2020

A Novel Pixel-based Multiple-Point Geostatistical Simulation Method for Stochastic Modeling of Earth Resources

Adel Asadi
Michigan Technological University, aasadi@mtu.edu

A NOVEL PIXEL-BASED MULTIPLE-POINT GEOSTATISTICAL SIMULATION METHOD FOR STOCHASTIC MODELING OF EARTH RESOURCES

By

Adel Asadi

A THESIS

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In Mining Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2020

This thesis has been approved in partial fulfillment of the requirements for the Degree of MASTER OF SCIENCE in Mining Engineering.

Department of Geological and Mining Engineering and Sciences

Thesis Advisor: *Dr. Snehamoy Chatterjee*

Committee Member: *Dr. Rouhollah (Radwin) Askari*

Committee Member: *Dr. Nathan Manser*

Department Chair: *Dr. Aleksey Smirnov*

Table of Contents

List of Figures

Preface

The author of this thesis is the primary author in the paper included in Chapter 2 of the thesis. Dr. Snehamoy Chatterjee, the second author of the paper, is the first author’s advisor for the Master of Science degree in the Mining Engineering program. Dr. Chatterjee is Assistant Professor of Geological and Mining Engineering at the Geological and Mining Engineering and Sciences Department of Michigan Technological University.

Acknowledgements

I would like to appreciate the great help and support by my kind advisor, Dr. Snehamoy Chatterjee, during my master’s study at Michigan Tech. I can never appreciate enough for his trust and patience while I was acquiring knowledge and building my skills to improve my academic capabilities, and to develop what I have achieved during my studies. The transformation and enhancement of my abilities in handling different aspects of undertaking a research project since I have attended Michigan Tech is absolutely significant. It was a great honor for me to study under his supervision, and the learning experience was really great.

In addition, I would like to express my gratitude to the Geological and Mining Engineering and Sciences Department at Michigan Technological University for the full financial support provided to me in order to attend the Mining Engineering MS program, and to continue my education towards graduation. It was a dream for me to study in the USA in such a prestigious program, and it wouldn’t be possible without the generous support of the Graduate School and the GMES Department at Michigan Tech. I want to specifically thank Prof. John Gierke for his aids and support, and the excellent faculty members for their remarkable teaching activities.

I also want to thank all individuals who helped me achieving this goal. My unlimited gratitude goes to my family and friends, who helped me coming to this great country to pursue higher education. I must especially appreciate my father for forming my ambitious character, my mother for her compassion, my brother for his sacrifice and responsible manner, and my great uncle, Dr. Bagheri, for helping me and my family pass extremely difficult situations.

List of Abbreviations

2D	: Two-Dimensional
3D	: Three-Dimensional
%	: Percentage
MPS	: Multiple-Pint Geostatistics
t-SNE	: t-Distributed Stochastic Embedding
DBSCAN	: Density-based Spatial Clustering for Applications with Noise
ccdf	: Conditional Cumulative Density Function
PCA	: Principal Component Analysis
EPS (ε)	: Epsilon
MinPts	: Minimum Number of Points to Form a Cluster
SD	: Standard Deviation
KL	: Kullback–Leibler Divergence
E-type	: Ensemble Plot

Abstract

Uncertainty is an integral part of modeling Earth's resources and environmental processes. Geostatistical simulation technique is a well-established tool for uncertainty quantification of earth systems modeling. Multiple-point statistical (MPS) algorithms are specifically advantageous when dealing with the complexity and heterogeneity of geological data. MPS algorithms take advantage of using training images to mimic physical reality. This research presents a novel and efficient pixel-based multiple-point geostatistical simulation method for mineral resource modeling. Pixel-based simulation implies the sequential modeling of individual points on the simulation grid by borrowing spatial information from the training image and honoring conditioning data points. The developed method borrows information by integrating multiple machine learning algorithms, including Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), and Density-based Spatial Clustering of Applications with Noise (DBSCAN) algorithms. For automation and to ensure high-quality realizations, multiple optimizations, and parameter tuning strategies were introduced. The proposed methodology proved its applicability by accurate reproduction of complex geological features honoring conditioning data while maintaining reasonable computational time. The model is validated by simulating a variety of categorical and continuous variables for both two and three-dimensional cases and conditional and unconditional simulations. As a three-dimensional case study for categorical stochastic modeling, the proposed method is applied to a gold deposit for orebody modeling. The proposed algorithm can be applied to a variety of contexts, including but not limited to petroleum reservoir characterization, seismic inversion, mineral resources modeling, gap-filling in remote sensing, and climate modeling. The developed model can be extended for spatio-temporal modeling, multivariate simulation, non-stationary modeling, and super-resolution realizations.

1. Introduction

1.1. Overview

Modeling subsurface geological features has been been an important topic in mathematical geosciences. The geostatistical simulation is a useful tool in stochastic modeling of Earth systems by providing equiprobable realizations. Geostatistics can be used to analyze the data effectively, and it has also opened its way to many other fields of studies for spatial modeling. Lack of data is an issue in the geoscience applications, which causes significant uncertainty in these problems. Kriging, as one of the most used conventional geostatistical tools, was developed to encounter such spatial modeling applications. Traditional variogram-based simulation of spatial patterns, e.g. Kriging, as a two-point statistical simulation method, has been criticized by its misrepresentation of geological information. The main drawback of the two-point based geostatistical simulations is their weakness in reproducing complex and heterogeneous spatial structures. In particular, these methods cannot convey connectivity and variability when the considered phenomenon contains definite patterns or structures. Increasing the number of points can help in reproducing the connectivity and complex features (Tahmasebi, 2018).

In the multiple-point statistical (MPS) methods, the statistics of data structures are not extracted via variograms or covariance matrices, but a conceptual tool named training image (TI), which is an example of the spatial structure isreproduced. These methods have proven their applicability in producing accurate simulations of geological domain with high complexity. The main reason for the use of training image as the expected formation to be simulated is to deal with the lack of available measurement data (Strebelle, 2012). The training images should represent all possible patterns and shapes of the formation. Thus, providing a representative TI, or a set of TIs, is one the most critical steps in the MPS simulations. The best and fastest way to generate training images is to perform unconditional reconstruction by object-based models (Strebelle, 2012). In general, TIs can be generated using the physics derived from process-based methods or statistical methods or by using the extracted and observed rules for each geological system. In a broader sense, TI can be constructed based on traditional statistical methods. These outcomes, however, do not represent the deterministic aspects of geological models, as they usually tend to signify the randomness, and thus, most of the images represent some degree of complexity and uniqueness in term of spatial patterns. The TI can be an image of statistical properties in space and time, and the subsequent MPS analyses can be performed both spatially and temporally (Tahmasebi, 2018).

The MPS algorithms can be classified into two groups: pattern-based and pixel-based simulationseach of which has its own advantages and disadvantages. In general, selecting the best geostatistical method would be based on the balance between making good physical realism, computational efficiency, and honoring conditioning data (Mariethoz and Caers, 2015). The pixel-based MPS algorithms are generally considered as computationally demanding methods (Tahmasebi et al., 2014). The pattern-based algorithms are recognized for better reproduction of large-scale features, but having limitation in honoring conditioning data. The pattern-based methods simulate a group of points at a time, while the pixel-based methods work with simulating every single point on the simulation grid by considering its surroundings.

Currently, the available MPS algorithms can be grouped based on their computational efficiency. Some of the fast methods include SNESIM (Strebelle, 2002), IMPALA (Straubhaar et al., 2013), HOSIM (Mustapha and Dimitrakopoulos, 2011), DISPAT (Honarkhah and Caers, 2010), CCSIM (Tahmasebi et al., 2012; Tahmasebi and Sahimi, 2015) and Image Quilting (Efros and Freeman, 2001; Mahmud et al., 2014). Medium-speed methods include WAVESIM (Chatterjee et al., 2012), FILTERSIM (Zhang et al., 2006), and Direct Sampling (Mariethoz et al., 2010). ENESIM (Guardiano and Srivastava, 1993), SIMPAT (Arpat and Caers, 2007), and Simulated Annealing (Peredo and Ortiz, 2011) could be named as relatively slow MPS algorithms (Mariethoz and Caers, 2015). Due to the high interest and need in using MPS algorithms in various disciplines to solve different problems, the MPS methods are under great developments, and many MPS algorithms and versions have already introduced.

1.2. Objectives

One of the important aspects of multiple-point geostatistical modeling is to encounter the analysis of high-dimensional data. In the past, some algorithms have been proposed to reduce the dimensionality of the MPS patterns database (Zhang et al., 2006; Wu et al., 2008; Honarkhah and Caers, 2010; Chatterjee et al., 2012). However, some of them are computationally expensive, and some do not maintain the original diversity and distance after dimensionality reduction.

Another major step in some MPS models was introduced by clustering patterns database extracted from the training image. Many MPS models have used clustering algorithms as unsupervised machine learning techniques for reducing the number of members within the patterns database to a reasonable amount. For instance, one of the most famous and widely used ones is the k-means clustering algorithm, which is an unsupervised learning algorithm. It is a simple fast and distance-based clustering technique. However, the limiting criterion of this algorithm is that users need to decide the number of clusters in advance to provide it to the method. It is not viable to decide on the number of clusters of image patterns within the training image is a certain way.

The main goal of this research is to address the two mentioned limitations above by the implementation of advanced machine learning algorithms within the MPS algorithms. In this study, I present a novel, high-speed, pixel-based MPS method thatrepresents the structures of the training images in realizations while honoring the hard data in reconstructions. The main idea of this research is to use a dimensionality reduction technique, named t-Distributed Stochastic Neighbor Embedding (t-SNE), and subsequent unsupervised clustering of the data using Density-based Clustering of Applications with Noise algorithm (DBSCAN). The t-SNE algorithm is a widely used dimensionality reduction method for visualization and mapping of high-dimensional data. The patterns from the database are mapped on a two-dimensional Cartesian environment for subsequent application of DBSCAN for clustering. This thesis utilizes DBSCAN for the purpose of clustering the patterns database. In addition, this research proposes a method to automate the input parameters selection for the MPS method. The proposed methodology is validated and applied to different scenarios In addition, a three-dimensional case-study of resource modeling using the proposed methodology is provided.

1.3. Outline

The thesis is organized in the following manner:

Chapter 1: An overview of the multiple-point geostatistics and its latest simulation methods are presented in this chapter, and some details and limitations of the previously proposed methods are discussed.

Chapter 2: A novel pixel-based multiple-point geostatistical simulation is presented and validated via different training images. A three-dimensional case study is also presented as a validation and application of the proposed method.

Chapter 3: Overall conclusions and recommendations for future work are presented.

2. Stochastic Embedding and Density-based Clustering of Image Patterns for Pixel-based Multiple-Point Geostatistical Simulation

Adel Asadi [1], Snehamoy Chatterjee [1]

[1] *Geological and Mining Engineering and Sciences Department, Michigan Technological University, 1400 Townsend Drive, Houghton, MI 49931, USA.*

(The material contained in this chapter is submitted for review and possible publication in *Mathematical Geosciences* Journal – Springer Publications)

Abstract

Multiple-point simulation (MPS) algorithms are established tools for uncertainty quantification of earth systems modeling, particularly when dealing with the complexity and heterogeneity of geological settings. This study presents a novel pixel-based MPS method for modeling spatial data using advanced machine learning algorithms.The pixel-based multiple-point simulation implies the sequential modeling of individual points on the simulation grid, one by one, by borrowing spatial information from the training image and honoring hard data points. The developed methodology is based on the mapping of the database of training image patterns using the t-Distributed Stochastic Neighbor Embedding (t-SNE) algorithm for dimensionality reduction. Then,the clustering of the patterns will be done by applying Density-based Spatial Clustering of Applications with Noise (DBSCAN) algorithm, as an efficient unsupervised classification technique. For automation, optimization, and input parameters tuning, we have implemented multiple stages to ensure the proposed method does not require the user's interference. Theynclude entropy-based determination of template size, and k-nearest neighbors search for clustering parameter selection. he proposed model, which shows acceptable accuracy and speed in the stochastic simulation, is validated using synthetic two- and three-dimensional data sets, both for conditional and unconditional simulations. Finally, the proposed method is applied to a

case study gold mine for stochastic orebody modeling. The runtime information are also provided for all synthetic data sets and the gold mine case study.

Keywords

Multiple-Point Geostatistics; Pixel-based Simulation; Dimensionality Reduction; Image Patterns Clustering; Parameters Tuning

2.1. Introduction

Since its introduction to the scientific community, geostatistical modeling has been significantly helpful in the stochastic simulation of geological systems by making realizations of the systems based on the conditional probability concept. Traditional variogram-based simulation of spatial patterns, which considers two-point statistics, has limitations in representing the complex and heterogeneous spatial structures present in the geological formations. Multiple-point statistics (MPS) algorithms can reproduce the connectivity and complex features by incorporating higher-order statistics through a multiple-point framework (Guardiano and Srivastava, 1993). In the MPS methods, the statistics of data structures are extracted from a training image (TI), which is an example of the spatial structures to be reproduced. The training image provides useful information to include physical reality while stochastic modeling (Strebelle, 2012; Mariethoz and Caers, 2015).

The MPS algorithms can be categorized into two distinct classes: the pattern-based and the pixel-based simulation methods. Each of them has its own advantages and limitations. The applicability of MPS algorithms is measured tthrough (a) accurate reproduction of complex geological features; (b) honoring conditional data; and (c) the computational efficiency. Thus, selecting the best geostatistical method would be based on the balance between the three mentioned criteria (Mariethoz and Caers, 2015). The pattern-based methods simulate a group of points at a time, while pixel-based methods sequentially simulate every single point on the simulation grid, considering its surroundings. Pattern-based algorithms are recognized for better reproduction of large-scale features, but with biased simulation when conditioned to hard data (Tahmasebi, 2018). Although, pattern-based approaches (Efros and Freeman, 2001; Arpat, 2004; Arpat and Caers, 2004; Zhang et al., 2004 & 2006; Wu et al., 2008; Honarkhah and Caers, 2010; Chatterjee et al., 2012; Mahmud et al., 2014; Tahmasebi et al., 2014; Tahmasebi et al., 2012; Tahmasebi and Sahimi, 2015; Abdollahifard, 2016; Li et al., 2016; Parra and Ortiz, 2011; Rezaee et al., 2015) are drawing attention for being fast,their general drawback is the lack of variability due to the verbatim copy of large areas from training image to the simulation grid (Mustapha and Dimitrakopoulos, 2010 & 2011). Furthermore, the pattern-based methods suffer from inefficiency in handling hard data, and they are generally difficult to apply when dense conditioning data are available to the model, particularly in mining.

Pixel-based MPS algorithms (Guardiano and Srivastava, 1993; Strebelle, 2002; Boucher, 2009; Mariethoz et al., 2010; Mustapha and Dimitrakopoulos, 2010 & 2011; Huysmans and Dassargues, 2011; Straubhaar et al. 2011; Abdollahifard and Faez, 2014; Mariethoz et al., 2015; Minniakhmetov et al., 2018; Yao et al., 2018; Gravey and Mariethoz, 2019) are generally considered to be computationally inexpensive while having limitations in reproducing connectivity of complex patterns. Their have advantages include the generation of more realistic simulations, no need for fusion of the patches, and flexibility in handling conditioning data (Gravey and Mariethoz, 2019). Guardiano and Srivastava (1993) introduced ENESIM as the first pixel-based MPS algorithm, in which the conditional distribution of simulation node is estimated based on all matches searched through the TI from which a value is sampled. Their approach is computationally expensive. Strebelle (2002) introduced a faster version, named SNESIM, by developing an approach using in advance storage of conditional probabilities in a tree structure. SNESIM also uses the concept of multi-grid for MPS simulation to improve the connectivity of complex structures. Straubhaar et al. (2011) introduced another extension, named IMPALA, by proposing a reduction in memory usage where the information is stored in lists instead of trees. Lists are sequences of data points, thus, visiting their nodes takes a longer time compared to trees. Straubhaar et al. (2013) pursued the same goal by using list and tree structures. For the same purpose, Direct Sampling (DS) method (Mariethoz et al., 2010), samples directly from the TI instead of a conditional probability distribution to decrease the memory requirement. Mustapha and Dimitrakopoulos (2010 & 2011) introduced the high-order simulation (HOSIM) algorithm for the simulation of complex nonlinear non-Gaussian systems. Their sequential simulation method works based on high-order spatial connectivity criteria, named as spatial cumulants. The local conditional distributions in the HOSIM algorithm are generated using high-order Legendre polynomials with the coefficients calculated from the cumulants population. Yao et al. (2018) provided an extension of the method, where the numerical approximation of the conditional probability density function (*cpdf*) was calculated from the spatial Legendre moments. To decrease the computational expense of their method, they limited the *cpdf* approximation to the computation of a unified empirical function within a local neighborhood. Minniakhmetov et al. (2018) developed another extension, where the *cpdf* is approximated using Legendre-like orthogonal splines, and the coefficients of spline approximation are derived from high order spatial statistics inferred from hard data and training image. However, with so much effort, the pixel-based algorithms are still computationally intensive.

To improve the performance of the pixel-based MPS algorithms, researchers have followed several strategies. One of the important aspects of a multiple-point geostatistical model is to encounter high-dimensional data, which, especially in pixel-based models, is a limiting criterion. In the past, some algorithms have been proposed to reduce the dimensionality of the MPS patterns database (Zhang et al., 2006; Wu et al., 2008; Honarkhah and Caers, 2010; Chatterjee et al., 2012). Zhang et al. (2006) used filter scores to reduce the dimensionality of the patterns database. Their selected filter statistics are specific linear combinations of each pattern's pixel values that represent directional mean, gradient, and curvature properties. Their proposed algorithm (FILTERSIM) reduces the dimensionality to 6 and 9 for two- and three-dimensional simulations, respectively. Wu et al. (2008) developed an extension of FILTERSIM by replacing the pixel-wise similarity detection between patterns and comparingtheir filter scores as their representatives. However, using only a few filter scores is not always possible to capture the complexity present in the available patterns, resulting in possible similar filter scores for different patterns. Chatterjee et al. (2012) extracted wavelet approximate sub-band coefficients of the patterns as their low-dimensional representation. Although wavelet decomposition is a computationally demanding technique, their WAVESIM method provides faster implementation and realizations that better represent the training image structures compared to FILTERSIM. The WAVESIM algorithm can also be sensitive to the number of the extracted wavelet coefficients.If the number of coefficients are high for a specific training image, the dimensionality may still remain high. Honarkhah and Caers (2010) used a multi-dimensional scaling (MDS) algorithm to map database of patterns on a two-dimensional Cartesian space for subsequent clustering. However, MDS is a pairwise distance-based technique with a slow computational performance on cases with a high number of patterns. In addition, it works only on local pairs of data. When high-dimensional data are located on a low-dimensional nonlinear manifold, the job of keeping very similar points near each other is very difficult with a linear method like MDS (Mead, 1992; van der Maaten and Hinton, 2008).

Another major step in decreasing the pixel-based methods' computational expense, implemented in some MPS models, is the clustering of patterns database extracted from the training image (Chatterjee et al., 2012a & 2012b; Honarkhah and Caers, 2010; Mustapha et al., 2014; Li and Aguilera, 2018). There are numerous clustering techniques provided by computer and data scientists, including hierarchical clustering, expectation-maximization, fuzzy c-means, and mean-shift clustering (Jain, 2010). Still, one of the most famous and widely used clustering methods in MPS methods is the k-means clustering (Zhang et al., 2004 & 2006; Wu et al., 2008; Honarkhah and Caers, 2010; Chatterjee et al., 2012-a & 2012-b), which is an unsupervised learning algorithm (MacQueen, 1967). It is a distance-based clustering technique that has the advantages of non-complexity and fast computation. However, the main drawback of this algorithm, especially in geostatistical simulations, is the fact that it requires the number of clusters as an input to run the method on data. This information is not known apriori, and there is no reliable way to determine that. Thus, this algorithm for clustering patterns requires performing a sensitivity analysis of the cluster number to achieve the best results. The other flaw of the k-means clustering method is its inability to detect dense areas of data, as it works on the distance between the points that can lead to wrong cluster detections by the algorithm in some cases and the efficiency of the method can be questioned.

