\alpha), \ldots, ti(u+h_{n\Gamma})\}, \tag{4}$$

where, the h_α vectors are the vectors defining the geometry of the n_Γ nodes of template Γ and $\alpha = \{1, 2, ..., n_\Gamma\}$. The vector $h_1 = 0$ represents the central location u of template Γ. The pattern database is then obtained by scanning $\boldsymbol{ti}$ using template Γ and stored the multi-point $ti_\Gamma(u)$ vectors in the database.

The replicates of geometrical configuration for $\boldsymbol{z}_{\Gamma_i}$ were obtained from the training image. Consider a simple case of estimating conditional probability function $f_{Z_0}(z(u_0)\,|\boldsymbol{z}_{\Gamma_0})$, where the conditional probability functions at location u_0 using the geometrical neighbourhood template Γ_0 which consists of two samples at locations u_1 and u_2 shown in Fig. 1(a). The replicates of template Γ_0 (Fig. 1(b)) were obtained by scanning the training image.

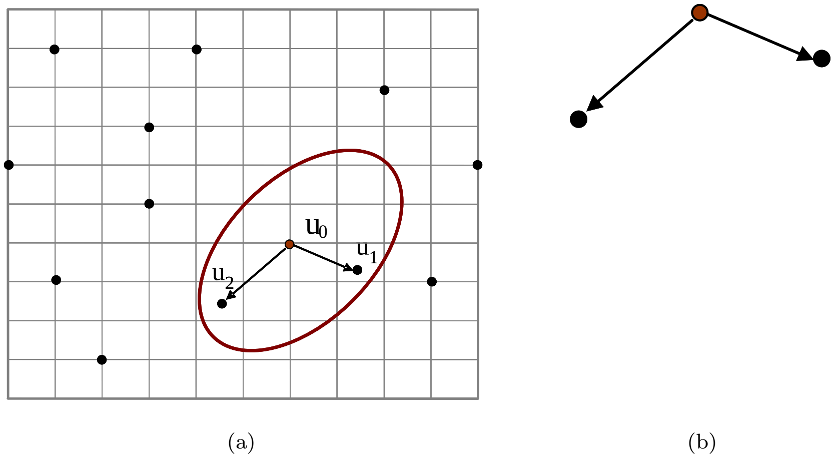


Figure 1: (a) An unknown value is at the location u_0 and the values at the locations u_1, u_2 in the neighbourhood are known; (b) Template used for scanning the training image.

2.1.2 Pattern database to indicators transformation

For a categorical training image with M categories, the pattern data is first transformed into M sets of binary values $I_m(u) = \{I_m(u+h_1), I_m(u+h_2), \ldots, I_m(u+h_\alpha), \ldots, I_m(u+h_{n\Gamma})\}$, $m = 1, \ldots, M, u \in \Gamma$

$$I_m(u+h_\alpha) = \begin{cases} 1, & \text{if } ti(u+h_\alpha) \text{ belongs to } m^{\text{th}} \text{ category,} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

However, for a continuous training image, the pattern is transformed into M sets of binary values $I_m(u) = \{I_m(u+h_1), I_m(u+h_2), \ldots, I_m(u+h_\alpha), \ldots, I_m(u+h_{n\Gamma})\}$ $m = 1, \ldots, M, u \in \Gamma$ by selecting M different threshold values $M = \{z_1, z_2, \ldots, z_m, \ldots, z_M\}$,

$$I_m(u) = \begin{cases} 1, & ti(u+h_\alpha) \leq z_m \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The linear as well as non-linear regression of any indicator data set can be considered as a probabilistic classification. The regression of an indicator data provides an estimate of the probability that the category m or grade value less than equal to z_m at location u given the neighboring rock type information or grade values

$$\Pr\{I_m(u) = 1 | I_{m\Gamma}) = E(I_m(u) | I_{m\Gamma}), \tag{7}$$

where, $I_{m\Gamma}$ is the indicator transformed of spatial template data Γ for location u of m^{th} category or m^{th} threshold value, and E is expected value. The expression of Eq. (7) provides the probability of category m present at location u or the probability of the grade value less than equal to m^{th} threshold value conditional to neighbouring data. Therefore, it is an estimate for the conditional probability of having m^{th} category or m^{th} threshold value at u.

In this paper, we propose to use the support vector machine (SVM) algorithm for solving the regression problem, which relies on quadratic optimization technique. The purpose of the quadratic optimization algorithm is to calculate the parameter values for the SVM model

$$f(I_{m\Gamma}, \boldsymbol{w}) = \Pr\{I_m(u) = 1 | I_{m\Gamma}), \tag{8}$$

where, f is the SVM function and $\boldsymbol{w}$ are parameters estimated using the quadratic optimization algorithm.

2.1.3 Generating regression models using the support vector machine algorithm

Suppose, $\{I_m(u+h_1), I_m(u+h_2), \ldots, I_m(u+h_\alpha), \ldots, I_m(u+h_{n\Gamma})\}^N$ is the indicator transformation of N number of replicates in the pattern database with geometrical template data Γ for location u for m^{th} category or m^{th} threshold value, where $\{I_m(u+h_1)\}^N, h = 0$ is the central node vector, and $\{I_m(u+h_2), \ldots, I_m(u+h_\alpha), \ldots, I_m(u+h_{n\Gamma})\}^N$ represents the neighborhood conditional data matrix. Therefore, if we are able to develop a regression model with $\{I_m(u+h_2), \ldots, I_m(u+h_\alpha), \ldots, I_m(u+h_{n\Gamma})\}^N$ as independent data and $\{I_m(u+h_1)\}^N, h = 0$ as dependent data, then the regression equations, which gives the conditional mean of $\{I_m(u+h_1)\}$ given $\{I_m(u+h_2), \ldots, I_m(u+h_\alpha), \ldots, I_m(u+h_{n\Gamma})\}$, will be able to predict the probability of m^{th} category or m^{th} threshold value at location $(u+h_1)$.

For the regression model, for the sake of simplicity, the independent data $\{I_m(u+h_2), \ldots, I_m(u+h_\alpha), \ldots, I_m(u+h_{n\Gamma})\}^N$, which is the input for the regression model, is defined by $\boldsymbol{x}$, where $\boldsymbol{x} = [x_1, x_2, \ldots, x_i, \ldots, x_N]$ and $x_i = \{I_m(u+h_2), \ldots, I_m(u+h_\alpha), \ldots, I_m(u+h_{nT})\} \in \Re^{n\Gamma-1}$, and the dependent data $\{I_m(u+h_1)\}^N, h = 0$, which is the output for the regression model, is defined by y_i, where $y_i = \{I_m(u+h_1)\} \in \Re$. Therefore, the conditional probability for location $(u+h_1)$ of m^{th} category or m^{th} threshold value can be represented as

$$\Pr\{I_m(u+h_1) = 1 | I_{m\Gamma}) \equiv \Pr\{y = 1 | x) \equiv f(I_{m\Gamma}, \boldsymbol{w}), h_1 = 0. \tag{9}$$

Equation (9) is solved by the SVM algorithm for all m categories or threshold values. The SVM is a supervised learning algorithm based on the structural risk minimization principle (Vapnik, 1995). SVM can perform both classification and regression; however, in this paper, the SVM is strictly applied for regression.