On the implementation side of the MPS methods, there are some key parameters that users need to select. The success of the MPS methods is highly dependent on the parameter selection. In the past, a number of optimization methods have been proposed for tuning the parameters of MPS algorithms. Yang et al. (2016) introduced an optimization-based MPS method by applying the Expectation-Maximization algorithm. Melnikova et al. (2015) proposed an algorithm to compare the training image and realizations of MPS in a quantitative manner. Dagasan et al. (2018) employed this methodology for the parameter tuning of the DS algorithm (Mariethoz et al. 2010; Abdollahifard and Faez, 2014), using Jensen-Shannon Divergence as the objective function that is optimized by simulated annealing (SA) algorithm. Abollahifard et al. (2019) also introduced a quantitative MPS results evaluation method by estimating the coherence map using keypoint detection and matching.

In this study, we present a novel, computationally efficient, pixel-based MPS method, which can preserve the complexity and continuity of the training images in simulations while honoring the conditioning data in generating realizations. The proposed research uses t-Distributed Stochastic Neighbor Embedding (t-SNE) for dimensional reduction of pattern database, and subsequent clustering of the patterns using an unsupervised classification technique, named Density-based Clustering of Applications with Noise algorithm (DBSCAN). The t-SNE algorithm is implemented as an efficient dimensionality reduction technique to map and visualize database of patterns on a two-dimensional Cartesian environment based on their joint-probabilities. The main advantage of t-SNE, as a nonlinear method, over linear algorithms such as multi-dimensional scaling (MDS) and principal component analysis (PCA), is the ability to preserve global and local data structures at the same time (van der Maaten and Hinton, 2008). The DBSCAN algorithm maps the patterns database using the output from the t-SNE algorithm. It is specifically more efficient on low dimensional data and has the main advantages of discovering arbitrarily shaped clusters, robust outlier removal and no need for cluster number selection (Ester et al., 1996). These methods are applied to reduce memory requirement for storing the training image configurations, and for reduction of computations. In order to automize and to optimize parameter selection for the proposed method, we have implemented different optimization methods to minimize the user's inputs. The proposed methodology is validated and tested using different synthetic datasets and applied in a three-dimensional case-study gold mine for simulating orebody model.

2.2. Materials and Methods

In this section, we illustrate the methodology by explaining the general idea of the pixel-based MPS simulation using training images and the patterns database generation in the first subsection.Then, the details of the machine learning algorithms implemented in our methodology and mathematics behind them are provided. Then, we illustrate the pixel-based stochastic simulation process designed in our methodology in the fourth subsection, and the last subsection brings a summary of our proposed algorithm in a step-by-step manner. The flow chart of the proposed MPS method is also presented in Fig. 1.

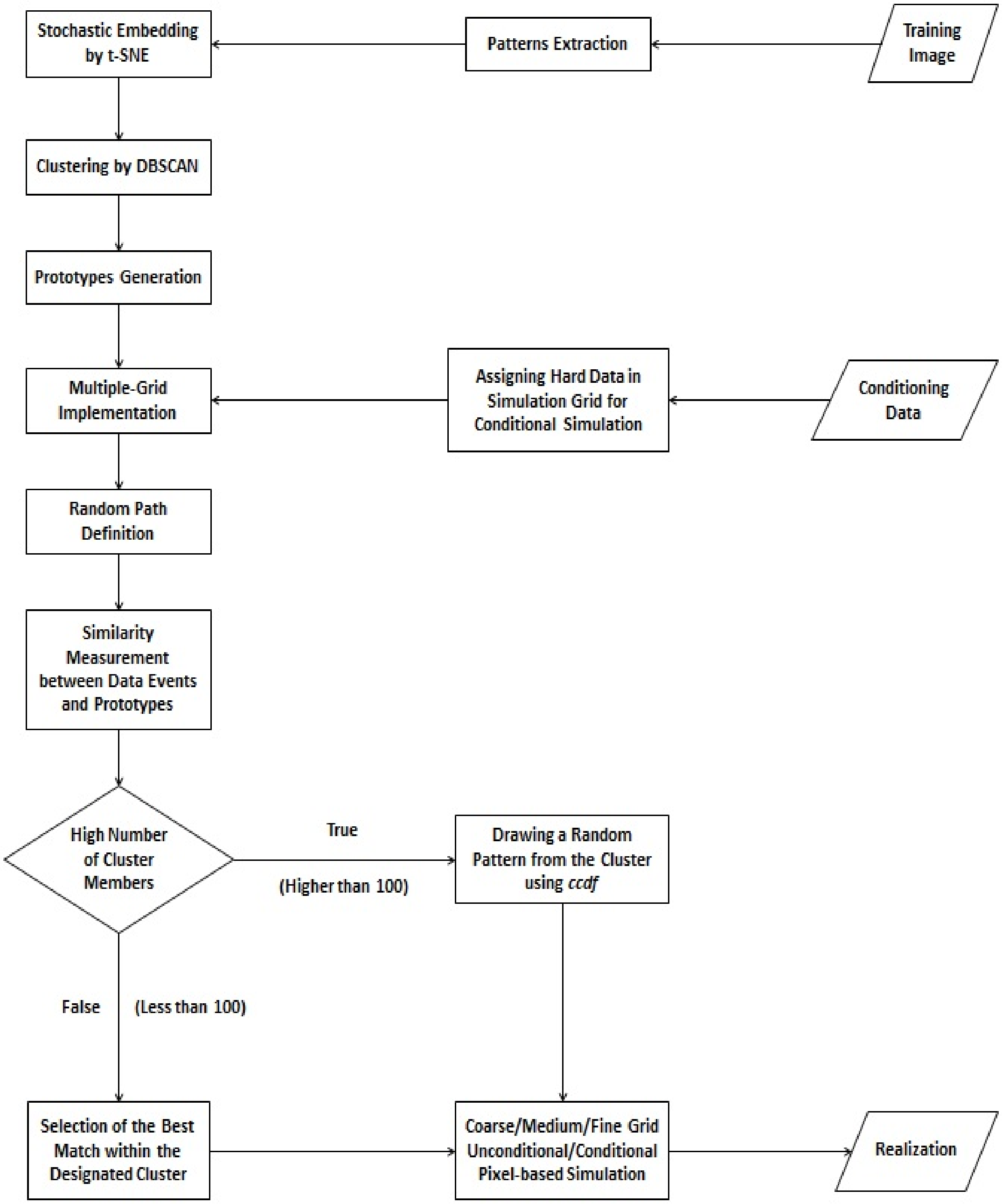
Stochastic Embedding by t-SNE
Patterns Extraction
Training Image
Clustering by DBSCAN
Prototypes Generation
Multiple-Grid Implementation
Assigning Hard Data in Simulation Grid for Conditional Simulation
Conditioning Data
Random Path Definition
Similarity Measurement between Data Events and Prototypes
High Number of Cluster Members
True
(Higher than 100)
Drawing a Random Pattern from the Cluster using *ccdf*
False
(Less than 100)
Selection of the Best Match within the Designated Cluster
Coarse/Medium/Fine Grid Unconditional/Conditional Pixel-based Simulation
Realization

Figure 1. Flow chart of the proposed pixel-based MPS method.

2.2.1. Patterns Extraction and Multiple-Point Statistics

The training image is the main source of information to borrow multiple-point statistics in the MPS modeling. To borrow the multiple-point information from a training image, the patterns are extracted and stored in a pattern database. The selection of the template size to extract patterns is the key decision for extracting multiple-point information. However, it would be a crucial part of the algorithm, as this template size can significantly impact the reconstruction process, and the quality of realizations depends on this size (Journel, 2003).The sensitivity analysis to template size has been the main part of many studies (Honarkhah and Caers, 2010). The template size should be as small as possible to reduce the computational time, and as large as necessary to represent the features of a specific training image. In this study, we use the method proposed by Honarkhah and Caers (2010), which uses the concept of entropy to determine the size of the template for scanning the training image. The algorithm for optimal template selection starts by scanning through the training image with different template sizes. The maximum log-likelihood profile is used to automatically detect the optimal value of the template size.

Define $\mathbf{TI}(u_i)$ as a value of the training image $\mathbf{TI}$ where $u \in \mathbf{G}_{\mathrm{TI}}$, and $\mathbf{G}_{\mathrm{TI}}$ is the regular Cartesian grid discretizing the training image, and $\mathbf{d_i}$ indicates a specific multiple-point vector of $\mathbf{TI}(u_i)$ within a template $\mathbf{t}$ centered at node u_i, that is:

$$\mathbf{d_i} = \{\mathbf{TI}(u_i + \mathbf{h}_1), \mathbf{TI}(u_i + \mathbf{h}_2), \dots, \mathbf{TI}(u_i + \mathbf{h}_{n\times n})\} \tag{1}$$

where, $\mathbf{h}_{\mathrm{i}}$ is the indicator of the vector of pixel addresses around the central node, u_{i}, using template size of $\mathbf{t}=\{n \times n\}$.

After extracting the pattern vector $\{\mathbf{d_i}; \mathrm{i} \in 1, \dots, m\}$, where m is the number of extracted patterns from the training image using the optimal template size $\mathbf{t}$, we store them in a patterns database, $\mathbf{D}_{\mathrm{m}\times\mathrm{nn}}$, which includes m number of rows each of whichrepresents a unique pattern. The middle node of each multiple-point vector will be the central node (u_i) of the extracted pattern.

The multiple-point statistics are expressed as the cumulative density functions (*ccdf*) for the random variable Z(u_i) conditioned to local data events $\mathbf{E}_i$ = {Z(u_i+$\mathbf{h}_1$), Z(u_i+$\mathbf{h}_2$), …, Z(u_i+$\mathbf{h}_{nn}$)} on the simulation grid (SG):

$$\mathrm{F}(\mathrm{z}, u_\mathrm{i}, \mathbf{E}_\mathrm{i}) = \mathrm{Prob}\{\mathrm{Z}(u_\mathrm{i}) \leq \mathrm{z}|\mathbf{E}_\mathrm{i}\} \tag{2}$$

Simulations based on multiple-point statistics proceed sequentially. At each successive location, the conditional cumulative distribution function (*ccdf*) of F(z, u_i , $\mathbf{E}_i$) is conditioned to both the previously simulated nodes and the actual data. A value for Z(u_i) is drawn from the *ccfd*, and the algorithm proceeds to the next location. Since F(z, u_i, $\mathbf{E}_i$) depends on the respective values and relative positions of all the neighbors of u_i simultaneously, it is very rich in terms of information content (Mariethoz et al., 2010).

However, in this study, the calculation and usage of *ccdf* is limited to the visited clusters of patterns with a high number of members. The reason behind this is the less computational expense of the pixel-based simulation because there is a possibility for the algorithm to search among many patterns within a populous cluster to simulate a single node. Instead of using *ccdf* for all clusters, we perform the second step of similarity measurement within a cluster, which has a limited number of patterns, and therefore,the *ccdf* might be unstable.

2.2.2. Dimensionality Reduction and Mapping by t-SNE

Generally, the patterns database, $\mathbf{D}_{m\times nn}$, can be a very high dimensional matrix, and the clustering of patterns can be considered a difficult task for such a large dataset of patterns. In order to solve such a problem, we apply the t-Distributed Stochastic Neighbor Embedding (t-SNE) algorithm. However, for the sake of optimization of the algorithm, and ease of computation, it is recommended to apply the Principal Component Analysis (PCA) algorithm (Hotelling, 1933; Jolliffe, 2002) to the data when the number of features (template size) is relatively large, to reduce the dimensionality to a reasonable number, e.g., 50 (van der Maaten and Hinton, 2008; van der Maaten, 2009). This will suppress some noise and speed up the computation of pairwise distances between samples. The PCA algorithm works with eigenvalues to investigate the variance in the dataset to select the features that are responsible for the highest variance. The number of principal components is as large as the original datasets, as each component is built by projecting all of the other observations on its axis. However, the first few principal components constitute most of the variance among all (Jolliffe, 2002).

After decreasing the number of dimensions to a reasonable size (p), we go through a two-dimensional stochastic mapping. In our method, t-SNE models each high-dimensional multiple-point pattern vector, $\mathbf{d_i}$, by a two-dimensional point $\mathbf{y}_i$ in such a way that similar patterns are modeled by nearby points, and dissimilar patterns are modeled by distant points with high probability. The algorithm firstly constructs a probability distribution, $\boldsymbol{P_i}$, over pairs of high-dimensional objects. Then, t-SNE defines a similar probability distribution over the points in the low-dimensional map, and it minimizes the Kullback–Leibler divergence between the two distributions with respect to the locations of the points in the map. The algorithm takes the following steps to embed the data in low dimensions (van der Maaten and Hinton, 2008; van der Maaten, 2009; van der Maaten, 2014):

1. The standardization of the patterns database to create $\breve{\boldsymbol{D}}_{m\times nn}$ is performed by subtracting $\mathbf{D}_{m\times nn}$ by the mean vector $\{\boldsymbol{m_j}; \mathrm{j} \in 1, \dots, \mathrm{nn}\}$, and dividing by the standard deviation vector $\{\boldsymbol{\sigma}_\mathbf{j}; \mathrm{j} \in 1, \dots, \mathrm{nn}\}$.

2. The standardized pattern database, $\breve{\boldsymbol{D}}_{m\times nn,}$ is mapped by Principal Component Analysis (PCA), and first p PCs for each row of the dataset in $\breve{\boldsymbol{D}}_{m\times nn}$, and creating the $\widetilde{\boldsymbol{D}}_{m\times p}$, for cases where *nn* is higher than p as a reasonable dimensionality before mapping by t-SNE.

3. The pairwise Euclidean distances (*dist*) between each row vector $\widetilde{\boldsymbol{d}}_{\boldsymbol{i}}\boldsymbol{,}$ $\mathrm{i} \in 1, \dots, m$, in $\widetilde{\boldsymbol{D}}_{m\times p}$ were calculated.

4. The standard deviation σ_i for each high-dimensional point $\widetilde{\boldsymbol{d}}_i$ were calculated, so that the perplexity of each point is at a predetermined level. Perplexity is the effective number of local neighbors of each point. Typical values for perplexity are proposed

to be between 5 and 50, and we have used the suggested trade-off value of 30. The performance of t-SNE is fairly robust under different settings of the perplexity (van der Maaten, 2009).

5. The conditional probability of each point $\widetilde{\boldsymbol{d}}_j$ given $\widetilde{\boldsymbol{d}}_i$ was defined via Eq. 3:

$$P_{j|i} = \frac{\exp(\frac{-dist\left(\tilde{d}_i,\tilde{d}_j\right)^2}{2\,\sigma_i^2})}{\sum_{k\neq i}\exp(\frac{-dist\left(\tilde{d}_i,\tilde{d}_k\right)^2}{2\,\sigma_i^2})} \qquad (3)$$

   Conditional probability $P_{j|i}$ is the measure of similarity between data points $\widetilde{\boldsymbol{d}}_i$ and $\widetilde{\boldsymbol{d}}_j$, and they would be considered neighbors based on their probability density under a Gaussian centered at $\widetilde{\boldsymbol{d}}_i$. Thus, $P_{j|i}$ is higher for nearby data points, and almost negligible for very far points, when the variance of the Gaussian, σ_i, has a reasonable value (van der Maaten and Hinton, 2008).

6. The similarity matrix was calculated, which is the joint probability distribution of $\widetilde{\boldsymbol{D}}_{m\times p}$ using Eq. 4:

$$\mathbf{P}_{ij} = \frac{P_{i|j}+P_{j|i}}{2m} \quad , \quad (\mathbf{P}_{ii} = 0) \qquad (4)$$

7. An initial set of low-dimensional points (in our case, two-dimensional), $\mathbf{Y}_{m\times 2,}$ as initialization of the embedding process was created for high-dimensional dataset $\widetilde{\boldsymbol{D}}_{m\times p}$.

8. The Kullback-Leibler divergence between the model Gaussian distribution of the vectors in $\widetilde{\boldsymbol{D}}_{m\times p}$ and a Student *t*-distribution of points $\mathbf{Y}_{m\times 2}$ in the low-dimensional space was maximized. The probability model q_{ij} of the distribution of the distances between points $\mathbf{y}_i$ and $\mathbf{y}_j$ is provided via:

$$q_{ij} = \frac{\left(1+(||\mathbf{y}_i-\mathbf{y}_j||)^2\right)^{-1}}{\sum_k\sum_{l\neq k}(1+(||\mathbf{y}_k-\mathbf{y}_l||)^2)^{-1}} \quad , \quad (q_{ii} = 0) \qquad (5)$$

9. The Kullback-Leibler divergence between the joint-distribution **P** and **Q** will be the following term, which needs to be minimized:

$$\mathbf{KL}(\mathbf{P}||\mathbf{Q}) = \sum_{j} \sum_{i \neq j} P_{ij} \ \log \frac{P_{ij}}{q_{ij}} \tag{6}$$

10. The low-dimensional points were iteratively updating to apply the Barnes-Hut algorithm as an approximate optimizer, which groups nearby points in the low-dimensional space, and performs an approximate gradient descent based on these groups. The idea is that the gradient is similar for nearby points, so the computations of decreasing the divergence between high- and low-dimensional distributions can be simplified. The minimization of the KL divergence leads the initial embedded data sets to the optimized $\mathbf{Y}_{m\times2}$ set.

The optimization procedure explained in step 10 is the most time-consuming part of the algorithm. The t-SNE optimization can also be implemented via the exact but expensive algorithm, which optimizes the Kullback-Leibler divergence of distributions between the original space and the embedded space. However, we perform the Barnes-Hut approximations that are more efficient to speed up and cut memory usage of the program. The idea is that the gradient is similar for nearby points so that the computations can be simplified (van der Maaten and Hinton, 2008; van der Maaten, 2013).

2.2.3. DBSCAN Clustering and Prototypes Generation

DBSCAN clustering works based on the density of the pattern database, as opposed to distance. The advantage of DBSCAN is that, unlike k-means clustering, it does not require the cluster number as an input parameter to perform clustering. DBSCAN is a non-parametric density-based clustering algorithm, which groups together points that are closely packed together in space (points with many nearby neighbors). Itmarkes points that lie alone in low-density regions as outliers, whose nearest neighbors are too far away (Ester et al., 1996, Schubert et al., 2017).

The algorithm requires two user-defined input parameters: ε (EPS) as the neighborhood distance, and the minimum number of points required to form a cluster (MinPts). It is suggested by the algorithm developers that a value of MinPts should be selected between the number of dimensions, d, and $d\times 2$, which has been widely used (e.g., Sander et al., 1998). In our case of a two-dimensional environment, we selectthe MinPts=2, which generally results in a higher number of clustersand prototypes as a consequence. This will lead to a higher chance for better similarity detection results because the algorithm will be provided a higher number of prototypes for comparison to select as the best match during simulation of the nodes. Assigning MinPts equal to 3 for three-dimensional simulations can decrease the computational time by relatively decreasing the number of clusters. However, we prefer to use MinPts=2 for quality reconstructions in three-dimensional as well as two-dimensional simulations.

In our study, we use a method proposed by Akbari and Unland (2016) to define the EPS parameter. For each mapped point $\mathbf{y}_i$, firstly, the MinPts nearest neighbors using the k-nearest neighbor search algorithm (Friedman et al., 1977) is found. Then, the matrix of distances of nearest neighbors (**K**) is calculated using the Chebychev distance function. Then, the vector of distances of points to their second (MinPts=2) distant neighbors is extracted (**v**). The EPS value is then calculated by Eq. 7, using $mean$ and SD as the mean and standard deviation of the values in **v**.

$$\text{EPS} = (mean + (3 \times SD)) \qquad (7)$$

DBSCAN algorithm forms clusters within the data in $\mathbf{Y}_{m\times 2}$ by performing the following steps (Ester et al. 1996):

1. Selection of the first unlabeled (unvisited) observation $\mathbf{y}_i$ as the arbitrary starting point (current point) from the embedded input data set, $\mathbf{Y}_{m\times 2}$, and assigning the first cluster label, l, to 1.
2. Retrieval of the set of points within the EPS neighborhood of the current point. If the number of neighbors is less than MinPts, then labeling the current point as an outlier point (noise) by assigning l=0. Otherwise, label the current point as a core point belonging to the existing cluster l=1. The noise point might later be found in a sufficiently sized ε-environment of a different point and hence be made part of a cluster.