Support Vector Machine algorithm was primarily developed as a classification algorithm; it identifies each class by first projecting the data into a higher dimensional space and then using a hyperplane to separate and classify the projected data. The hyperplane that an SVM learns has a maximized margin amongst all hyperplanes that separate the data. Support Vector Regression is similar to SVM Classification in that it learns a linear regression function in a higher dimensional space. The learnt function deviates the least from the data amongst all such linear surfaces in the expanded space, according to some loss function. This regression surface is taken to be a weighted linear combination of a set of basis functions in the input space (Vapnik, 1995), so the surface is non-linear in the input space and linear in the feature space.

In SVM regression, the input x is first mapped onto a k-dimensional feature space using fixed (nonlinear) mapping, and then a linear model is constructed in this feature space (Hush and Scovel, 2003). The linear model (in the feature space) $f(\boldsymbol{x}, w)$ is given by

$$f(\boldsymbol{x}, w) = \sum_{j=1}^{k} w_j g_j(\boldsymbol{x}) + b, \tag{10}$$

where $g_j(\boldsymbol{x}), j = 1, \ldots, k$ denotes a set of nonlinear transformations, and b is the "bias" term.

The quality of estimation is measured by the loss function $L(y, f(\boldsymbol{x}, w))$. SVM regression uses a new type of loss function called ε-insensitive loss function proposed by Vapnik (1995)

$$L_\varepsilon(y, f(\boldsymbol{x}, w)) = \begin{cases} 0 & \text{if } |y - f(\boldsymbol{x}, w)| \le \varepsilon \\ |y - f(\boldsymbol{x}, w)| - \varepsilon & \text{otherwise.} \end{cases} \tag{11}$$

SVM regression performs linear regression in the high-dimension feature space using ε-insensitive loss and, at the same time, tries to reduce model complexity by minimizing $||\omega||^2$. This can be described by introducing (non-negative) slack variables ξ_i, ξ_i^* $i = 1, \ldots N$, to measure the deviation of training samples outside ε-insensitive zone. Thus SVM regression is formulated as minimization of the following functional

$$\min \frac{1}{2}||\omega||^2 + C \sum_{i=1}^{N} (\xi_i + \xi_i^*), \tag{12}$$

subject to

$$\begin{cases} y_i - f(\boldsymbol{x}_i, \omega) \le \varepsilon + \xi_i^* \\ f(\boldsymbol{x}_i, \omega) - y_i \le \varepsilon + \xi_i \\ \xi_i, \xi_i^* \ge 0, i = 1, \ldots, N \end{cases} . \tag{13}$$

Using a Lagrangian, this optimization problem can be converted into a dual form which is a Quadratic Programming (QP) problem (Gill et al. 1981) where the objective function Ψ is solely dependent on a set of Lagrange multipliers α_i

$$\max \Psi(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} y_i \alpha_i y_j \alpha_j K_{ij}, \tag{14}$$

subject to

$$\begin{matrix} \sum_{i=1}^{N} y_i \alpha_i = 0 \\ 0 \le \alpha_i \le C \end{matrix} , \tag{15}$$

where, $K_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $K(\boldsymbol{x}, \boldsymbol{x}_i) = \sum_{j=1}^{N} g_j(\boldsymbol{x}) g_j(\boldsymbol{x}_i)$. The solution of the above optimization formulation can be solved by any QP optimizer (Gill et al. 1981) and the solution is given by

$$f(\boldsymbol{x}) = \sum_{i=1}^{n_{SV}} (\alpha_i - \alpha_i^*) K(\boldsymbol{x}_i, \boldsymbol{x}) \tag{16}$$

where n_{SV} is the number of Support Vectors (SVs) where the value of $\alpha_i \neq 0$.

A number of kernel functions are available in the literature (Genton, 2001; Hofmann et al. 2008). In this paper, Gaussian kernels are used: $K(x, x_i) = \exp(- \|x - x_i\|^2 / \sigma^2)$, where σ is the bandwidth of the kernel function.

Equation (19) provides the conditional probability of value 1 at the central node $(u + h_1)$ conditioning to its neighbouring data for a specific threshold *m*. Therefore, m $(m = 1, 2, \ldots, M)$ different support vector models are developed to generate the *ccdf*.

2.1.4 Developing conditional cumulative distribution function (ccdf) from M regression models

After calculating the conditional probability for location u for m^{th} category or m^{th} threshold value using the quadratic programming optimization algorithm as discussed in Section 2.1.3, the conditional cumulative distribution (*ccdf*) is calculated. We know that the conditional cumulative distribution function at location u conditioning to its neighbouring data can be represented as

$$\begin{aligned} F_{Z_u}(z(u) = m \,|\boldsymbol{z}_{\Gamma_u}) &= \sum_{m=1}^{m} \Pr\{ I_m(u) = 1 |\, I_{m\Gamma}), \text{ for categorical data,} \\ F_{Z_u}(z(u) = z_m \,|\boldsymbol{z}_{\Gamma_u}) &= \sum_{m=1}^{m} \Pr\{ I_m(u) = 1 |\, I_{m\Gamma}), \text{ for continuous data,} \end{aligned} \tag{17}$$

where, $m = 1, 2, \ldots, M$, M is number of categories or number of threshold values for continuous data. The function F_{Z_u} is a monotonically increasing function and bounded within the range of 0 to 1. However, the solution of Eq. (16) does not ensure that F_{Z_u} probability is within [0 1]. However, experience has shown that the violations are very small. Moreover, it will also not guarantee that the F_{Z_u} will be a continuously increasing function due to order relation violation. In this paper, the Gaussian regression smoothing technique was used to correct the order relation violation.

During the sequential simulation, after finding the *ccdf* F_{Z_u}, a uniform random number is generated. For the categorical simulation, the category corresponding to the generated random number is considered by the simulated category at location *u*. After assigning the simulated category at a simulated node *u*, the next node is visited in random path. The same similarities measured and patterns drawing algorithm is performed until all nodes are simulated. The algorithm stops when no nodes are left un-visited.

In case of continuous data, the *ccdf* is interpolated within each class of thresholds (z_{m-1}, z_m) and extrapolates the *ccdf* values beyond the smallest z-data value and the largest z-data value. In this paper, the linear interpolation and extrapolation models (Goovaerts, 1997; Deutsch and Journel, 1998) were used. To perform the extrapolation within the bound region, the training image as well as hard conditioning data (case of conditional simulation) is scaled within 0 and 1 ([0 1]) before the simulation. After simulation, simulated values scaled back to their original range. If z_i is data value at node i in training image or hard data, $z_{\max} = \max(z_i; i = 1, 2, \ldots, N)$ is the maximum of all z_i, and $z_{\min} = \min(z_i; i = 1, 2, \ldots, N)$ is the minimum of all z_i, then scaled value $\tilde{z}_i$ can be calculated as

$$\tilde{z}_i = \frac{(z_i - z_{\min})}{(z_{\max} - z_{\min})}, i = 1, 2, \ldots, N; N \text{ is number of points in training image.} \tag{18}$$

2.2 The proposed simulation algorithm

The main steps of the proposed algorithm are as follows:

1. Define a random path visiting once and only once all un-sampled nodes.
2. Define the geometrical spatial template Γ for each un-sampled location u using its search neighbours.
3. Scan the training image using the given template Γ, save the extracted patterns in the pattern database.
4. Generate M binary pattern database for M category (for categorical) or M threshold values (for continuous).