3. Iteration over each neighbor (new current point) and repeating step 2 until no new neighbors are found that can be labeled as belonging to the current cluster *l*. This process continues until the density-connected cluster is completely found.
4. Selection of the next unlabeled point in $\mathbf{Y}_{m\times2}$ as the current point, and increasing the cluster count by 1 to form new clusters or to detect a new outlier.
5. Repeating steps 2–4 until all points in $\mathbf{Y}_{m\times2}$ are labeled, and generation of the $\mathbf{idx}_{m\times1}$ vector of cluster numbers for all the points of $\mathbf{Y}_{m\times2}$.

After termination of clustering, we have a vector of numerical integers, $\mathbf{idx}_{m\times1}$, showing that each row of our data (each embedded pattern) corresponds to a cluster number, and this clustering index is valid for the raw pattern database $\mathbf{D}_{m\times nn}$.

2.2.4. Pixel-based Simulation

After clustering the patterns, we produce a prototype for every cluster by pixel-wise averaging of the cluster members; so that we have a single pattern as a reference for every class. During the sequential simulation, we measure the distance between the data event in the simulation grid to the prototypes, and we find the best match. Then, we go to the best match class to find the best pattern member in that class using a second distance function. If the number of patterns in a class is large, we use *ccdf* to draw a random pattern. This way, during sequential simulation, the MPS algorithm just searches for similarity among a limited number of prototypes and patterns, not among all available patterns within patterns database, $\mathbf{D}_{m\times nn}$. After performing clustering and prototypes generations, we merge the clusters with exactly similar prototypes in order to improve the computational performance during similarity search. Although rare, this can save some computational time, especially for categorical variables.

To optimize the simulation process by enhancing the quality of the realizations, we perform a multi-grid approach proposed by Arpat and Caers (2007). We use the multi-grid size of three to avoid high computational complexity by performing multi-level simulations, as the three-level grids approach is still capable of optimizing the simulation process in a more efficient way. The coarse-medium-fine grids configuration of our simulation method is depicted in Fig. 2, in which a fine grid simulation example is provided as well.

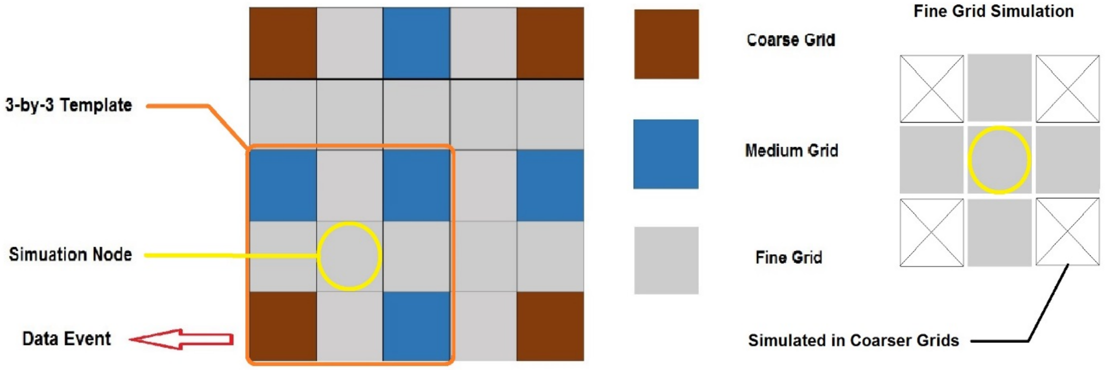

Figure 2. Three-level Coarse/Medium/Fine Multiple-Grid approach implemented.

Consider **SG** as the simulation grid, in which all the pixels (nodes) are filled with a default value of, e.g., -999 as the indication of being empty (not yet simulated). In the case of conditional simulation, the hard data locations is filled by the sample values in the coarse to fine SGs. The simulation grid is sequentially filled based on a random path using the pixel-based approach, where only the central node of the matched pattern isassigned to the simulation node. The coarser grids are simulated before simulating the finer grids. This process is continued until visiting all nodes within the simulation grid, **SG**.

We use a weight distribution in our methodology while measuring similarity in template matching. As our method is a pixel-based simulation method, there is a risk of mismatching due to the number of nodes that take part in similarity detection steps. It is likely that without assigning higher weights to the nodes near the simulation node, the selected pattern is not the best match for the data event. Thus, as shown in Fig. 3, we implement a weight distribution while measuring distances. Wee have chosen first order Minkowski distance (Manhattan distance), so the weights have their appropriate impact in the template matching process. The distance function is given by

$$d = \sum_{i=1}^{nn}(w_i \times |x_i - v_i|)^p \quad ; \quad (p = 1) \tag{8}$$

where, w_i is the weight associated with the x_i and v_i, which are corresponding nodes of the data event and patterns/prototypes database, respectively, and nn is the length of the multiple-point vectors of data events and patterns.

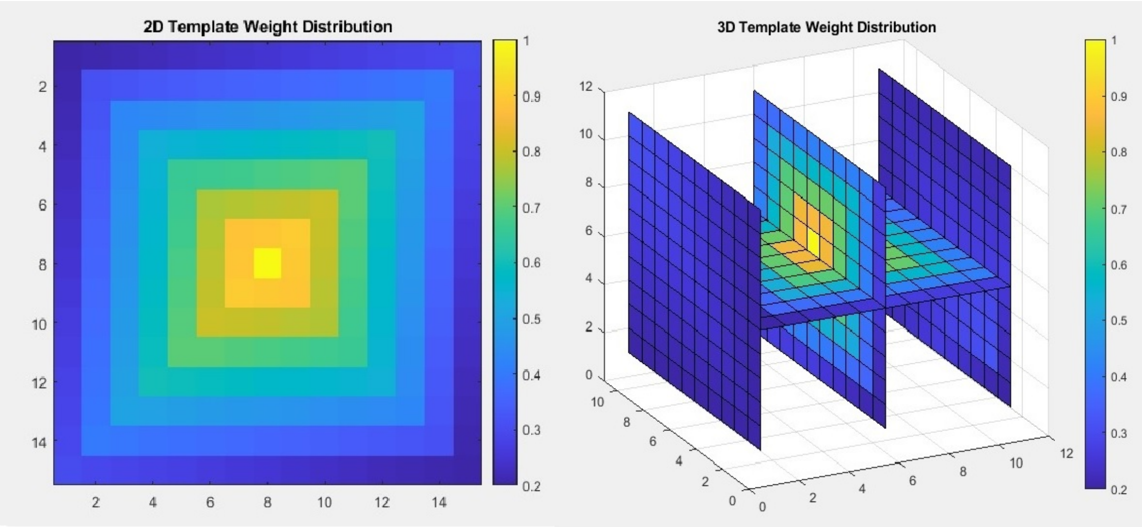

Figure 3. Plots of the weights distributions used while two- and three-dimensional template matching.

The similarity comparison step is expensive when the template size is big, and all or a high number of corresponding nodes around the simulation node are known. This problem is intensified for three-dimensional simulations, as the computational complexity is higher. We do a second-step similarity detection within the clusters with $c <= 100$ members to find the best match among the members. For the sake of increasing the speed of the simulation, we apply the *ccdf* function to draw random samples from the distribution only for highly populated clusters ($c \geq 100$). Figure 4 shows a sample of the *ccdf* plot of a cluster with three classes (Class 0, 1, and 2) available in the central node. By generating a random number between 0 and 1, the probability of the dominating class is higher to be selected. For instance, in the provided example plot, the probability of selection of class 2 is 14.75%. According to the example plot shown in Fig. 4, the generation of the random numbers of 0.15, 0.55, and 0.95 between 0 and 1 leads to the selection of classes of 0, 1, and 2, respectively. For continuous variables, however, the second step similarity detection isperformed by using the distance function to find the best match within the cluster in all cases.

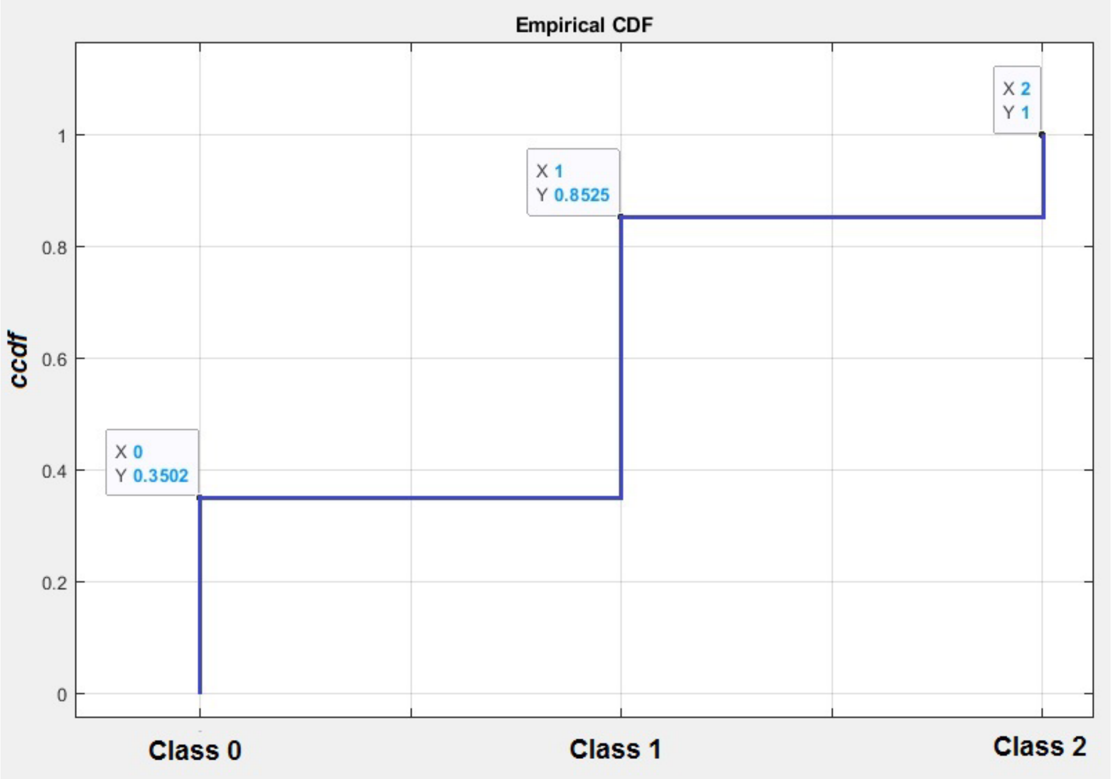

Figure 5. Cumulative conditional distribution function (*ccdf*) plot for selecting random pattern by drawing a random sample from the probability distribution for highly crowded clusters.

2.2.5. Summary of the Methodology

Here, we provide a summary of the proposed methodology:

1. Automatic determination of the template size **t** using an Entropy-based approach.
2. Scan the training image TI using automatically determined template size **t**, and extracting all the patterns.

3. Reduce the dimensionality of the pattern database by PCA algorithm, if needed, and subsequent stochastic mapping to two-dimensional by the t-SNE algorithm.
4. Clustering the patterns based on the two-dimensional map by DBSCAN algorithm.
5. Calculate the class prototype using the point-wise averaging of all patterns within a class.
6. In the case of conditional simulation, hard data will be assigned within the coarse to fine simulation grids, and the nodes will be marked as seen (sampled) points.
7. Define a random path visiting once and only once all unseen nodes.
8. Use the same template **t** at each unseen location to extract the data event on SG.
9. Find the best match between class prototypes and data event in the simulation grid.
10. Sampling a value of the central node from the best match class using either second-stage distance function or *ccdf*.
11. Assign the sampled value to the current simulating point.
12. Continue until all grid points are filled with simulated value.

Repeat steps 7 to 12 of the simulation process to generate different equiprobable realizations.

2.3. Validation Results

In this section, we demonstrate the performance of our model by visual and statistical comparison of the generated realizations with the continuous and categorical training images used in this study. In addition, the realizations produced by FILTERSIM algorithm are also provided for comparison of the simulation results. A total of four training images are used for validation and testing of the method, including one two-dimensional categorical TI, one two-dimensional continuous TI, one three-dimensional two-categorical TI, and a three-dimensional case-study of three-categorical data. The categorical (size=[101×101]) and continuous (size=[100×128]) two-dimensional TIs are shown in Fig. 5. The binary training image in Fig. 5-a (Honarkhah and Caers 2010; Strebelle 2000) represents a deposit with complex channels. For the simulation of continuous data, an exhaustive two-dimensional continuous horizontal slice (Fig. 5-b) is obtained from a three-dimensional fluvial reservoir of the Stanford V Reservoir Dataset (Mao and Journel, 1999), and the channel configurations and orientation are complex in nature form one slice to another in the vertical direction.

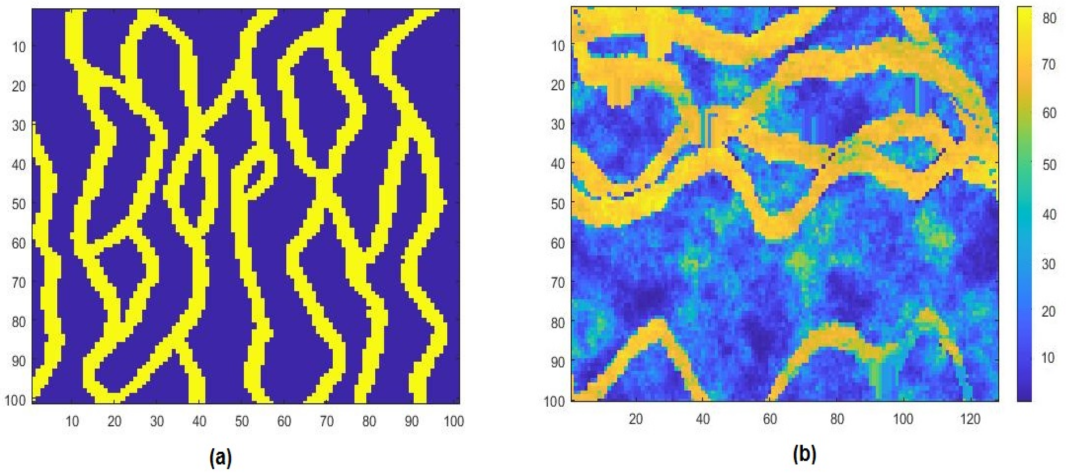

Figure 5. Categorical binary TI named channel (a), and continuous TI (b).

We performed the automatic template size determination on our training images, based on the approach explained in Section 2.2.1. The mean Entropy, the variance Entropy, and the Log-likelihood profile of the Entropy plots of different template sizes for training images of Fig. 5 are shown in Fig. 6. The optimal value is calculated based on defining a threshold on the slope change of the curve, which detects the point of which the curve is going to flatten. From the Entropy plots, we achieved the optimal template size (**t**) for the categorical channel TI, **t**=[15 15], and the continuous TI shown in Fig. 5-b, **t**=[25 25].

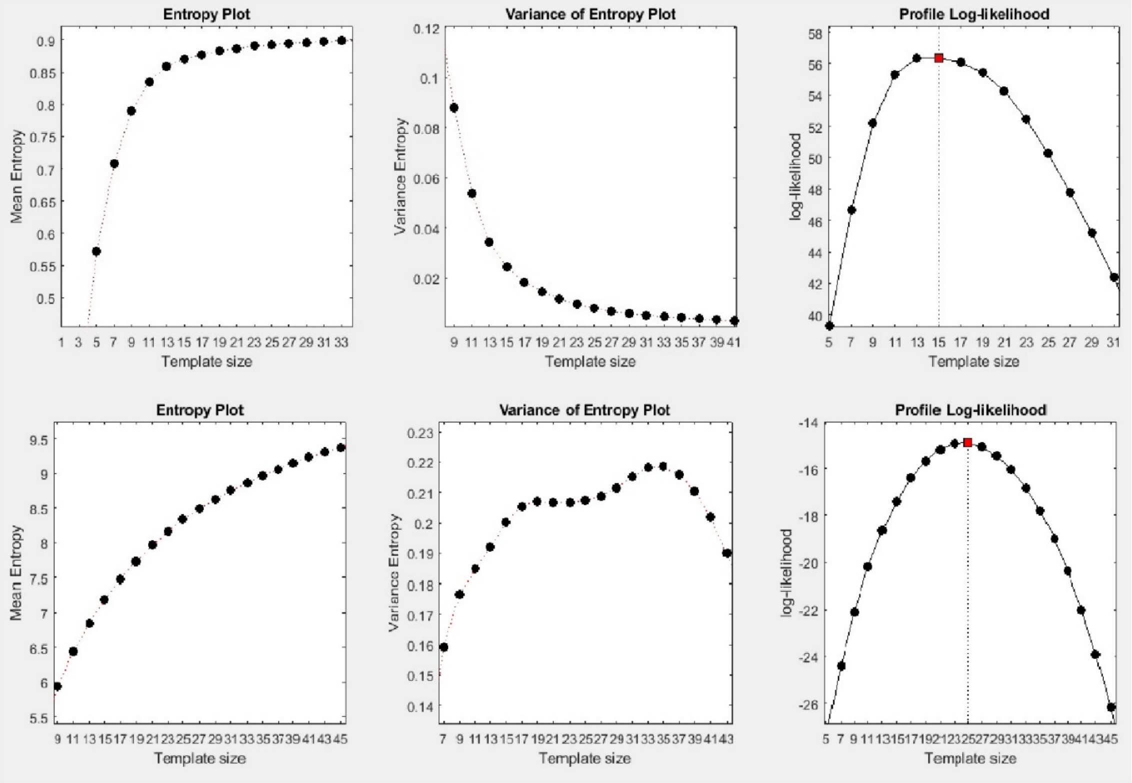

Figure 6. Entropy-based template size determination plots for the channel TI (top row) and continuous TI (bottom row) shown in Fig. 5.

The stochastic embedding and clustering of pattern database, generated using the automatically selected template size, were performed. In a single run, the algorithm provided 206 and 1004 distinct clusters for patterns database of the categorical and continuous TIs, respectively. As an example, a visualized plot of the clustering for the patterns database of the channel TI after mapping high-dimensional patterns database is shown in Fig. 7. As explained in Section 2.2.2, for better t-SNE performance, we should decrease the dimensionality by PCA when applicable. Thus, based on the suggested value by the t-SNE algorithm developers and performing experimentation by varying numbers between 50 and 100, we chose the value of 80 as a cut-off for PCA implementation. This leads to a more robust embedding performance. We decreased the dimensionality of the patterns to 80 in case of having a higher dimensionality. However, we did not observe a tangible impact in the performance of the t-SNE within the 50-100 range.

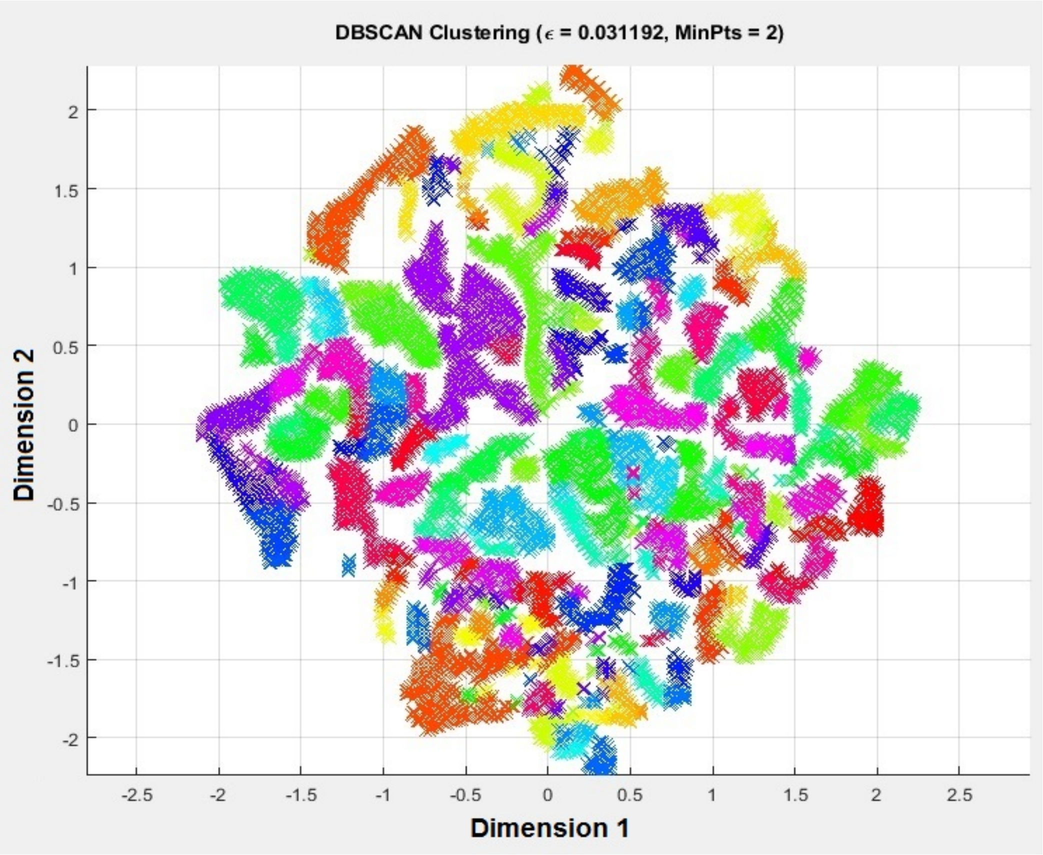

Figure 7. DBSCAN Clustering on the embedded patterns database, for the original-resolution channel TI shown in Fig. 5-a.