5. Develop M number of SVM models as discussed in Section 2.3.
6. Develop *ccdf* as discussed in Section 2.4.
7. Correct for order-relation deviations and draw a random variable from the corrected *ccdf*.
8. Repeat Steps 2–7 for the next node in the random path defined in Step 1.
9. Repeat Steps 1 to 8 to generate different realizations using different random paths.

3 Validation of the Method Using an Exhaustive Data Set

The SVM-based multi-point geostatistical algorithm proposed in this paper is validated by conditional and unconditional simulation of known images. The exhaustive data sets are obtained from different sources. The results of our proposed approach are compared with the *snesim* algorithm for categorical simulation and with *filtersim* for continuous data simulation. To execute *snesim* and *filtersim*, we have used an open-source computer package from Stanford Geostatistical Modeling Software or SGeMS (Remy et al. 2009)

3.1 Unconditional simulation

Unconditional simulations are performed for both the binary training image and continuous training image, and presented herein.

3.1.1 Categorical unconditional simulations

A categorical unconditional simulation is performed for two-category (sand and shale) data sets. The binary training image consists of sand and shale materials (Honarkhah and Caers 2010). This training image represents complex channels present in a deposit. The training image is presented in Fig. 2. The isotropic search neighbourhood is considered for unconditional simulation. The number of conditioning points in the template is restricted to 40 to reduce the computational time. The parameters λ of Eq. (11) and σ for the kernel function K are selected by trial and error basis and the values are 1000 and 0.5 respectively. It is noted that the value of σ plays an important role for trading off the regularising error and fitting error. Therefore, careful selection of σ may improve the results; however that is beyond the scope of the paper.

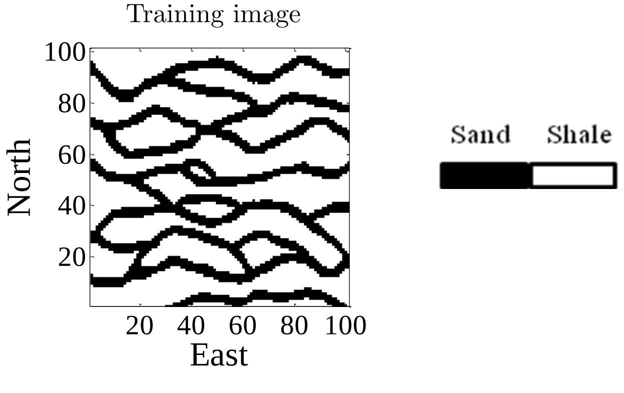
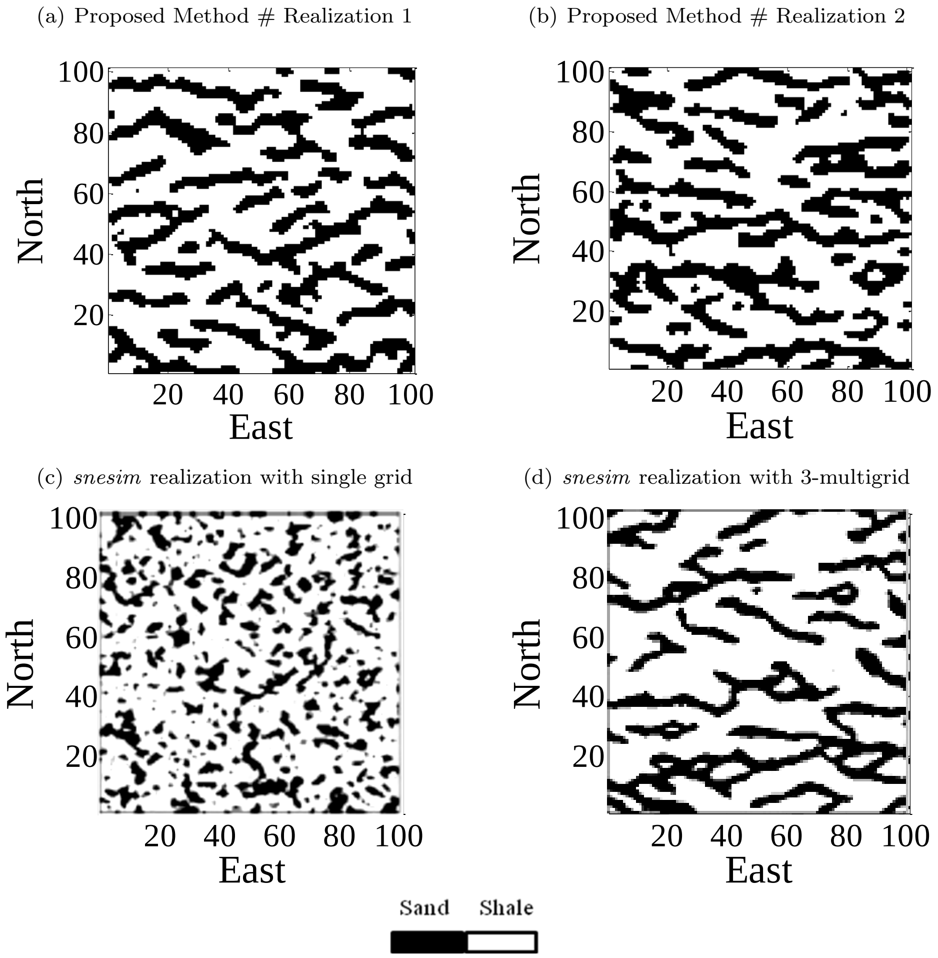


Figure 2: Training image used in this categorical simulation example.

Two simulated realizations of the proposed method are presented in Fig. 3(a and b). It is observed from the figures that our proposed algorithm can reproduce the continuity of channels presented in the training image. Clearly, the method reproduces approximately similar features as in the training images. The results of our proposed approach are then compared with *snesim* approach.

The same template size is used in *snesim* as we have used in our algorithm (40). Our algorithm is developed in single grid; therefore, to make a valid comparison, we have generated single-grid *snesim* simulation results. However, we also compared our results to 3-multigrid *snesim* results. Figure 3 shows a simulated realization of the *snesim* algorithm with single grid (3(c)) and 3-multigrid (3(d)). It is observed from the results that our



(a) Proposed Method # Realization 1

(b) Proposed Method # Realization 2

(c) *snesim* realization with single grid

(d) *snesim* realization with 3-multigrid

Figure 3: Two different simulated realizations using our proposed method (a, b); single grid simulated realization (c), and 3-multigrid simulated realization (d) using *snesim*.

proposed algorithm is performing better as compared to single-grid results of the *snesim* algorithm; however, the 3-multigrid *snesim* is performing equally well as our single-grid algorithm. The need for multigrids for such types of algorithms is clearly understood. The results suggest that the use of multigrids in our proposed algorithm may improve final results. The use of multigrid with our proposed algorithm is under development. The main advantage of our proposed approach is that it can reproduce the statistics of the data set used for simulation. To check how the models reproduce the correct statistics, the proportion of shale-sand, and the variograms of the simulated realizations are compared. The proportion of sand and shale in the training image are 0.325, and 0.675, respectively. The proportions of sand in four simulated realizations of our proposed method are 0.317, 0.329, 0.321, and 0.33, respectively. However, the proportions of sand in four simulated realization of the *snesim* algorithm are 0.28, 0.289, 0.31, and 0.298, respectively. It is observed from the results that our proposed method clearly reproduces the proportion of sand and shale. The directional variograms of the simulated realization generated from our algorithm and training image are calculated and presented in Fig. 4. The variogram results clearly reveal that the proposed algorithm reproduces two point statistics of the training image.

3.1.2 Continuous training image

In this example, we have presented an unconditional simulation for a continuous training image. The training image is obtained from a three-dimensional fluvial reservoir data set (Mao and Journel 1999). One of the slices of the 3-dimensional reservoir data is considered as the training image. Figure 5(a) represents the

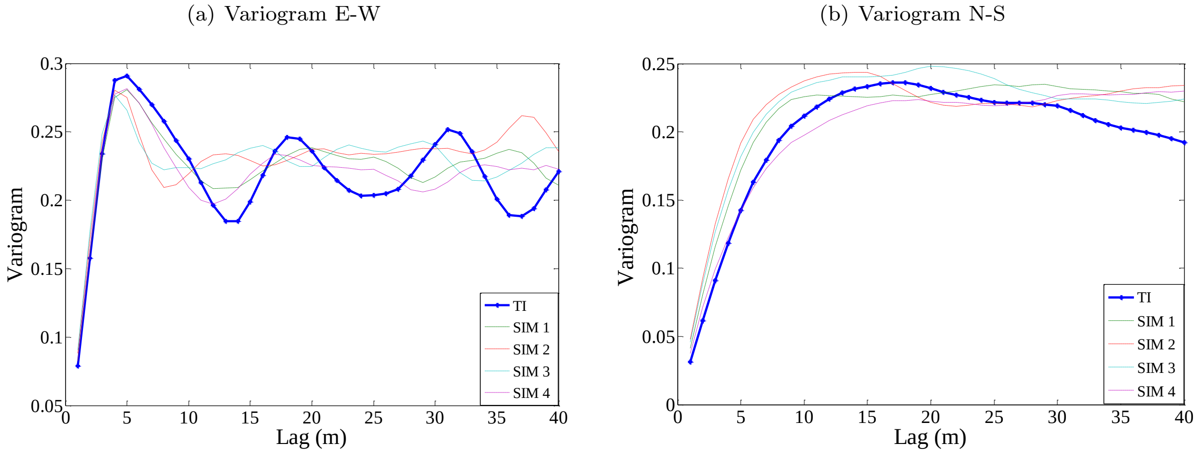


Figure 4: (a) East-West and (b) North-South variogram of simulated realizations (dashed line) by our proposed method and training image (solid line with circle).

training image used for this example. The size of the training image is 100×128. We have simulated the same size domain using our proposed approach. Before performing the simulation, the continuous data are scaled within 0 to 1. The template size used in this study is 80. The number of threshold values used for generating the *ccdf* is 9. All deciles (.1, .2, .3,. . . ,.9) are used for thresholding. The value λ and σ are chosen as 1000 and 0.5, respectively for support vector machine modeling. The results of our proposed algorithm for continuous data are compared with the results of *filtersim* (Zhang et al. 2006; Wu et al. 2008). For generating the *filtersim*, we have used the template size 15 by 15, and number of clusters 200.

Two different realizations generated using our proposed approach and *filtersim* algorithm are presented in Fig. 5(b) to 5(d). Same as in the comparison to *snesim,* we have employed a single-grid in our method comparing to single-grid and multigrid in *filtersim* method. It is observed that our proposed algorithm is performing better than single-grid *filtersim* results as well as 3-multigrid *filtersim* results, as shown in Fig. 5. It is observed from the figure that with our proposed algorithm, channels are well reproduced; however *filtersim* fails to reproduce continuous channels. We have calculated histogram, and variograms of the training image and generated realizations as shown in Fig. 6. This figure shows that our proposed algorithm reproduces both the histogram and the variogram of the training image.

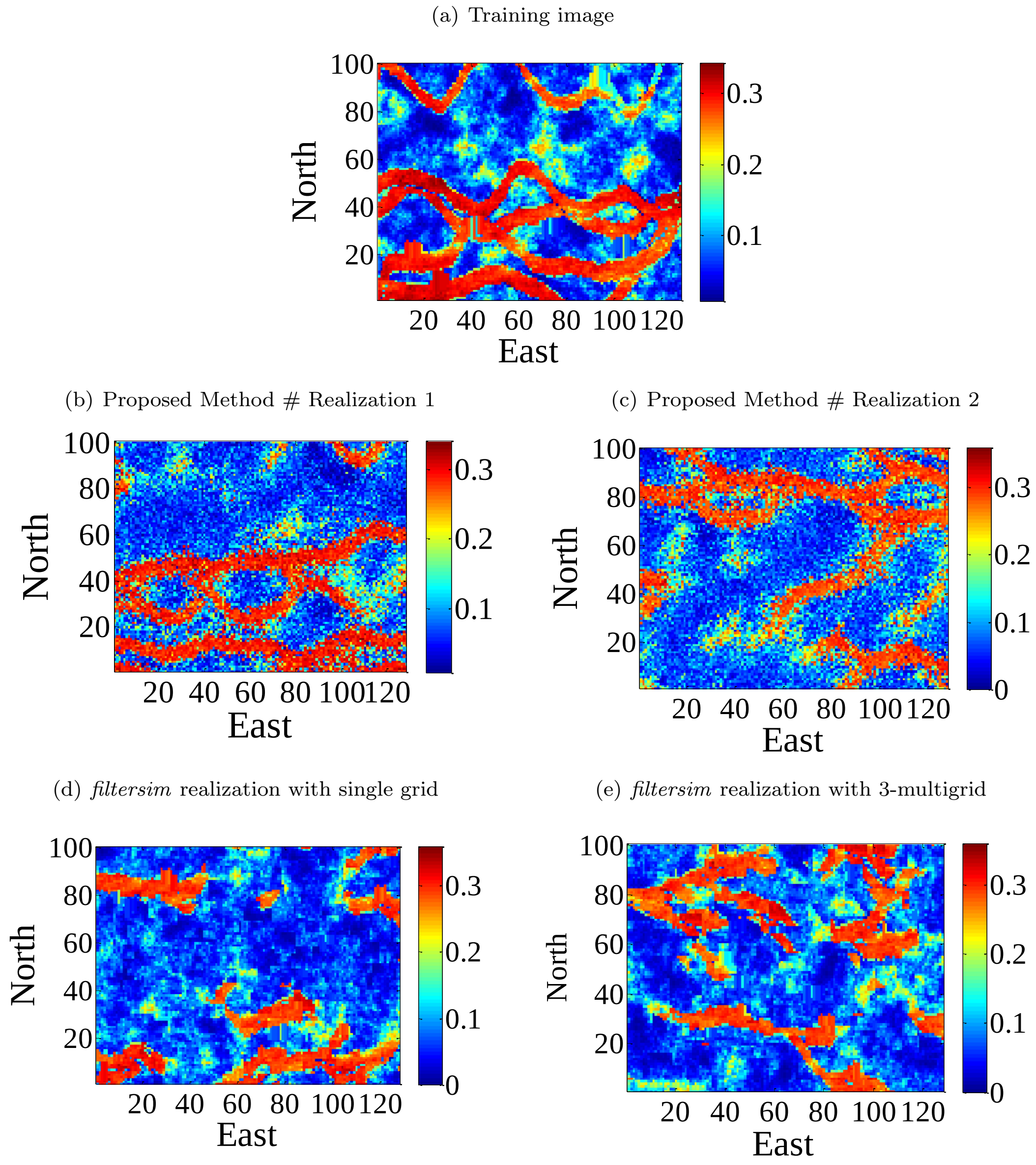
3.2 Conditional simulation

Two different examples are shown for conditional simulation with our proposed method: one two-dimensional categorical and one two-dimensional continuous. The categorical results are compared with the *snesim* and continuous results are compared with the *filtersim* algorithm.