To verify the combined impact of t-SNE mapping and the DBSCAN clustering algorithm, we inspected the patterns from clusters to see the results visually. To provide an example, all patterns which were formed as a cluster for categorical and continuous training image by our algorithm are depicted in Fig. 8, and the generated prototype of the cluster is also shown for each TI. As t-SNE works based on probability (not distance), measuring the Euclidean distance in high-dimensional space and the two-dimensional space will not help to understand the goodness of fit in embedding. Due to the stochastic manner of the algorithm, it is a good idea to run the algorithm a few times and save the results when the KL divergence is the lowest. In our cases, the algorithm provided acceptably small values of KL divergence in all runs, and we didn't recognize the need to run the t-SNE multiple times.

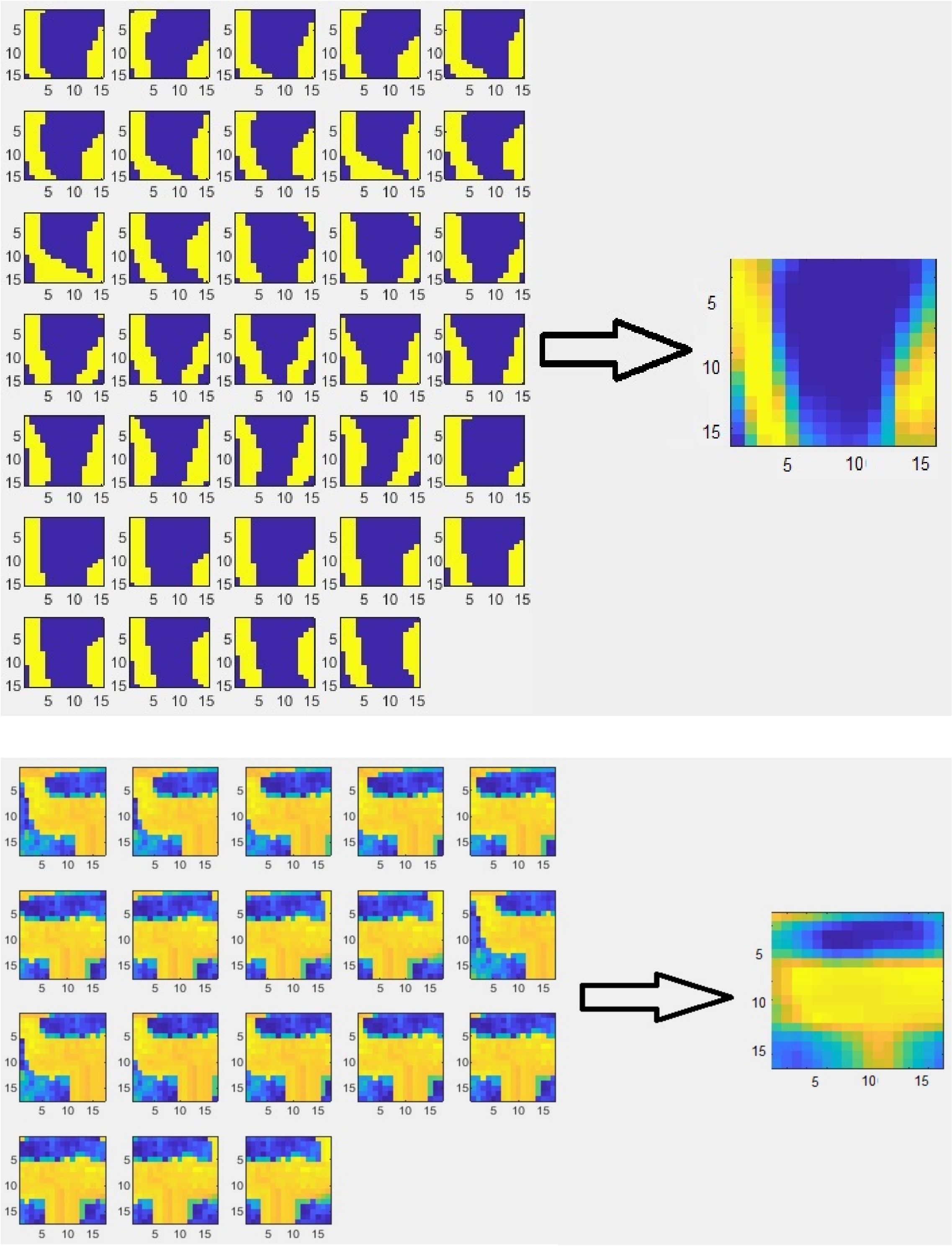

Figure 8. Members of randomly selected clusters (left) and their calculated prototypes (right), extracted from categorical channel (top) and continuous (bottom) training images.

2.3.1. Unconditional Simulation

The results of unconditional simulation on the introduced categorical and continuous training images are presented herein. Generally, a minimum number of realizations should be generated to have an understanding of the uncertainty of the simulation (Journel, 2003). We generated 100 realizations for each TI shown in Fig. 5, in order to understand the uncertainty of the modeling. Three generated realizations of the channel TI (Fig. 5-a) provided by our proposed method are depicted in Fig. 9. In addition, the realizations provided by the FILTERSIM algorithm using the same template size are provided for a visual comparison. We used 200 as the number of clusters to perform the FILTERSIM algorithm. Our proposed method outperforms in terms of reconstruction of the TI patterns and the continuity of the channels compared to the FILTERSIM method.

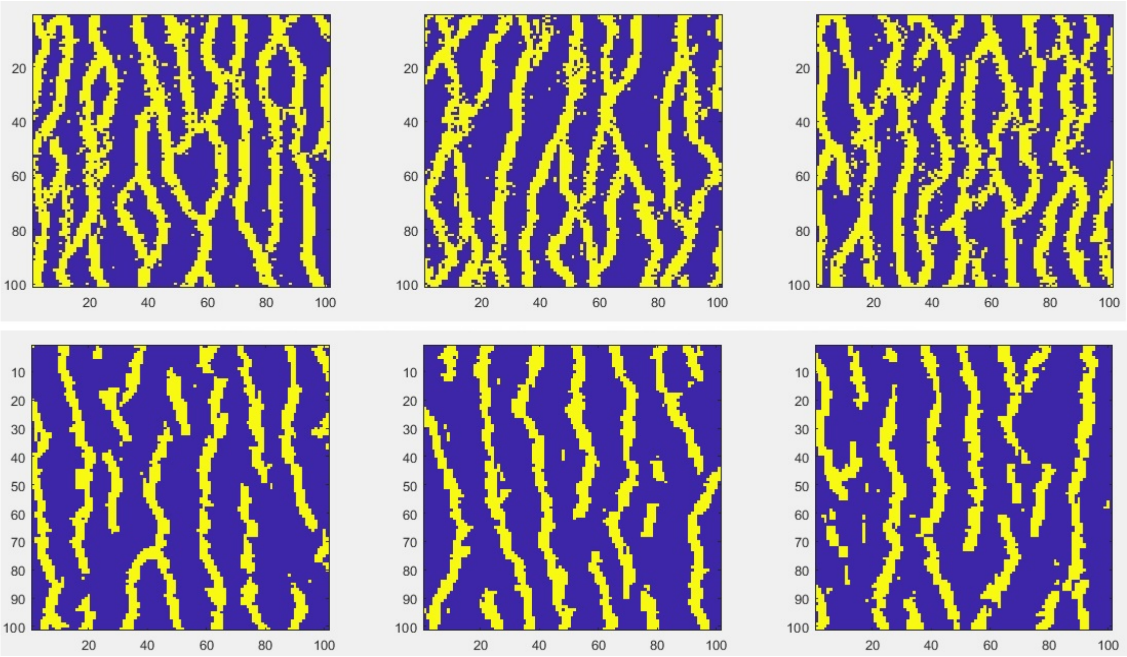

Figure 9. Categorical TI realizations of the proposed MPS method (top) versus FILTERSIM realizations (bottom).

It is observed that the continuity of the training image streams is reproduced well by our proposed method; however, FILTERSIM fails to maintain the continuity of the channels through the simulation process. The main differences of our proposed method to the FILTERSIM algorithm can be stated as two-dimensional mapping of the patterns database by t-SNE instead of using filter scores, and subsequent density-based clustering of the mapped data by DBSCAN, instead of using k-means clustering on the high-dimensional database. In addition, we use a pixel-based simulation approach, opposed to the pattern-based simulation implemented in the FILTERSIM method.

Statistical analysis was performed on the simulation results to show the efficiency of the methodology. Figure 10 shows a first order statistical comparison of the realizations and training images using the box and mean comparison plots. It can be inferred from the plots that the simulations tend to reproduce the one-point statistics of the training image, and there is almost no uncertainty regarding this criterion, as the mean plot shows no considerable deviations from the red line (TI average value) among the 100 realizations. Figure 11 shows variogram comparison as a measure of two-point statistics, the E-type (ensemble) and variance maps of the realizations. From the variogram comparison, a near-perfect match is observed between the two-points statistics of the training image and the 100 generated realizations.High quality in maintaining two-points statistical characteristics of the TI is achieved for all lag distances overall generated realizations. E-type and variance maps confirm that the stochastic nature of the simulation method is preserved in the unconditional simulation, as the simulations are not constrained by any hard data. The variability is almost high across the whole domain, and the channels are propagated in different locations within different realizations. Figure 12 depicts the third-order spatial cumulant maps (Mustapha and Dimitrakopoulos, 2010 & 2011) of the realization and training image for comparison as a higher-order validation method. From the cumulant maps, it can be inferred that for both small and large lag distances, the proposed algorithm is able to reproduce the third-order statistics of the channel TI well in unconditional categorical simulation.

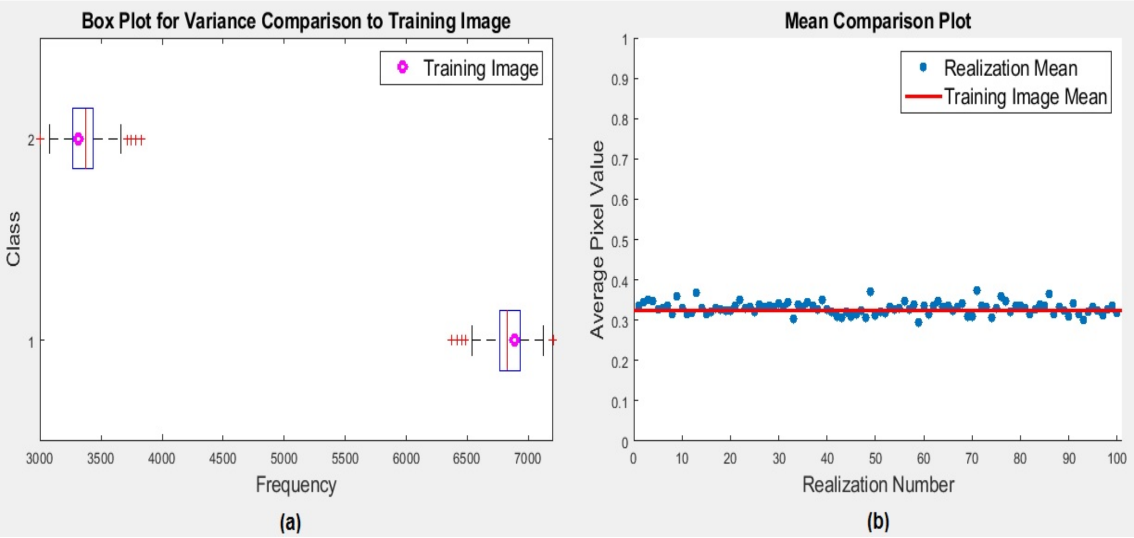

Figure 10. Categorical boxplots (a) and mean comparison plot (b) of generated 100 unconditional realizations simulated by categorical training image shown in Fig. (5-a).

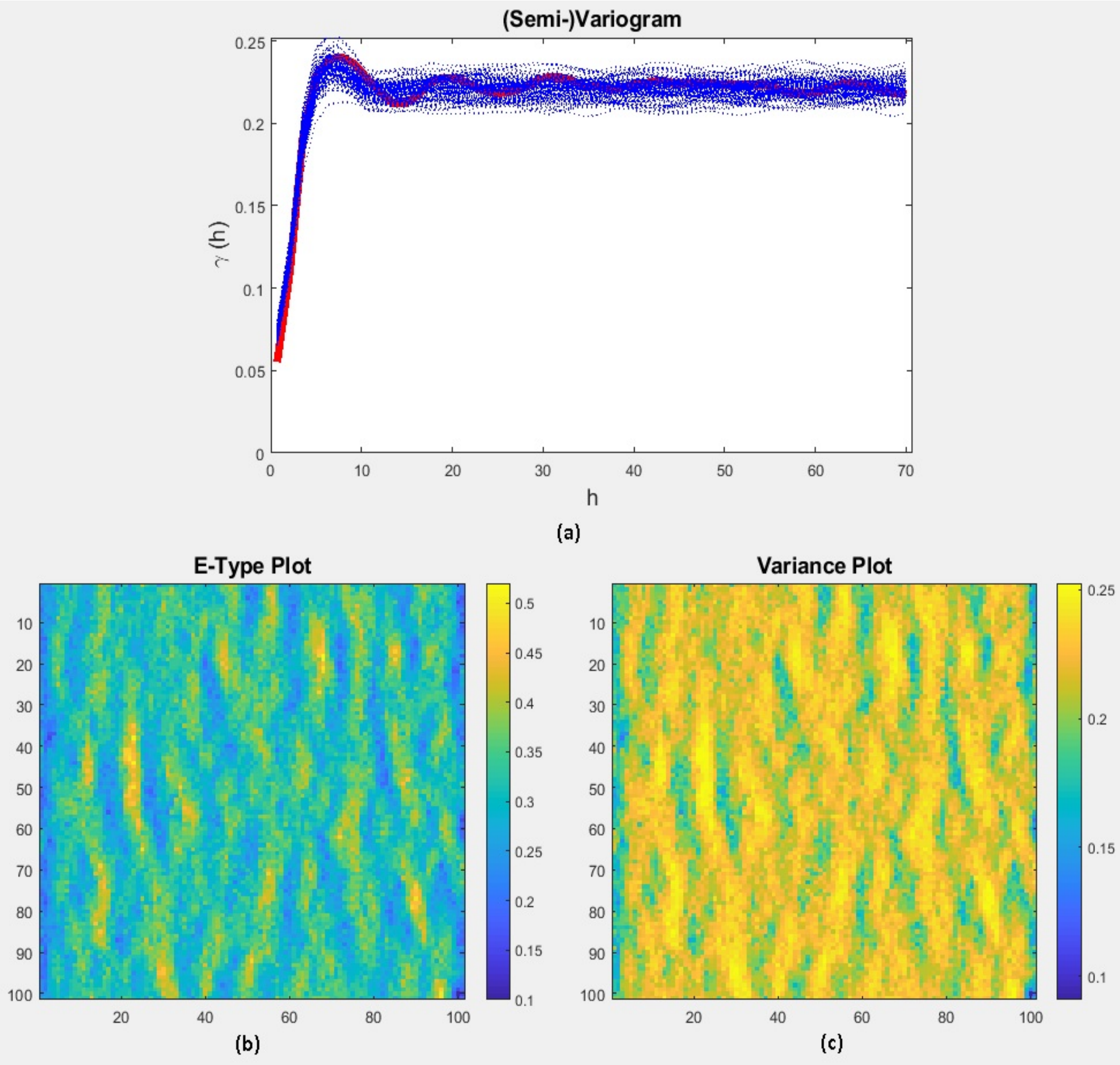

Figure 11. Variograms of the TI (in red) and100 unconditional realizations (a), and E-Type plots (b&c) of realizations of categorical channel training image shown in Fig. 5-a.

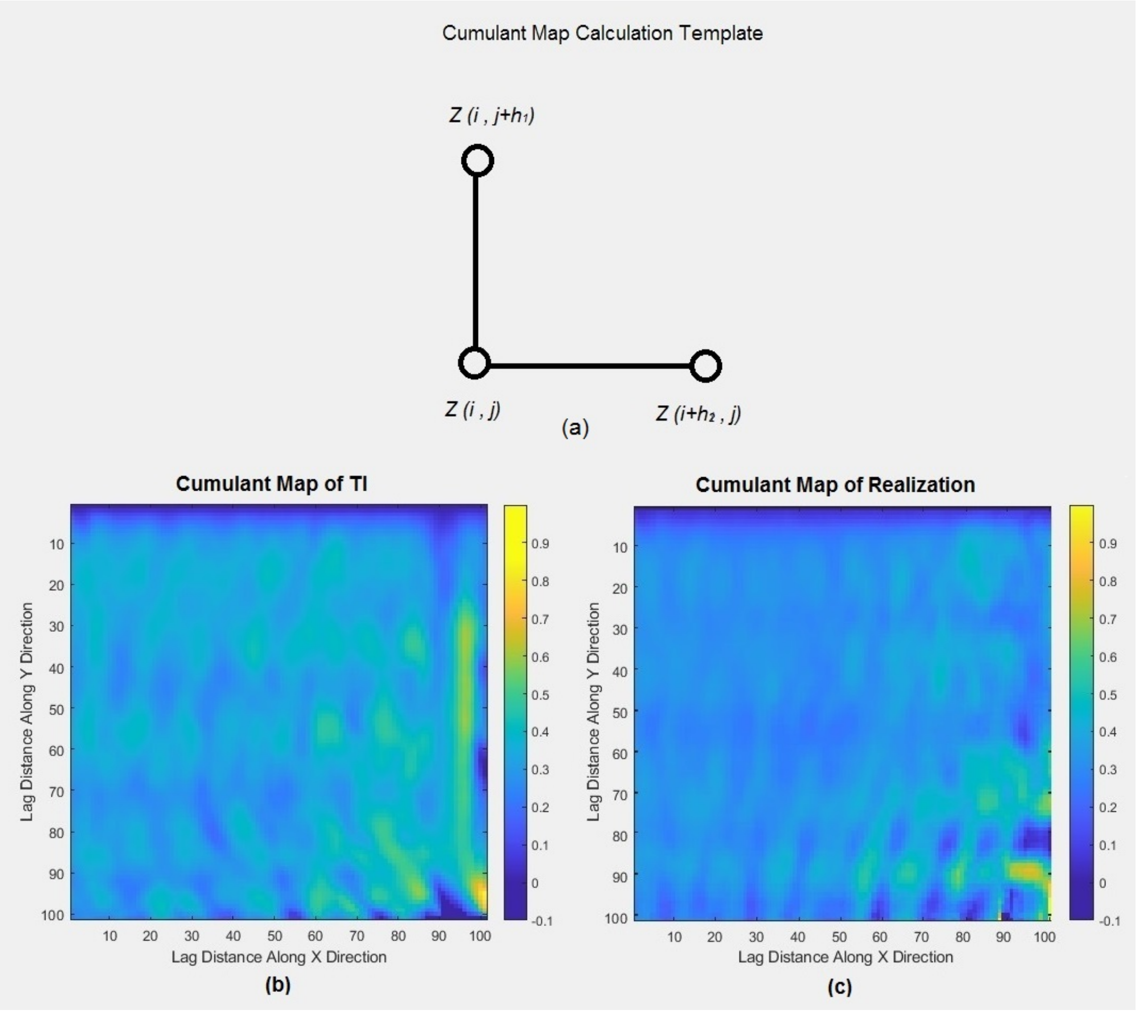

Figure 12. Cumulant maps of the categorical channel TI (b) and an unconditional simulation (c) plotted for higher-order validation of the proposed method.

The results of performing the methodology on continuous training image (Fig. 5-b) also confirm the applicability of the proposed simulation method for continuous variable. Figure 13 compares the quality of realizations with the FILTERSIM algorithm. It is visually observed that the continuity of high-valued streams (channels) is reproduced better by our method, in comparison with FILTERSIM. Figure 14 shows the histogram and variogram comparison plots for validation. The complex bimodal histograms of the generated realizations match that of the training image according to Fig. 14-a, which confirms the acceptable performance of the method in reproducing first-order statistics of the TI. The histogram of the TI (in red) falls in between those of the realizations, which means that realizations' average values are almost near to training image pixel values. There can be seen higher deviations from the TI average on the two modes, although the values between the heads of the histogram are matching better to the TI. The mean plot provided via Fig. 14-c proves that there is little uncertainty about the reproduction of the first-order statistics of the continuous TI, as none of the 100 realizations has the mean value far from the TI, shown via the red line. In addition, the realizations were successful in reproducing variograms similar to that of TI according to Fig. 14-b. The two-points statistics of realizations are maintained similar to the continuous TI over different lag distances, and deviations from the red line (TI variogram) are observed for only a few of the realizations' variograms mostly in larger lag distances. Figure 15 shows the E-type and variance maps calculated using 100 realizations. Figure 15 also presents cumulant maps of the TI and a realization From the cumulant maps, in can be inferred that the proposed algorithm is able to reproduce the third-order statistics of the continuous TI in unconditional simulation, especially in small lag distances. For larger lag distances, a reasonable match is also observed in most of the areas within the maps. The variability shown in variance plot is almost high across the whole domain, and high-value channels are propagated in different locations within different realizations according to the E-type plot, which shows the unconditional simulation is performing well in generating equiprobable reconstructions of the TI.

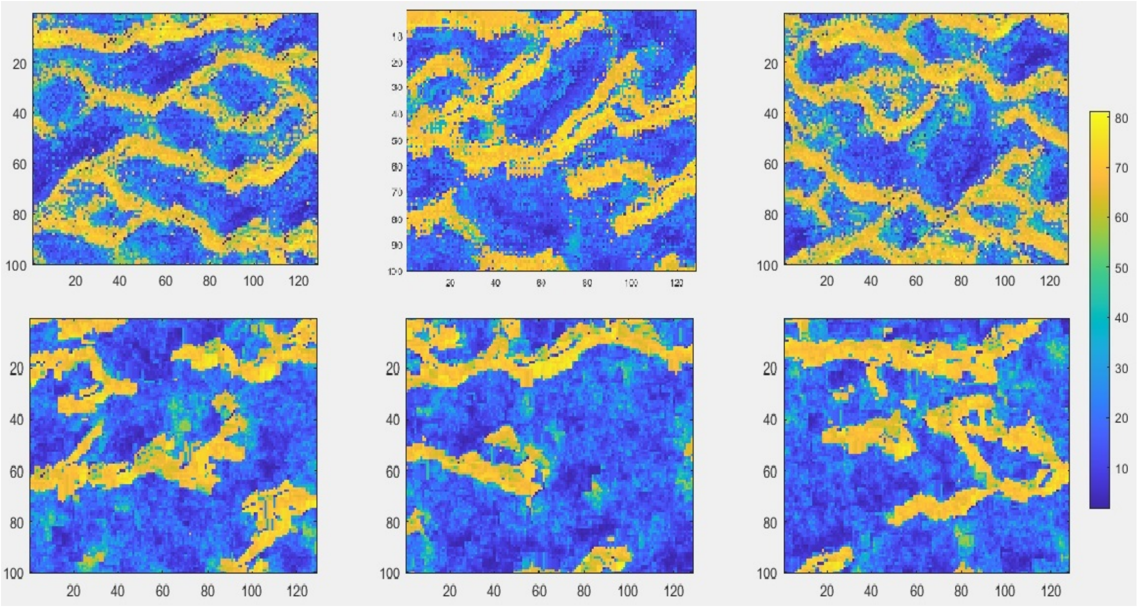

Figure 13. Continuous TI realizations of the proposed MPS method (top) versus FILTERSIM realizations (bottom).

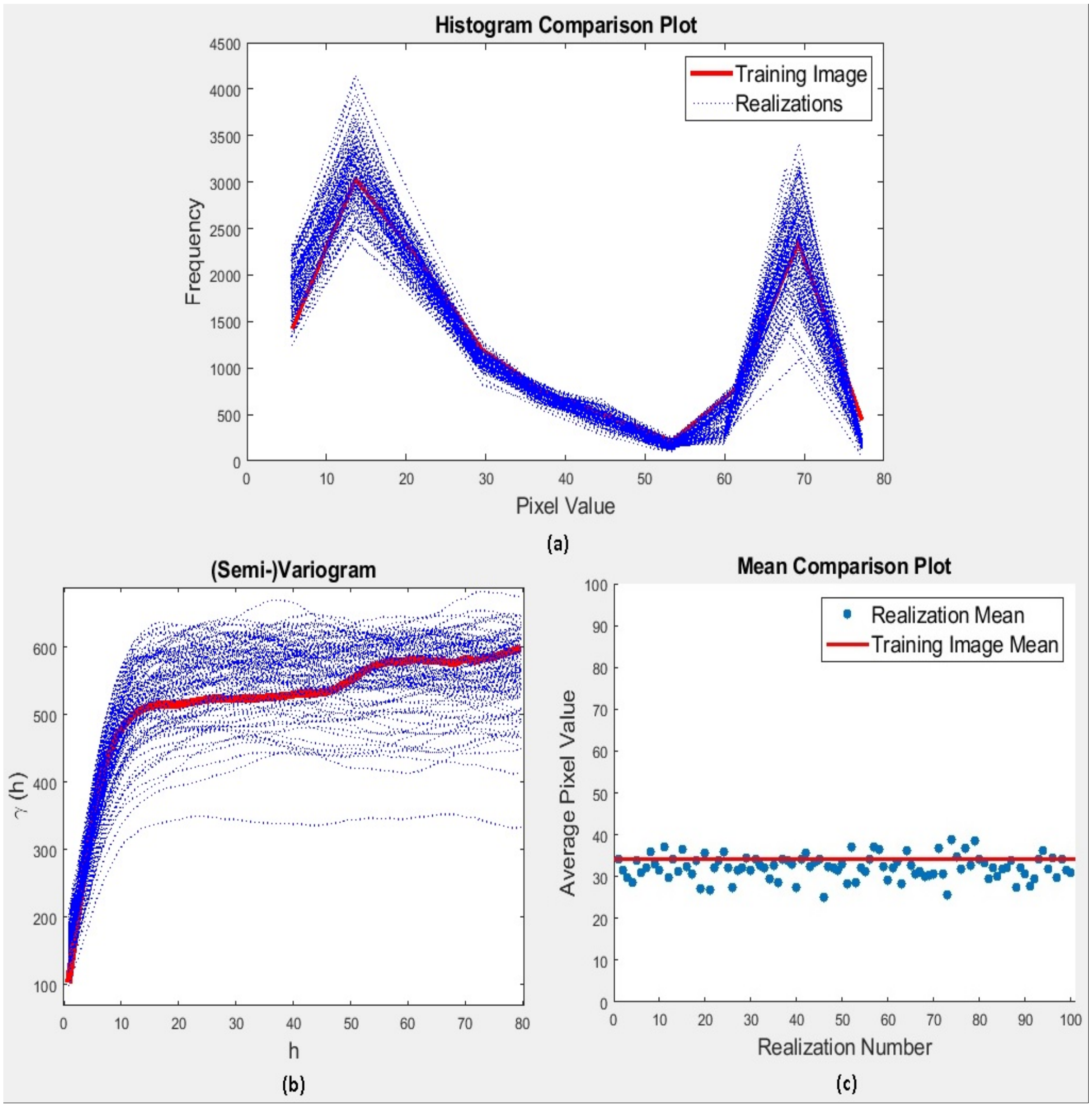

Figure 14. Histogram (a), variogram (b) and mean comparison (c) plots of the continuous training image shown in Fig. 5-b and the 100 generated unconditional realizations.

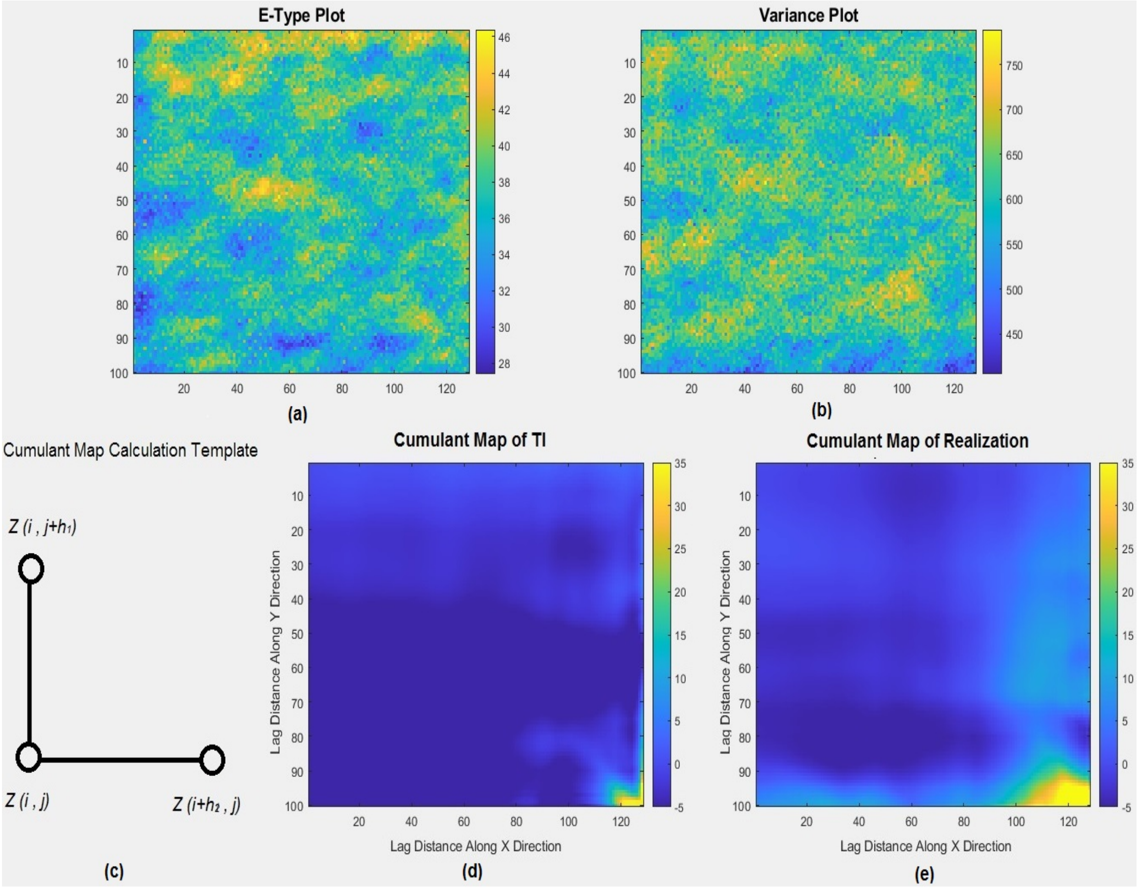

Figure 15. E-Type plots (a & b), and the cumulant maps of the continuous TI (d) and an unconditional simulation (e) for higher-order validation.

2.3.1.1. Three-Dimensional TI

Additionally, we tested our algorithm on a three-dimensional TI with a size of [69×69×39], shown in Fig. 16, accompanied by its cross-sectional views (middle row). We used a template size of **t**=[13×13×13], as concluded from the Entropy-based method for which the results are shown in Fig. 16 (bottom row). Figure 17 shows an unconditional simulation and its cross-sectional views to compare the training image with the realization and its structures. A realization is also depicted in Fig. 18, and compared to the realization by the FILTERSIM algorithm, for which the same template size was used. It can be seen that the method produced the desired shapes and continuities present in the three-dimensional TI with reasonable accuracy. Figures 19 and 20 show the comparison of the simulations and TI in terms of first- and higher-order statistics. The boxplots in Fig. 19 show that the median values for the frequencies of simulated nodes related to each category are near the corresponding values of the training image (red circle), and the TI class frequencies are within the 50% boxes of realizations' categorical frequencies. The mean plot also confirms the similarity of the first-order statistics of the TI (red line) and realizations by comparing their average values. From the cumulant maps, a good match for different lag distances from small to large is observable; thus, the proposed algorithm is able to reproduce the third-order statistics of the categorical three-dimensional TI well in unconditional simulation.

Looking at the realizations produced by the proposed method (Figs. 19 and 20), and comparing them to the TI (Fig. 16), it can be observed that continuity of the channels was reproduced within the simulation domain. The trapezoids available in the TI are very difficult to reproduce, even using pattern-based MPS simulation methods. The results show that those trapezoid-shaped structures are reproduced via unconditional simulations; however, as shown in Fig. 17, they are not precisely similar to the shapes available in the training image.

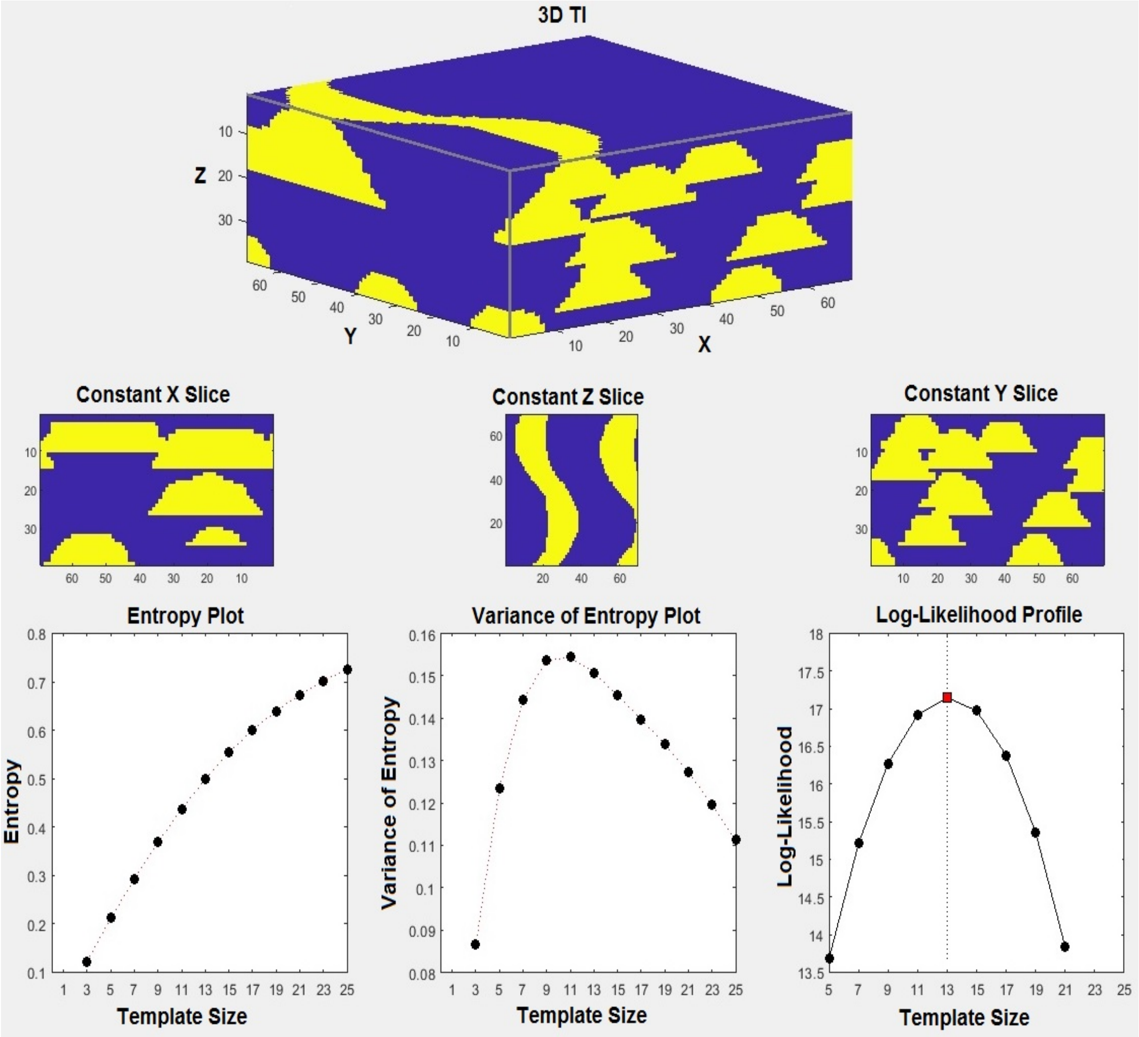

Figure 16. Three-dimensional training image (top) with three slice views (middle row) for checking textures and shapes, and the entropy-based determination of template size.

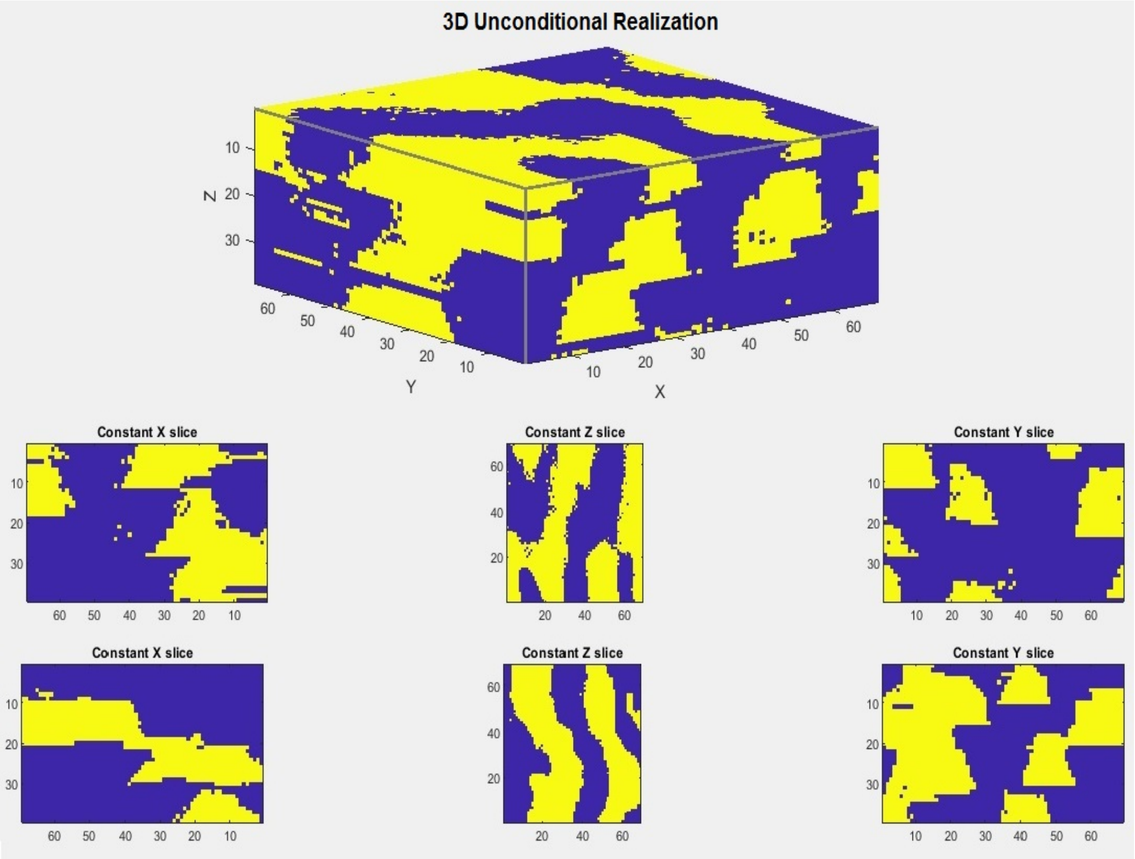

Figure 17. Unconditional three-dimensional simulation results and cross-sectional view.