3.2.1 Two-dimensional conditional simulation with continuous data

The same Stanford V Reservoir data set (Mao and Journel 1999) is used for the conditional simulation example. One slice of the three-dimensional reservoir data is used as a reference image from which conditioning data are sampled. Another slice is used as the training image. The reference image to be simulated is presented in Fig. 7(a). The conditioning data set consists of 208 data at irregular spacing scattered all over the domain. Figure 7(b) represents the hard data set used for this study. The training image used in this study is the same one used in the previous example (Fig. 5(a)).

The parameters used for conditional simulation with continuous data are the same as the unconditional simulation of Section 3.1.2 for both the algorithms. The ensemble value of all the generated realizations is used to check the quality of the conditioning. If the algorithm conditioning is performed properly, the ensemble of realizations should provide less uncertainty near hard data locations, and on an average, the


(a) Training image

(b) Proposed Method # Realization 1

(c) Proposed Method # Realization 2

(d) *filtersim* realization with single grid

(e) *filtersim* realization with 3-multigrid

Figure 5: Continuous training image (a); two different simulated realizations using our proposed method (b, c); single-grid realization (d) and multigrid realization (e) using the *filtersim* algorithm.

ensemble map will reproduce a reference image. Five realizations are generated using the proposed method and *filtersim*.

The conditionally simulated realizations generated by our proposed method and the *filtersim* method using the hard data set are presented in Fig. 8. The realization results show that the high valued channels are well reproduced. The comparison study with *filtersim* realizations shows that channel continuity is well reproduced using our proposed approach as compared to single gird as well as 3-multigrid *filtersim*. The ensemble map of our proposed approach and *filtersim* are presented in Fig. 9. The ensemble map shows that our proposed method on an average reproduces the channel shape much better than even 3-multigrid *filtersim* algorithm.

A comparative study of the histograms, variograms of different realizations, and reference image reveals that the first- and second-order statistics are well reproduced by our proposed algorithm, as shown in Fig. 10.

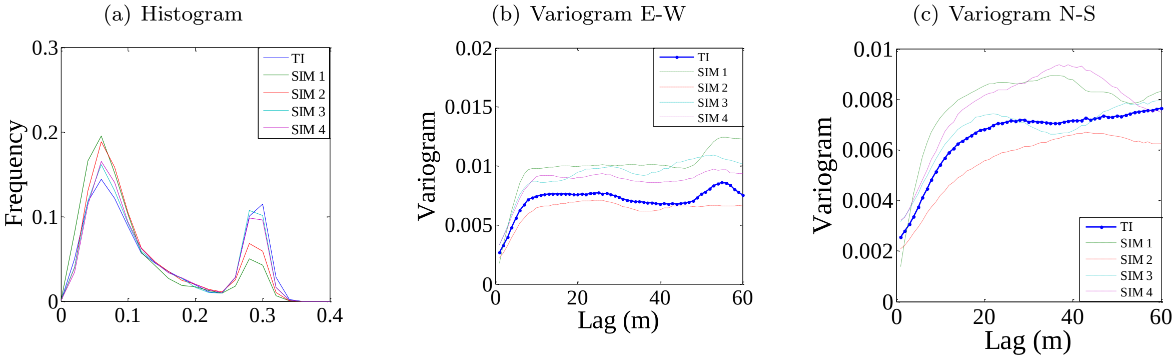


Figure 6: (a) Histogram, (b) East-West and (c) North-South variogram of unconditionally simulated realizations (dashed line) by our proposed method and training image (solid line with circle).

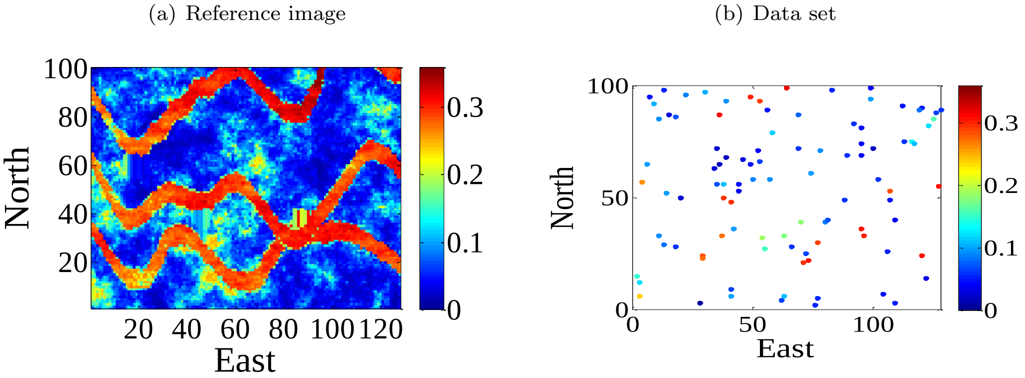


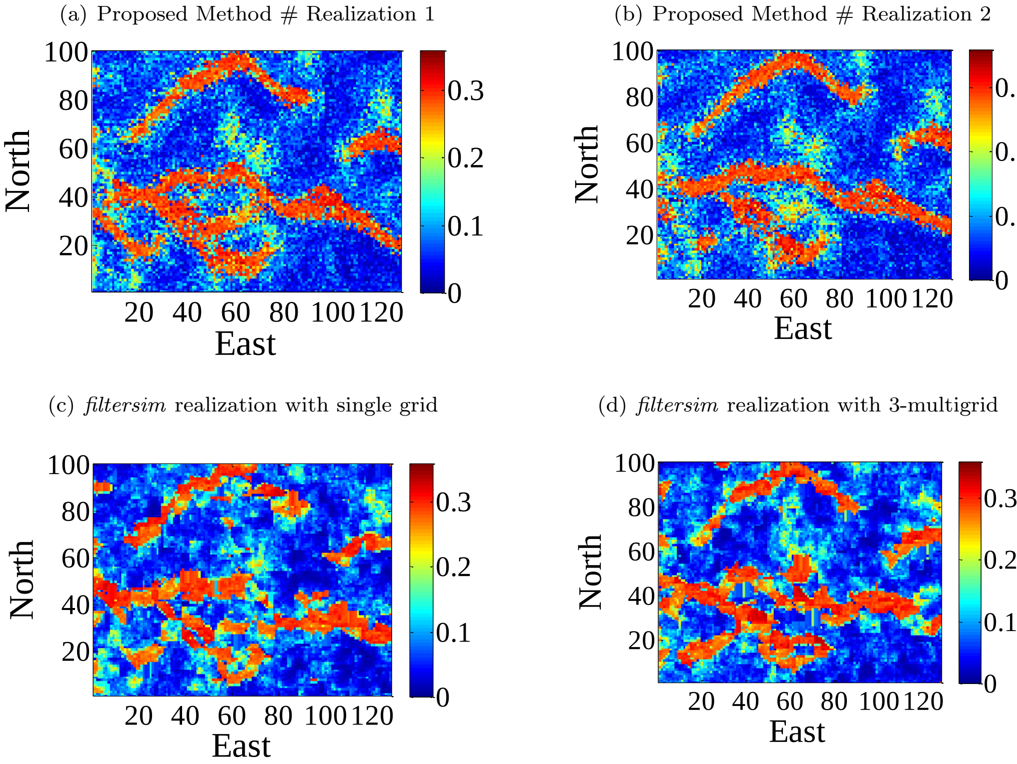
Figure 7: (a) Reference image; (b) Hara data set.