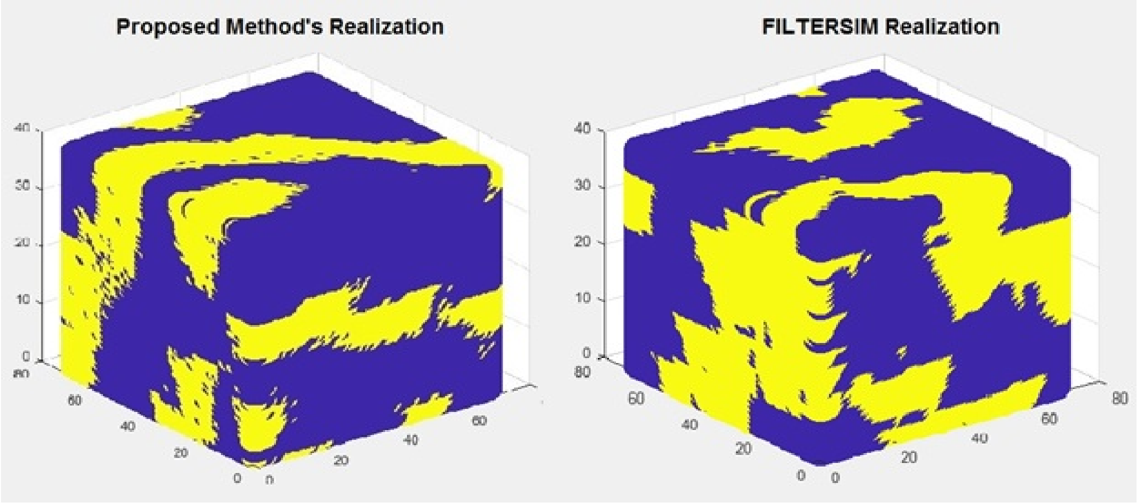

Figure 18. Three-dimensional realization generated by our proposed method (left) and realization generated by FILTERSIM algorithm (right).

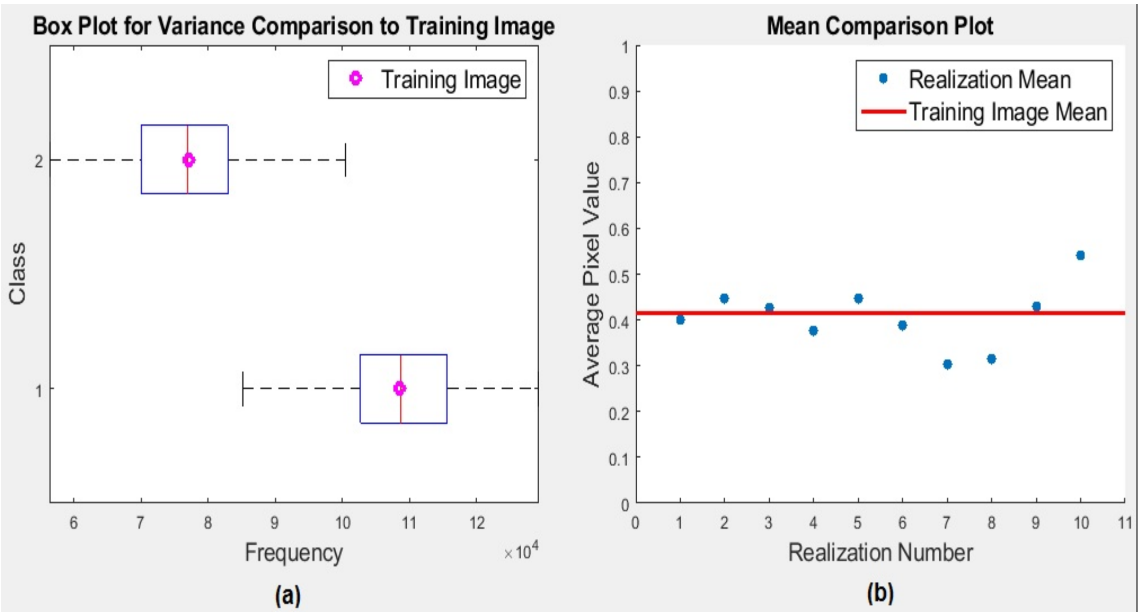

Figure 19. Categorical boxplots (a) and mean comparison plot (b) for the three-dimensional unconditional simulations, showing the comparison of the one-point statistics.

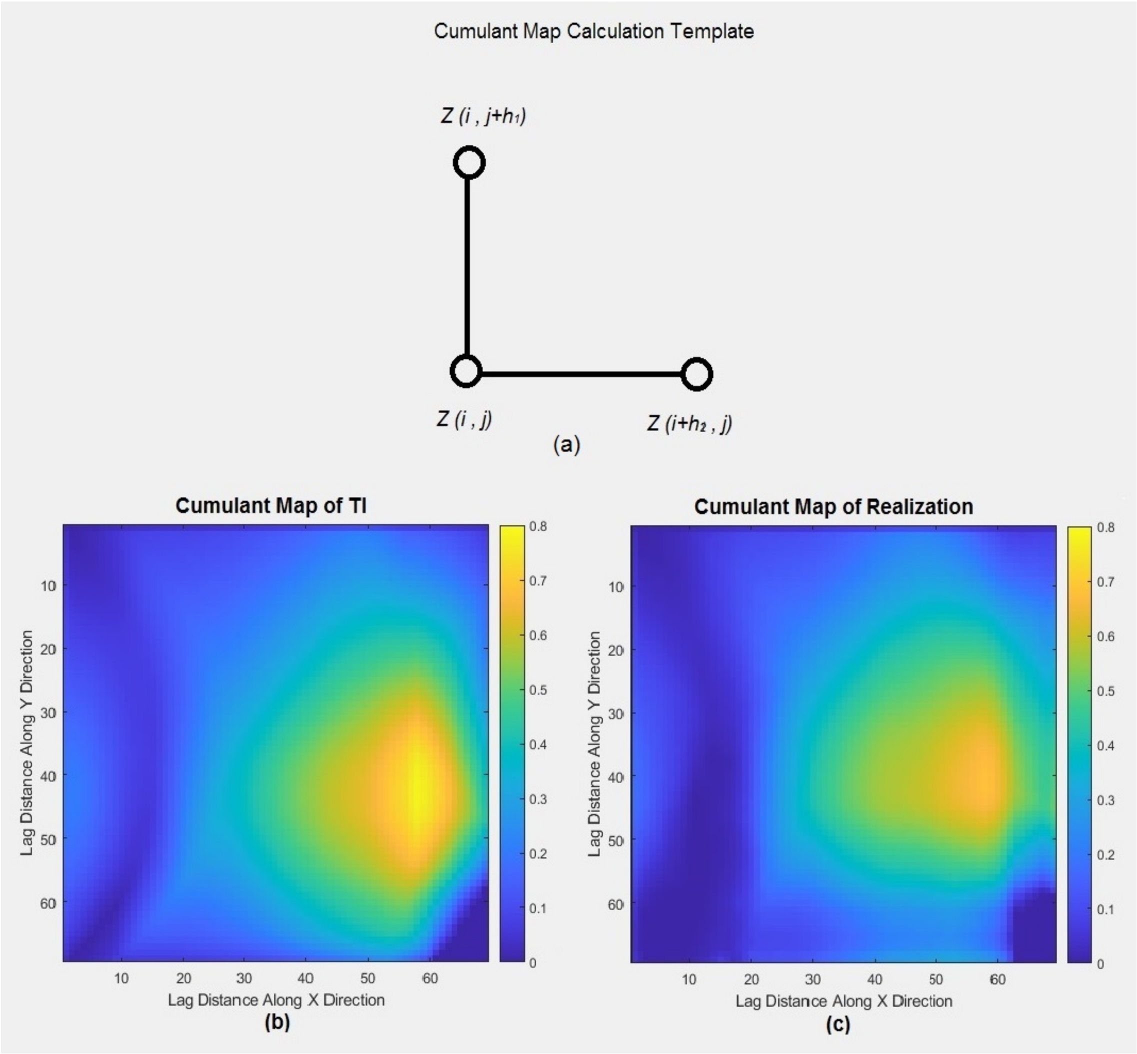

Figure 20. Cumulant maps of the categorical three-dimensional TI (b) and an unconditional simulation (c) plotted for higher-order validation of the proposed method.

2.3.2. Conditional Simulation

We tested our proposed method for the conditional simulation to check the accuracy of simulations in honoring hard data during the simulation process. Figure 21-a depicts 361 points of the hard data used for conditional simulation of the channel training image, arranged in a way to form a dense area of points, in addition to the vertical discrete streams. The purpose is to analyze the impact on the E-type and variance maps (Figs. 21-d & 21-e), in comparison with hard data plot (Fig. 21-a) and the unconditional simulations' E-type plots (Figs. 11-b & 11-c). Variance is very small in areas where the conditioning data are available. Among the hard dataset, 72 data (20%) are located in the center of the grid, forming a rectangular area, playing the role of a dense area of conditioning data. The other 289 hard data are distributed with equal distances across the grid, as shown in Fig. 21-a. Two regions are marked by red rectangle and the red ellipsoid, which contain a few non-channel-class hard data (in blue), not interrupted via channel-class points (in yellow). Those regions, as expected, show a low variance in simulations and non-channel-class dominance in the ensemble plot (Figs. 21-d & 21-e). The elliptical region contains an area of proportionally lower variance, which clearly shows how the algorithm honors the hard datasets, and how the variability increases as distance increases from the hard data. The conditioning data available in that area are all of the same class, so the other class has a little presence in that area (leading to low variance). The training image contains structures with "Y-like" shapes or shapes looking like "Y" partially. Thus, each local straight stream tends to turn right, left, or to both directions with certain angles after a certain length of vertical continuity. This can be inferred from the E-type plot of 100 realizations, shown in Fig. 21-d. The vertical streams were reconstructed with a reasonable accuracy according to the E-type plot, and the variability over those locations is less, even with such a small number of hard conditioning data points. The dense rectangular area of conditioning data of the same class is dominated by the opposite class in its left and right side in the E-type plot and in almost all of the realizations. It is an indicator of the ability of the model to respect the formation of channels. In fact, the width of original TI channels is honored in that area of dense conditioning data. The variability at above and bottom of the rectangular channel-class stream is high. It is due to the shapes that are present at the channel TI, as the channels maintain a straight path for a limited length, and after that, they tend to turn to other directions with certain angular deviations. The orange circle markes a single channel-class point, which is dominated by non-channel-class points as hard data. Looking at the E-type plot, it is observed that even this single point is honored by the algorithm, and it became a point where channels were passed through, mostly in two distinct directions. In addition, it is clear from the variance plot, that the variability increases gradually by increased distance from the point marked by the orange circle. In fact, as a pixel-based method, our proposed algorithm has a higher potential for respecting conditioning data. Another visible matter is that in Fig. 21 on the left side, we have the conditioning data near the domain border, whereas the conditioning datasets have a higher distance from the border at the right side. By looking at the variance plot and comparing the two sides, a clear representation of the difference in the amount of variability (uncertainty) of the simulations by the proposed method with and without conditioning data points at the borders is seen.

Figures 22 and 23 show the statistical validation results for 100 realizations of the conditional simulation for the channel TI, shown in Fig. 5-a. The first-, second- and third-order statistical comparison of the realizations with the corresponding channel TI shows a good match. In addition,the uncertainty of conditional simulation of the categorical variable via the proposed algorithm seems to be low. The box plots imply that the TI has the frequencies of the classes almost similar to the medians of the frequencies over all the realizations. The variograms from realizations also show a good agreement with the variogram of TI over approximately all lag values. As shown in Fig. 23, the three-point statistical comparison of the training image with the conditional simulation shows a reasonable visual match. From the cumulant maps, it can be inferred that for both small and large lag distances, the proposed algorithm is able to reproduce the third-order statistics of the channel TI well by conditional simulation.

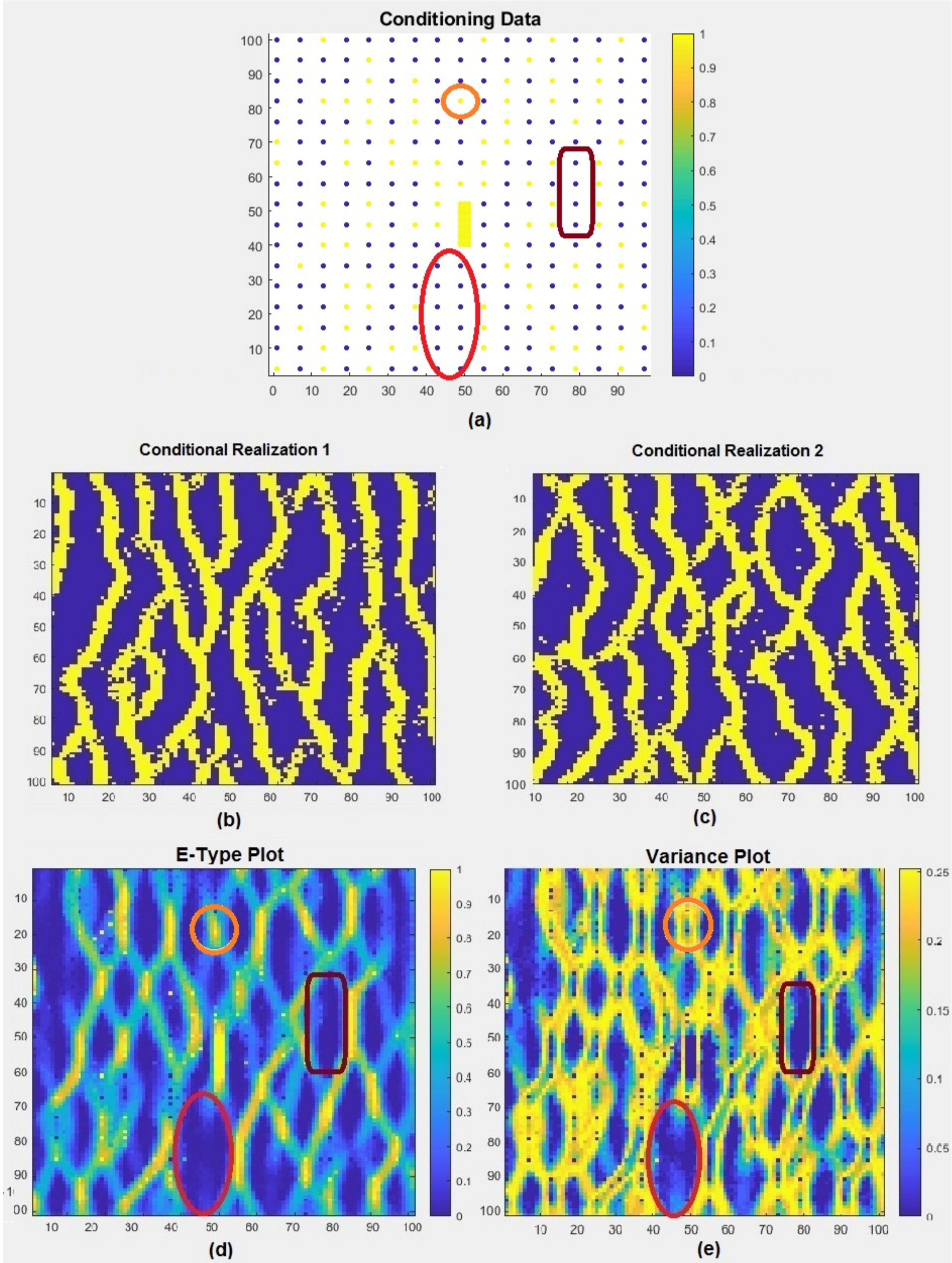

Figure 21. Conditioning data for the TI shown in Fig. 5-a (a), two conditional realizations (b & c), and the E-type plots of the 100 conditional simulations (d & e).

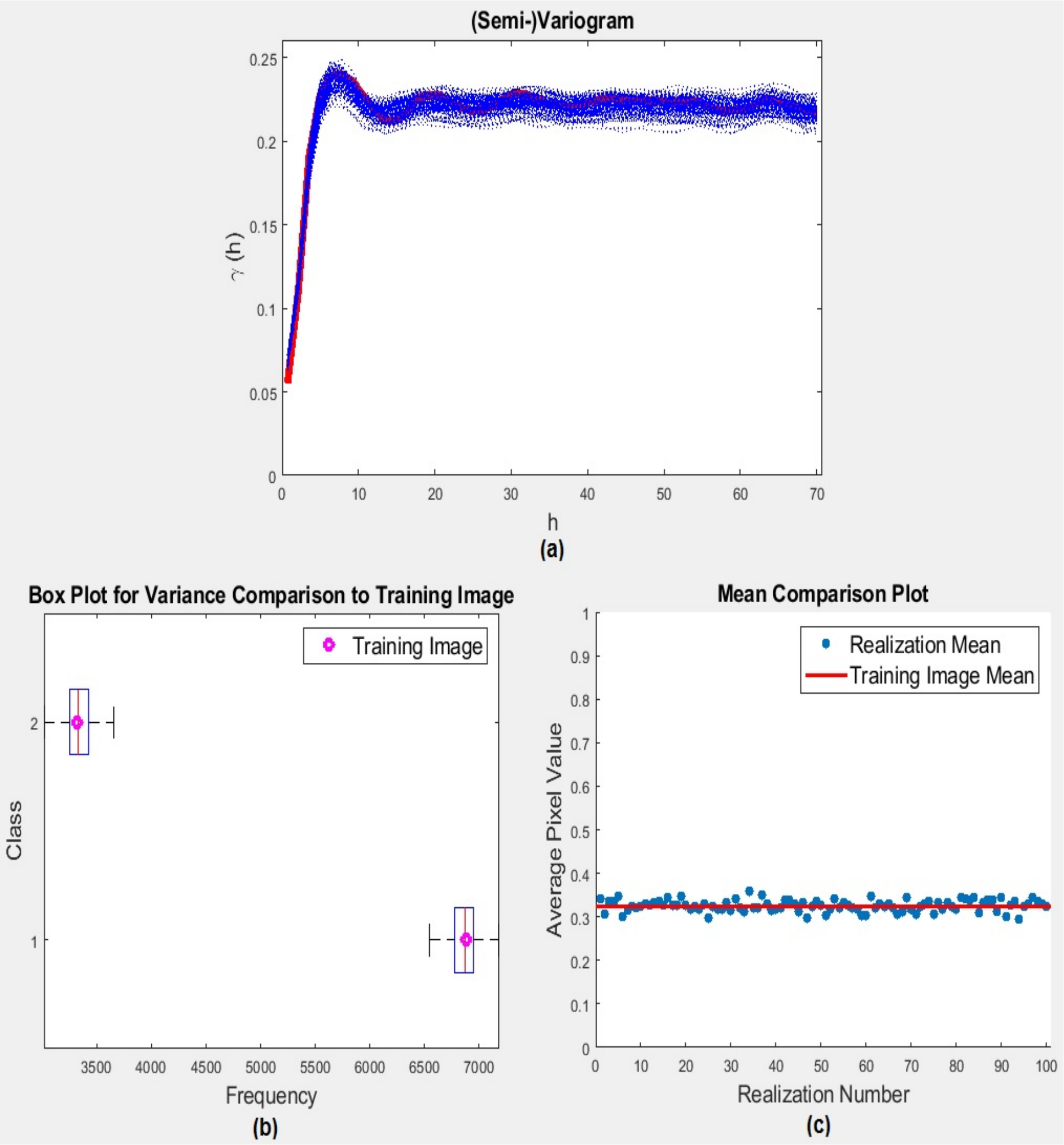

Figure 22. Variograms of the channel TI (in red) and the 100 conditional realizations (bottom), categorical boxplots (top row), and the mean comparison plot (middle).

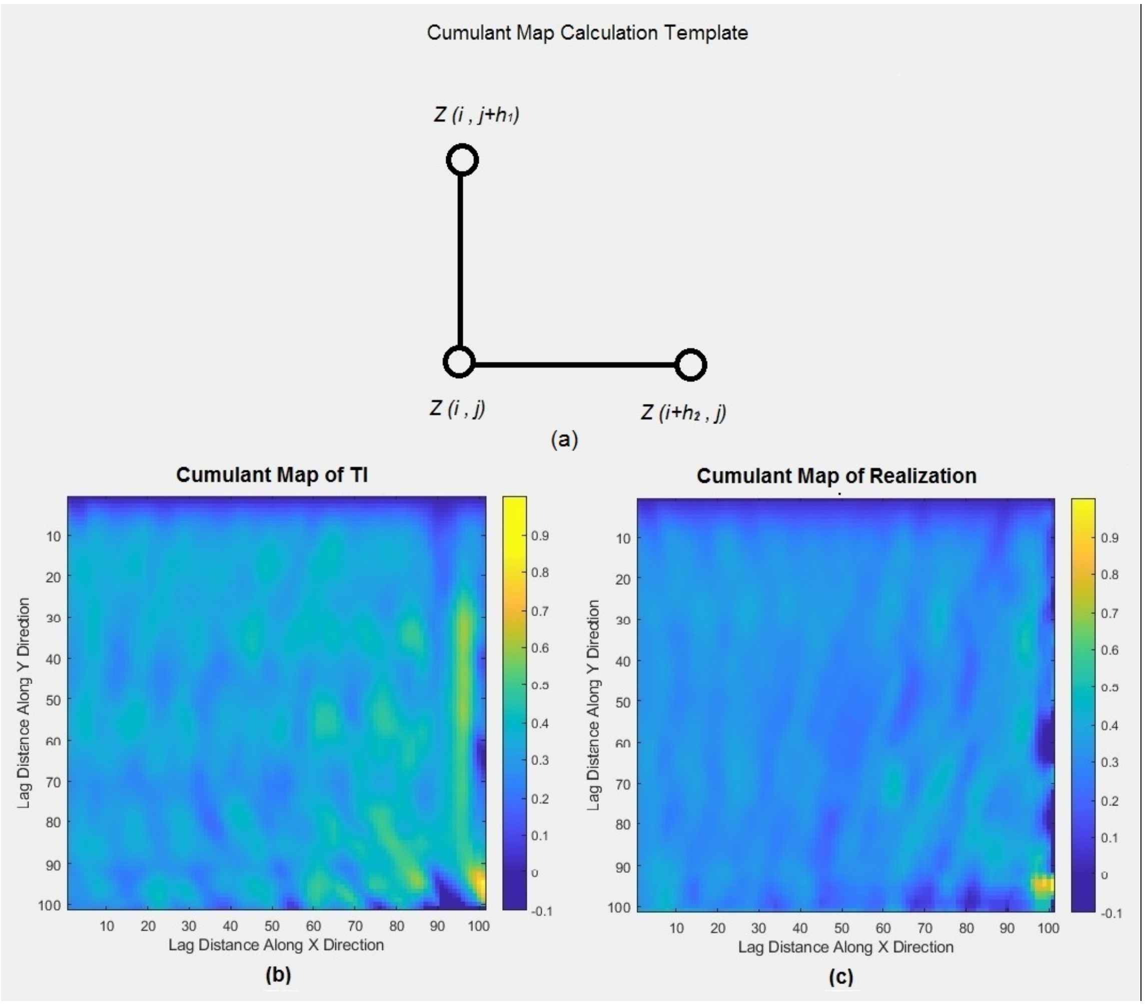

Figure 23. Cumulant maps of the categorical channel TI (b) and a conditional simulation (c) plotted for higher-order validation of the proposed method.

The quality of such realizations depends on maintaining the continuity and shape of the channels across the simulated images in the same manner as they are available in the training image. As the hard data are derived from the TI, the realizations should approximately match the TI in the sense of where specific textures are located, and how the statistical distribution is. In addition, the E-type map reflects the exhaustive training image when the number of hard data is high. We tested the algorithm on the channel TI via 1000 randomly distributed hard data, as shown in Fig. 24-a, and the results presented in Fig. 24 confirmed this claim. According to the variance plot, the variability of the 100 simulation values over most of the image is low, resulting in minimal uncertainty in modeling, compared to the previous case with fewer available hard data (Fig. 21). A conditionally simulated realization is also provided in Fig. 24-b, which is similar to the categorical channel TI. Looking at the E-type plot (Fig. 24-c), channels with continuity over larger distances are visible, compared to the previous case (Fig. 21). Thus, the effectiveness of hard data conditioning on the quality of realizations produced by our method is confirmed.

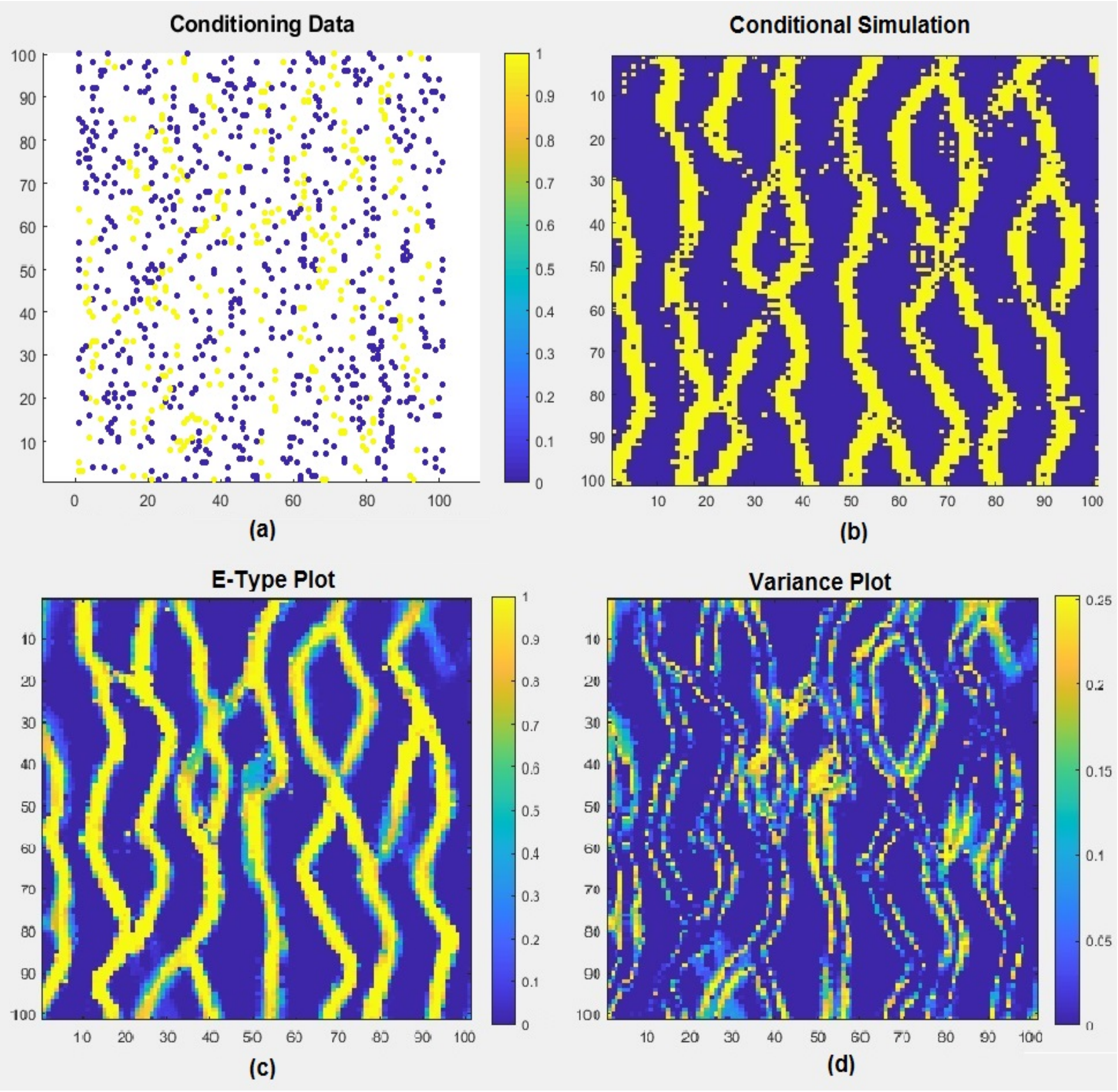

Figure 24. One thousand randomly distributed hard dataset (a), conditional realization (b), and the E-Type (c) and variance (d) maps for the channel TI.

For a conditional simulation of a continuous variable, 208 hard conditioning data were used (Fig. 25-a). The hard data are irregularly spaced and scattered all over the simulation grid domain. The two conditional simulation maps are shown in Figs. 25-b and 25-c. The visual comparison between training image and simulation maps confirm that the proposed method respect the hard data. The main channels' shapes and locations in the exhaustive continuous training image are well reproduced in the simulated images, especially at the top half of the image. The ensemble maps (E-type) are used to check how the realizations respect hard conditioning data. The E-type map is generated by 100 realizations. Looking at the green circular areas marked on in Fig. 25-a and 25-e, it can be seen that the lack of conditioning data lead to a higher uncertainty, and the simulations provide a high variance in areas with a sparse hard dataset. The areas highlighted by red ellipsoids were selected as the indicators of regions with low-value dominated hard data. Looking at those areas in the E-type and variance plots, it can be inferred that the algorithm respects the hard data by simulating the pixels within those regions with the majority of low values with low variability. On the other hand, within the area marked by the yellow rectangle, as the hard data were sampled from both heads of the bimodal distribution of the reference image, we can see a high variability among the realizations in the variance map. The white diamond in the conditioning data plot shows a single high-value pixel dominated by a few other low-value pixels as hard data. It is obvious from the E-type map that the algorithm honors the hard data point as the average values around that point seems to be higher than the surroundings. Honoring all hard data points is a clear advantage of the pixel-based simulation method, and demonstrated by the proposed algorithm, over pattern-based simulation. By using pattern-based simulation, individual points could be neglected while pasting patterns to the simulation grid.

The continuous training image used here has a complex bimodal histogram, and the algorithm is able to reproduce this distribution over all realizations (Fig. 26-a). A small overestimation of the left head frequency, and small under-estimation of the right head frequency, on average, were observed. The variogram comparison plot in Fig. 26, reveals that the second-order statistics of the continuous training image were also well reproduced, and the TI variogram falls in almost the middle of the ones for realizations for different lag distances. To show the multiple-point reproduction of the TI using the proposed pixel-based method, the three-points cumulant maps of the training image and a simulated realization were generated and are presented in Fig. 27. The algorithm output ensures the reproduction of the TI's third-order statistical characteristics.

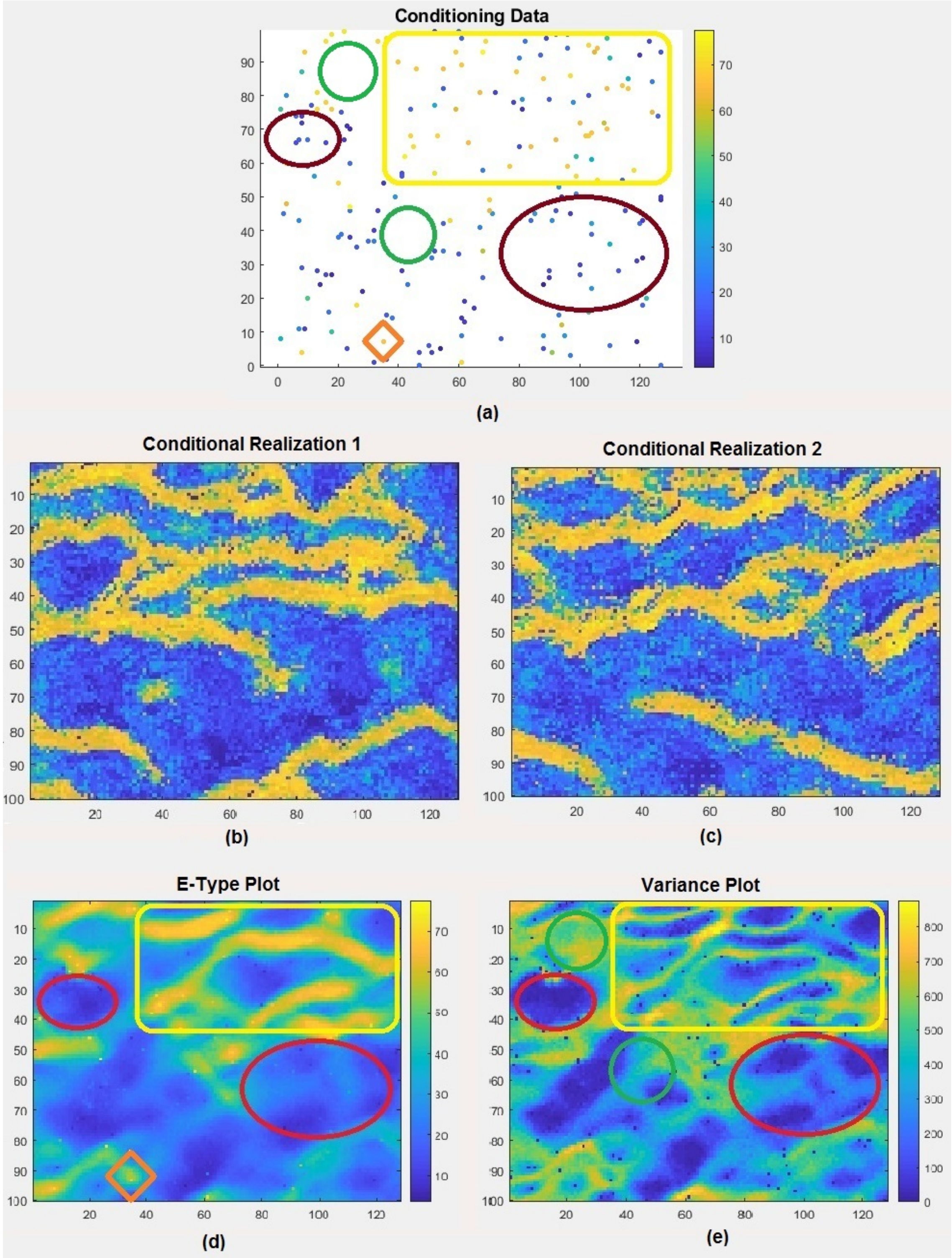

Figure 25. Conditioning data for the training image shown in Fig. 5-b (a), two conditional realizations by the proposed method (b & c), and the E-type plots (d & e) of 100 conditional simulations for the continuous TI.

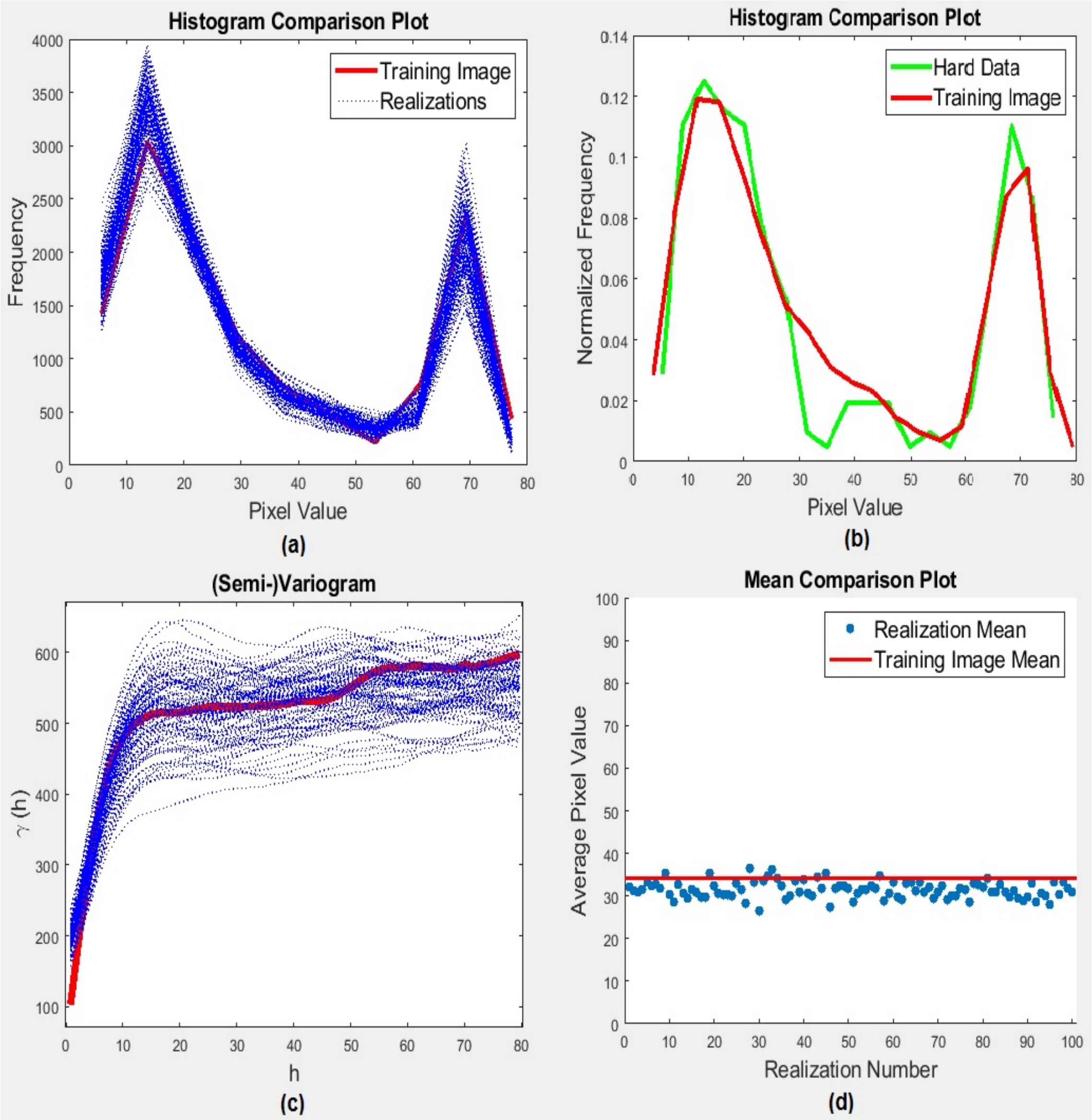

Figure 26. Histogram of the continuous training image shown in Fig. 5-b and the generated 100 conditional realizations (top), and mean comparison plot (bottom).

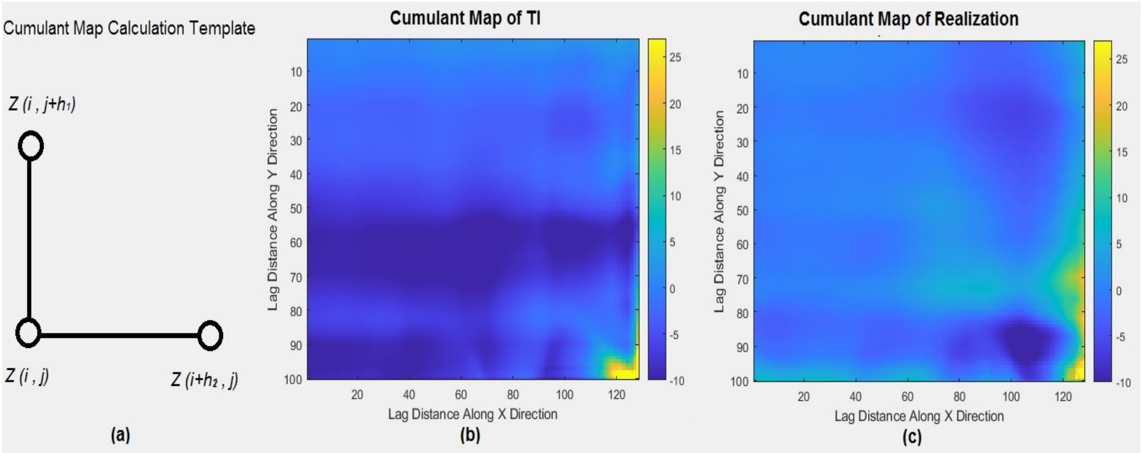

Figure 27. Cumulant maps of the continuous TI (b) and a conditional simulation (c) plotted for higher-order validation of the proposed method.