3.2.2 Two-dimensional conditional simulation with categorical data

For conditional categorical simulation, the same training image (Fig. 2) used in the first example is used here. In this example, the hard data are sampled from the training image and only 4 data are sampled as conditioning data (Fig. 11).

The template size used in this study is 40 in both algorithms (our algorithm and *snesim*). The parameters for SVM models are the same as previous examples. Two conditional realizations generated by this study are presented in Fig. 12(a) and (b). The realizations generated show that the channels are well reproduced.

The simulation results are then compared with the *snesim* results. The same hard data and training image are used for simulation in the *snesim*. Figure 12(c) and (d) present single-grid and 3-multigrid realizations of the *snesim*. The results demonstrate that the reproductions of channels are quite better by our proposed algorithm as compared to single-grid *snesim* results; however, when 3-multigrid is used, the *snesim* is performing equally well. The ensemble maps (Fig. 13) revealed that our proposed algorithm is conditioning better than *snesim*, since the ensemble map clearly reproduces the shape of the channels. Figure 14 presents the variogram and histogram of simulated realizations and the reference image. Results demonstrate that the proposed algorithm is clearly reproducing the statistics of the reference image.


(a) Proposed Method # Realization 1

(b) Proposed Method # Realization 2

(c) *filtersim* realization with single grid

(d) *filtersim* realization with 3-multigrid

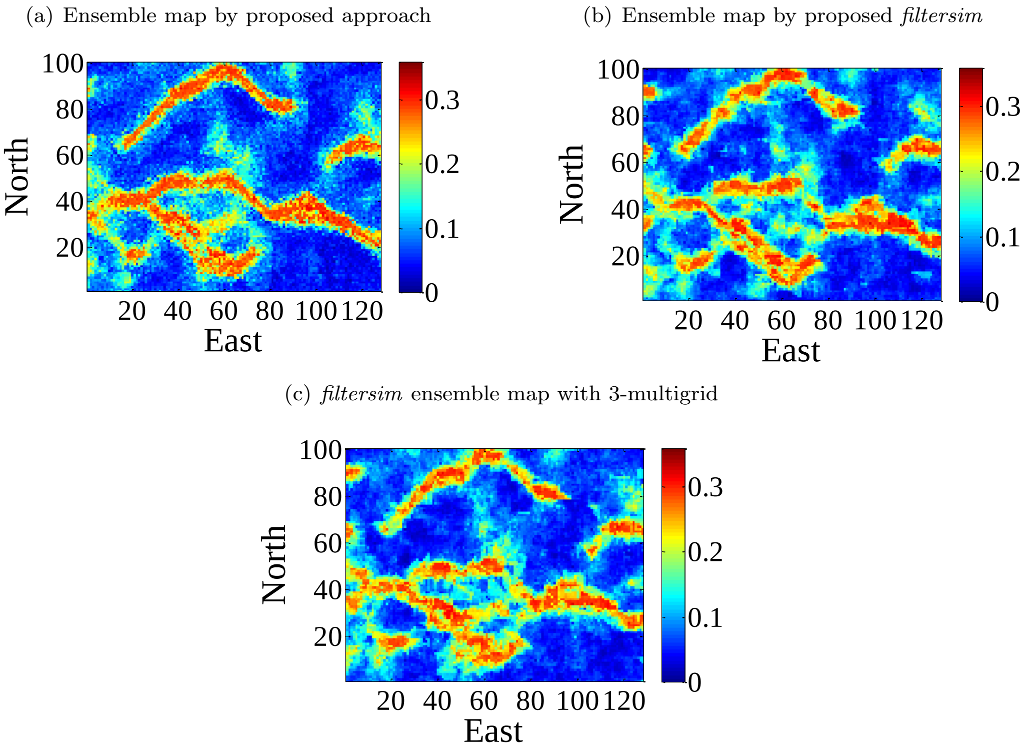
Figure 8: Two different simulated realizations using our proposed method (a, b); single-grid realizations (c) and multigrid realization (d) using the the *filtersim* algorithm.

4 Conclusions

A new multi-point simulation algorithm is proposed in this paper. The algorithm uses quadratic optimisation-based support vector machine algorithm to generate the *ccdf* at unknown point conditioning to its neighbouring data. The technique is based on generating the database by extracting the replicates of template geometry from the training image. The indicators transformed of the pattern data are used to calculate the probability value of specific threshold using a support vector machine algorithm. The main advantage of applying the SVM is that it solves the problem with quadratic optimisation which ensures the optimum solution for the problem. The probability values at different threshold are used to calculate the *ccdf*. The algorithm is verified by two- dimensional conditional and unconditional simulation using different data. The algorithm reproduced the continuity of the channels for two-dimensional examples of conditional and unconditional simulation. The comparative study with the *snesim* and *filtersim* algorithm showed that the proposed algorithm performed better than the *snesim* and *filtersim* algorithm for reproducing the channels.

The major advantage of the proposed algorithm is that the chances of reproduction of statistics are much better than *filtersim* as well as *snesim*. Since our algorithm is based on generating the exact *ccdf* from the conditioning data by convex optimization, the solution is a unique.

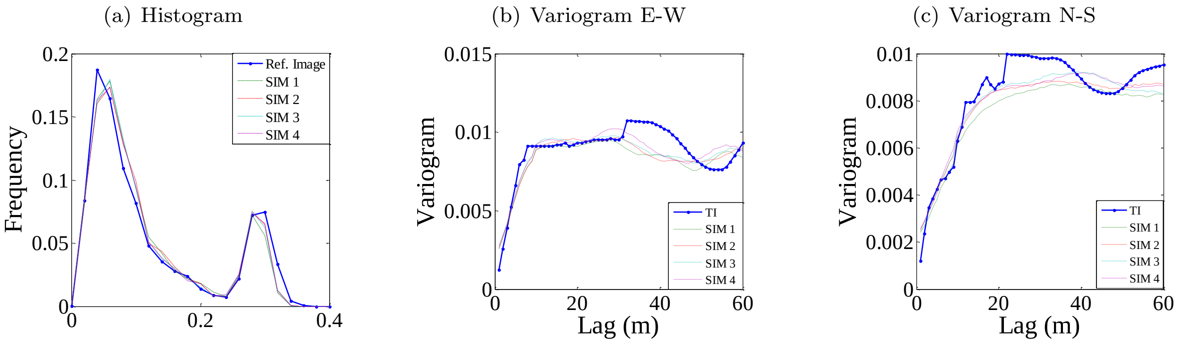
The main limitation of the proposed algorithm is that the computational time is significantly high. As simulation progresses sequentially, the number of conditioning data are significantly increased; and thus computational time. However, we have made some effort to reduce computational time by restricting the maximum number of conditioning data for simulating a point; it is noted however that this may reduce the accuracy of our algorithm. By mapping the high dimensional conditioning data into another domain with significantly less number of mapped data while preserving the data variability may reduce the computational time as well as maintaining the quality of results.


(a) Ensemble map by proposed approach

(b) Ensemble map by proposed *filtersim*

(c) *filtersim* ensemble map with 3-multigrid

Figure 9: Ensemble map of (a) the proposed approach, (b) the *filtersim* with single grid, and (c) the *filtersim* with 3-multigrid approach generated from 5 conditionally simulated realizations.


(a) Histogram

(b) Variogram E-W

(c) Variogram N-S

Figure 10: (a) Histogram, (b) East-West and (b) North-South variogram of conditionally simulated realizations (dashed line) by our proposed method and training image (solid line with circle).

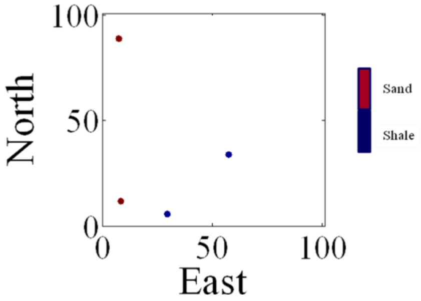
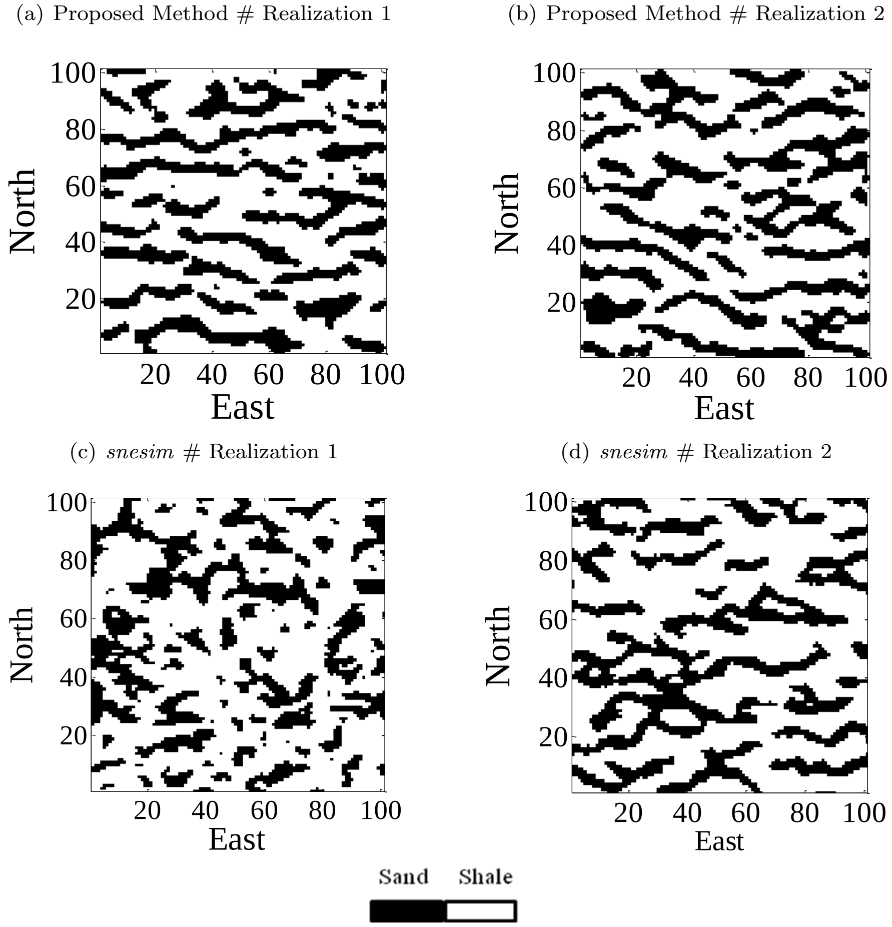


Figure 11: Hara data location for categorical simulation.

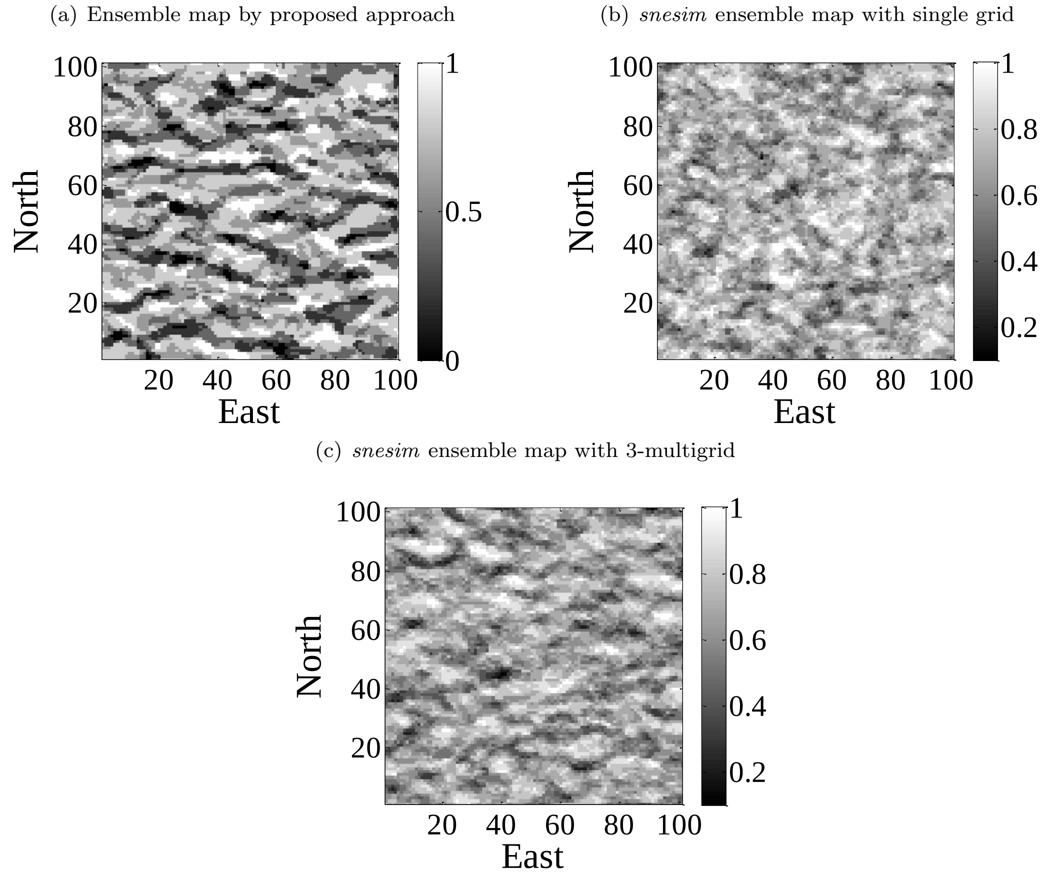

(a) Proposed Method # Realization 1

(b) Proposed Method # Realization 2

(c) *snesim* # Realization 1

(d) *snesim* # Realization 2

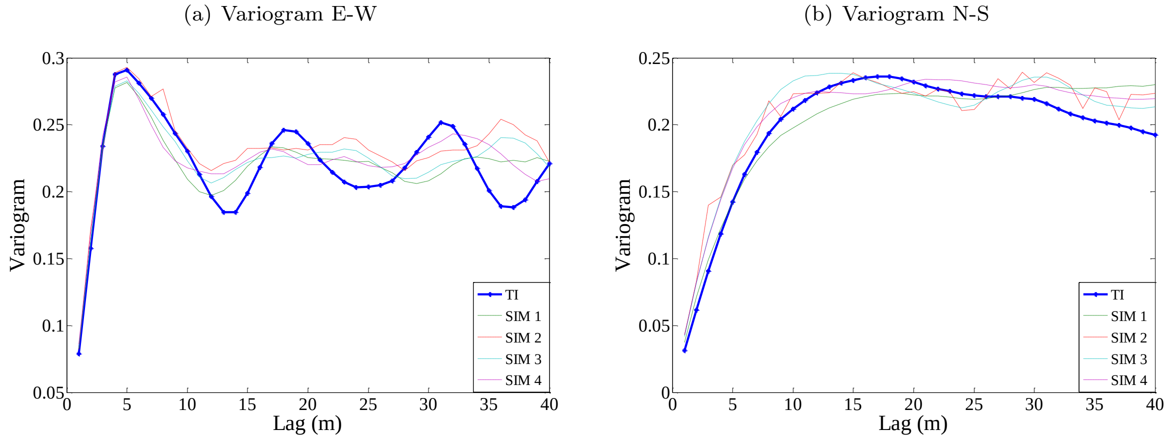
Figure 12: Two different simulated realizations using our proposed method (a, b); single grid simulated realization (c), and 3-multigrid simulated realization (d) using *snesim*.


(a) Ensemble map by proposed approach

(b) *snesim* ensemble map with single grid

(c) *snesim* ensemble map with 3-multigrid

Figure 13: Ensemble map of (a) the proposed approach, (b) the *snesim* with one-multigrid, and (c) the *snesim* with 3-multigrid approach generated from 5 conditionally simulated realizations.


(a) Variogram E-W

(b) Variogram N-S

Figure 14: (a) East-West and (b) North-South variogram of simulated realizations (dashed line) by our proposed method and training image (solid line with circle).

References

Arpat G, Caers J (2007) Conditional simulation with patterns. Mathematical Geology 39(2): 177–203.

Arpat GB (2004) Sequential simulation with patterns. PhD thesis, Stanford University.

Caers J (2001) Geostatistical reservoir modelling using statistical pattern recognition, Journal of Petroleum Science and Engineering 29: 177–188.

Chatterjee S, Dimitrakopoulos R (2011) Multi-scale stochastic simulation with a wavelet-based approach. Computer and Geosciences. doi:10.1016/j.cageo.2011.11.006.

Chatterjee S, Dimitrakopoulos R (2011) Pattern-based Simulation using Self Organized Maps, COSMO Research Report 5(2).

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

Chiles JP, Delfiner P (1999) Geostatistics: modeling spatial uncertainty. Wiley, New York.

Daly C (2004) Higher order models using entropy, Markov random fields and sequential simulation. In: Leuangthong O, Deutsch CV (eds) Geostatistics, Banff 2004. Springer, Dordrecht, pp. 215–224.

Deutsch CV, Journel AG (1998) GSLIB: Geostatistical software library and user's guide. Oxford University Press, New York.

Deutsch CV,Wang L (1996). Hierarchical object-based geostatistical modeling of fluvial reservoirs. Paper SPE 36514 presented at the 1996 SPE Annual Technical Conference and Exhibition, Denver, Oct 6–9.

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. Mathematical Geosciences 42(1): 65–99.

Dimitrakopoulos R, Xiaochun Luo (2004) Generalized Sequential Gaussian Simulation on Group Size ν and Screen-Effect Approximations for Large Field Simulations, Mathematical Geology 36(5): 567–591.

Genton MG (2001) Classes of kernels for machine learning: A statistics perspective. Journal of Machine Learning Research 2: 299–312.

Gill PE, Murray W, Wright, MH (1981) Practical Optimization, Academic Press, London, UK.

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

Goovaert P (1997) Geostatistics for Natural Resources Evaluation (Applied Geostatistics Series). Oxford University Press, Oxford.

Guardiano F, Srivastava RM (1993) Multivariate, geostatistics: beyond bivariate moments. In: Soares A (ed) Geostatistics Troia. Kluwer Academic, Dordrecht, pp. 133–144.

Haldorsen HH, Lake LW (1984) A new approach to shale management in field-scale models. Soc Pet Eng J 24(8): 447–452.

Hofmann T, Schölkopf B, Smola AJ (2008) Kernel methods in machine learning. Ann. Statist. 36(3): 1171–1220.

Holden L, Hauge R, Skare O, Skorstad A (1998) Modeling of fluvial reservoirs with object models. Math Geology 30(5): 473–496.

Honarkhah M and Caers J (2010) Stochastic simulation of patterns using distance-based pattern modelling. Mathematical Geosciences 42: 487–517.

Journel AG (1983) Non-parametric estimation of spatial distributions. Math Geology 15(3): 445–468.

Journel AG (1997) Deterministic geostatistics: a new visit. In: Baafy E, Shofield N (eds) Geostatistics Woolongong '96. Kluwer, Dordrecht, pp. 213–224.

Hush D, Scovel C (2003) Polynomial-time decomposition algorithms for support vector machines. Machine Learnin, 51: 51–71.

Kjønsberg H, Kolbjørnsen O (2008) Markov mesh simulations with data conditioning through indicator kriging. In: Proceedings of the Eighth International Geostatistics Congress, Santiago, Chile.

Mao S, Journel AG (1999) Generation of a reference petrophysical and seismic 3D data set: The Stanford V reservoir, in Stanford Center for Reservoir Forecasting Annual Meeting. Available at: http://ekofisk.stanford.edu/SCRF.html.

Mustapha H, Dimitrakopoulos R (2010) High-order stochastic simulation of complex spatially distributed natural phenomena, Mathematical Geosciences, 42(5): 457–485.

Mustapha H, Chatterjee S. Dimitrakopoulos R (2012) Efficient Pattern-based Spatial Simulation through Decomposition of Cumulative Distribution Functions of Transformed Spatial Patterns, Mathematical Geosciences (Submitted after revision).

Remy N, Boucher A, Wu J (2009) Applied geostatistics with SGeMs: a users's guide. Cambridge University Press, Cambridge.

Sarma P, Durlofsky LJ, and Aziz K (2008) kernel Principal Component Analysis for efficient, differentiable Parameterization of Multipoint Geostatistics. Mathematical Geosciences 40(1): 3–32.

Scheidt C, Caers J (2008) Representing spatial uncertainty using distances and kernels. Math Geosciences 41(4): 397–419.

Stoyan D, Kendall WS, Mecke J (1987) Stochastic geometry and its applications. Wiley, New York.

Strebelle S (2000) Sequential Simulation Drawing Structures from Training Images. PhD thesis, Stanford University.

Strebelle S (2002) Conditional simulation of complex geological structures using multiplepoint statistics. Mathematical Geology 34(1): 1–21.

Tjelmeland H (1996) Stochastic Models in reservoir characterization and Markov random fields for compact objects. Doctoral Dissertation, Norwegian University of Science and Technology. Trondheim, Norway.

Tjelmeland H, Eidsvik J (2004) Directional metropolis-Hastings updates for posteriors with non linear likelihood. In: Leuangthong O, Deutsch CV (eds) Geostatistics, Banff. Springer, Dordrecht, pp 95–104.

Vapnik VN (1995) The Nature of Statistical Learning Theory. Springer, New York.

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

Yin H (2008) On multidimensional scaling and embedding of self-organising maps. Neural Networks 21: 160–169.

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