For continuous variable as well, the E-type map closely reproduces the reference TI when the number of hard data is sufficiently high. We tested the algorithm using 500 randomly distributed hard data (Fig. 28-a), and the results presented in Figs. 28-c and 28-d, confirmed this claim. It can be seen from Fig. 28-b that the channels occurring in the continuous TI are almost perfectly reproduced, and this happens due to the presence of a higher number of conditioning datasets. Within the region limited by the red lines (60<y<80), there are almost no high-value hard data available, and consequently, no high-value streams were reproduced. It shows the appropriate effectiveness of hard data on the simulation by the proposed algorithm. The variance plot confirms that as well, because the variability remained very low in that region as it was dominated by almost similar hard data pixel values. A comparison of the E-type maps in Figs. 28 and 26 proves the effect of the number of hard data on the simulation quality. In Fig. 28-c, the channels of the TI are almost perfectly reproduced as opposed to the ensemble map in Fig. 26-d, where higher uncertainty was revealed. Still, the percentage of the hard data for this case is under 4%.

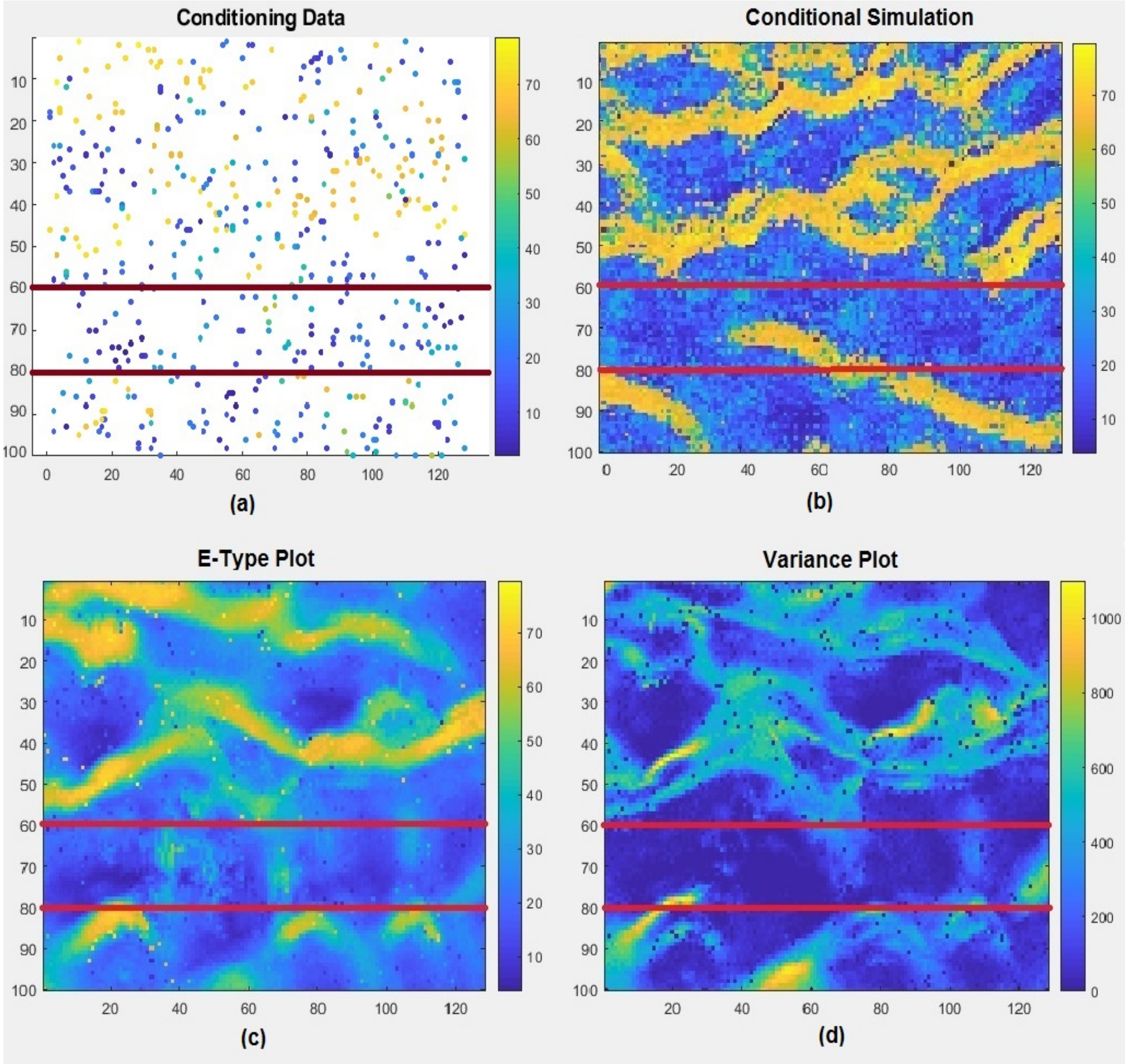

Figure 28. Five hundred randomly distributed hard datasets (a), conditional realization (b), and the E-Type and variance plots (c & d) for the continuous TI.

2.4. Three-Dimensional Case Study

After validation and testing, we applied the proposed method to a gold deposit to simulate categorical variables. The mining company has categorized the exploration drilling data into three different categories, i.e., high-grade, low-grade, and waste, to create wireframes (solid models) for volumetric analysis. We used the solid model developed by the mining company as the training image, and categorically-transformed composited exploration drilling data as hard conditioning data for our simulation. A total number of 8759 composite samples with a composite length of 10 m is available for this study. The pixel size of the training image is 25 m × 25 m × 10 m. We selected the same pixel size for simulation. Figure 29 shows the training image and hard conditioning data used in this study. The training image used in this case study has the size of [70×60×57], and a total of 239,400 blocks were simulated by the proposed method.

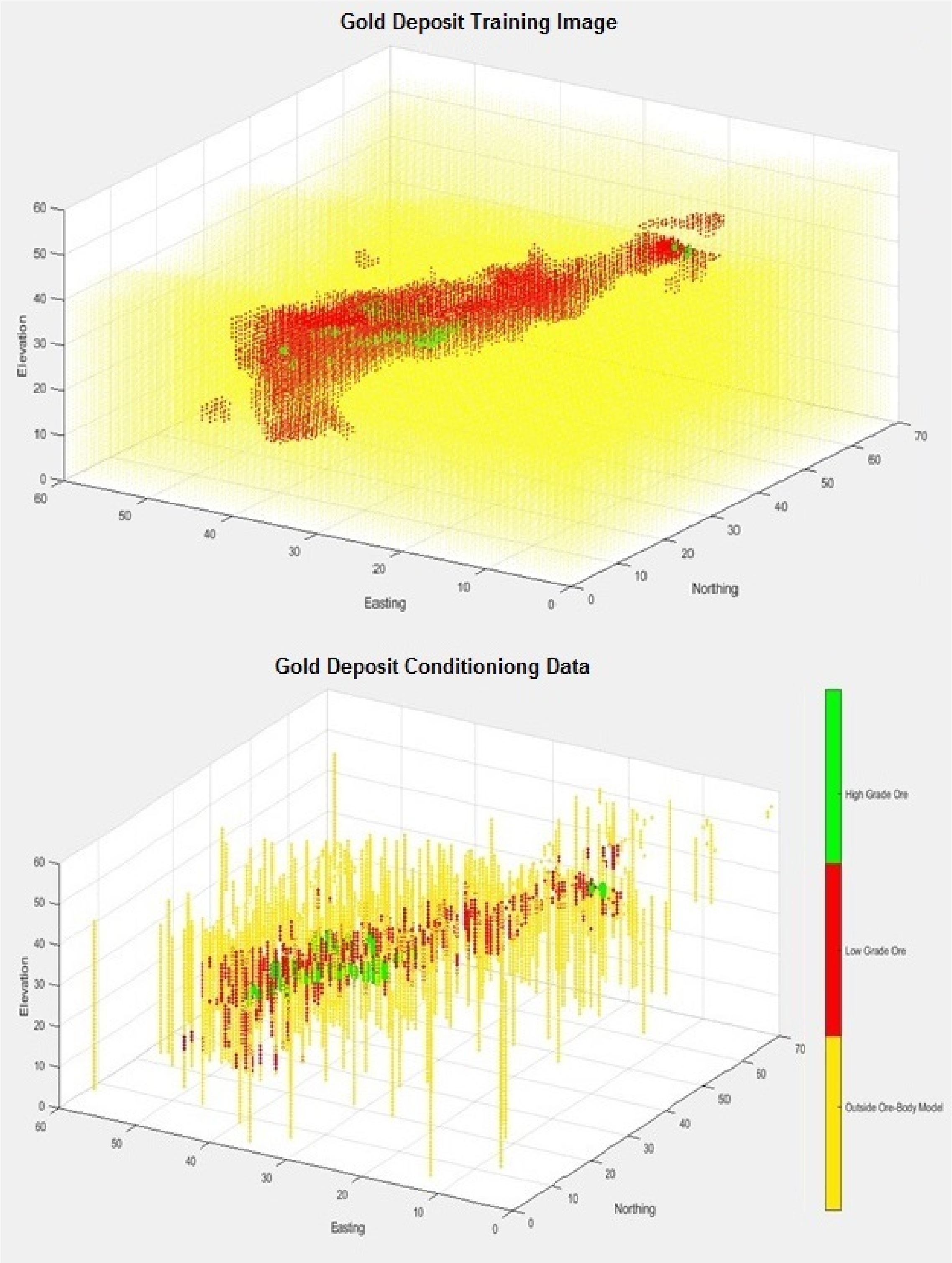

Figure 29. Three-categorical Training image of the gold deposit (top) and hard (conditioning) data gathered (bottom).

The solid, which is used as a training image, is generated based on updated geological interpretations, including the existence of faults, mineralization zones, lithological characterization, and lithological contacts. Drilling data were considered as the source of information with almost the highest certainty. To perform simulation, we firstly determined the template size to extract the patterns database from the training image, for which the results showed in Fig. 30 indicates the optimal size would be **t**=[13 13 13]. For simulation, all nodes that are falling above the topography were excluded from the simulation process.

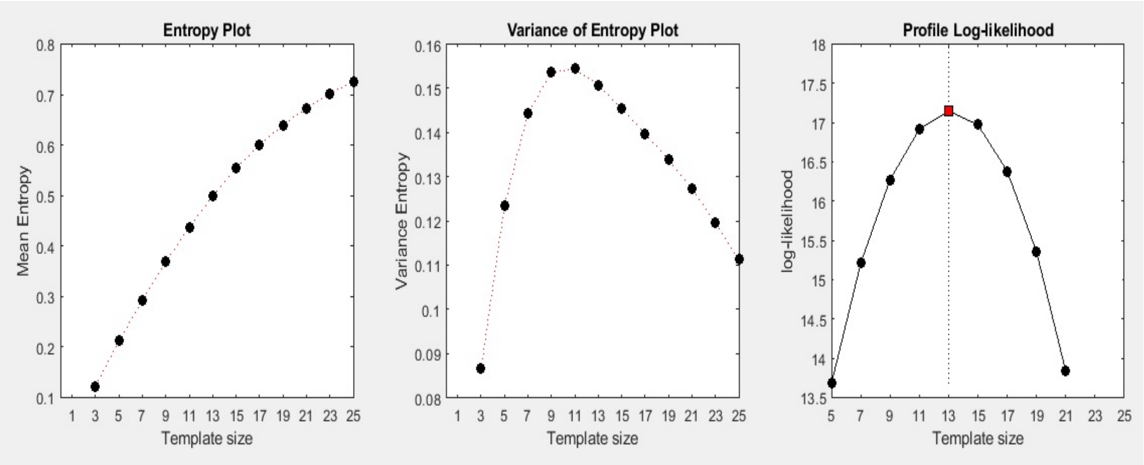

Figure 30. Entropy-based determination of template size for the three-dimensional TI in Fig. 29.

After extraction of patterns database, we removed the patterns containing the above-topography values, and then stochastic embedding and clustering were performed. We performed the same approach mentioned for the previous three-dimensional simulation, which means that we used *ccdf* for sampling from the highly populated clusters (more than 100 members). Figure 31 shows two generated realizations. Figure 32 shows a statistical comparison between hard data, TI, and realizations.

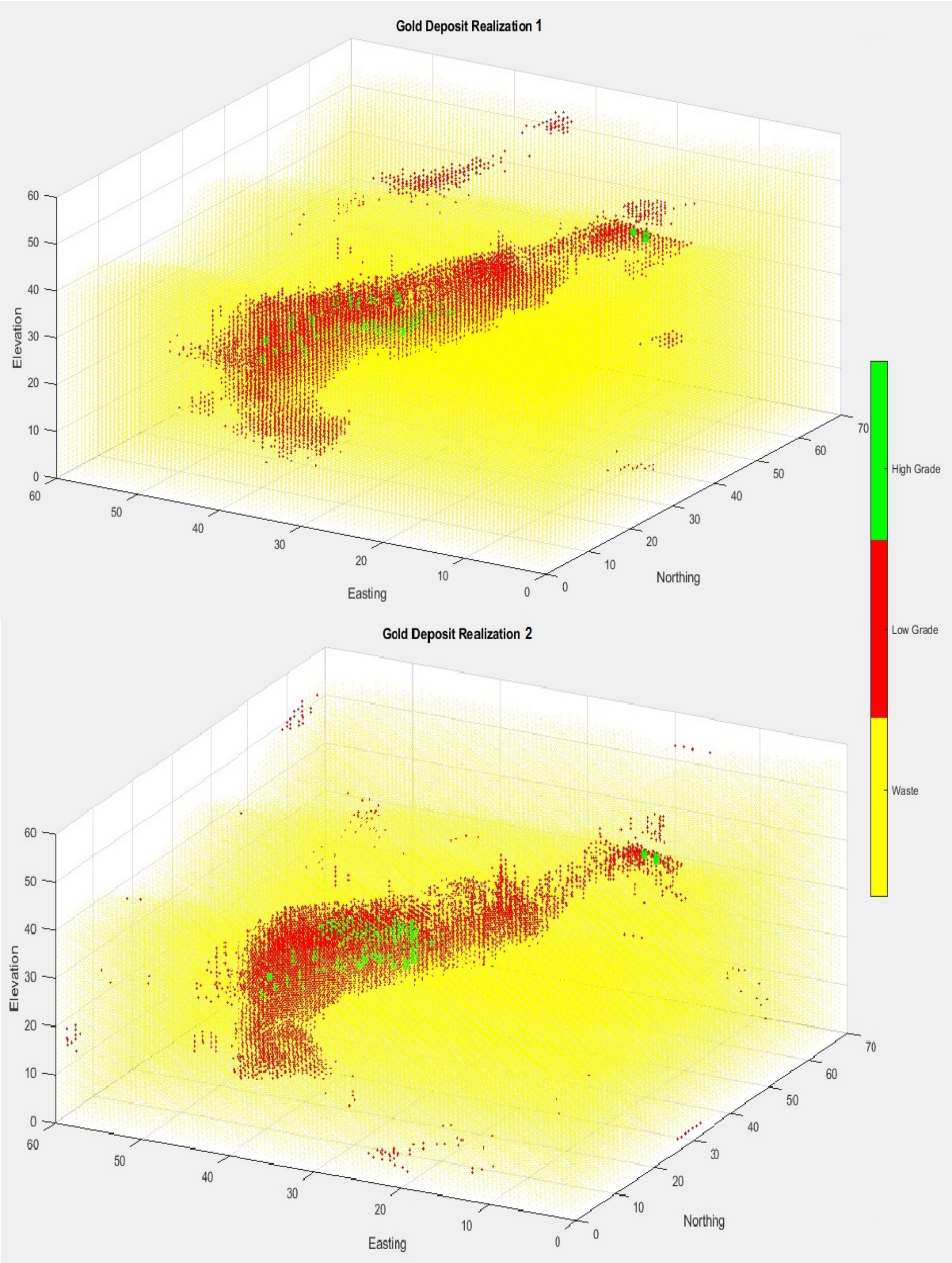

Figure 31. Two generated three-dimensional realizations of the conditional three-categorical simulation.

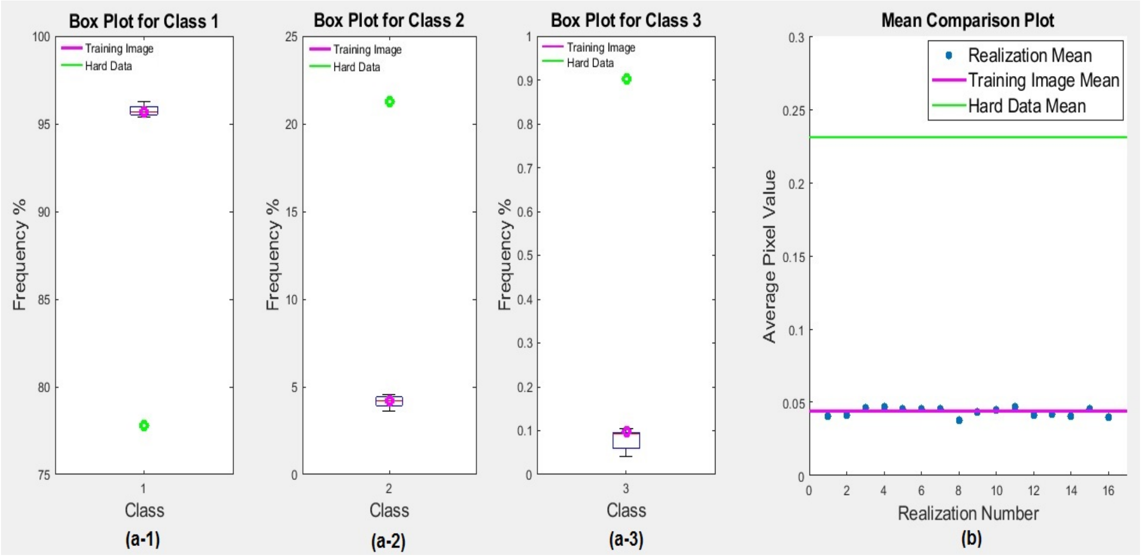

Figure 32. Boxplots for comparison of the statistics of the three-dimensional TI, the hard (conditioning) data, and the generated realizations (a), and mean comparison plot (b).

From Figure 32, we note that there are some deviations between the hard data and simulations statistics; however, the results show an excellent match between training image and simulations. The reason for this observation is that the statistics and frequency of different classes are not the same for both the conditioning data and the TI. Thus, the simulations, as expected, honor the statistics of the TI rather than the hard data. The proposed method, like other MPS methods, is supposed to reproduce the TI statistics, as the realizations derived by using this methodology are training-image-driven. The statistics of the simulations tend to get skewed towards the conditioning data while remaining similar to the TI. It is specifically clear for the class three as shown in Fig. 32-a-3. Thus, it can be concluded that, in case of having similar statistics between TI and hard data, the simulations reproduce the statistical properties that honor both of the TI and conditioning data. Figure 33 shows a cross-sectional view of TI and a generated realization to check the ability of the model to reconstruct structures in addition to honoring the conditioning data. Having a visual comparison between the two plots confirms the accuracy of the proposed pixel-based MPS method in the generation of realizations that reconstruct the structures of the training image, and at the same time, honoring the conditioning data.

In this case study three-dimensional TI, we deal with the dense areas of the conditioning data indicating the areas in which cumulations of orebody materials with different grades happeng. This necessitates the evaluation of the results in this regard, in addition to checking for the reproduction of textures and continuities. As observed in Fig. 33(the slice views), even the high-grade zones, which were a minority class compared to the other two classes, were formed via the pixel-based simulation presented in this study. The number of high-grade class among the training image pixels is 189 (0.08%), and among the hard datasets is 79 (0.14%).

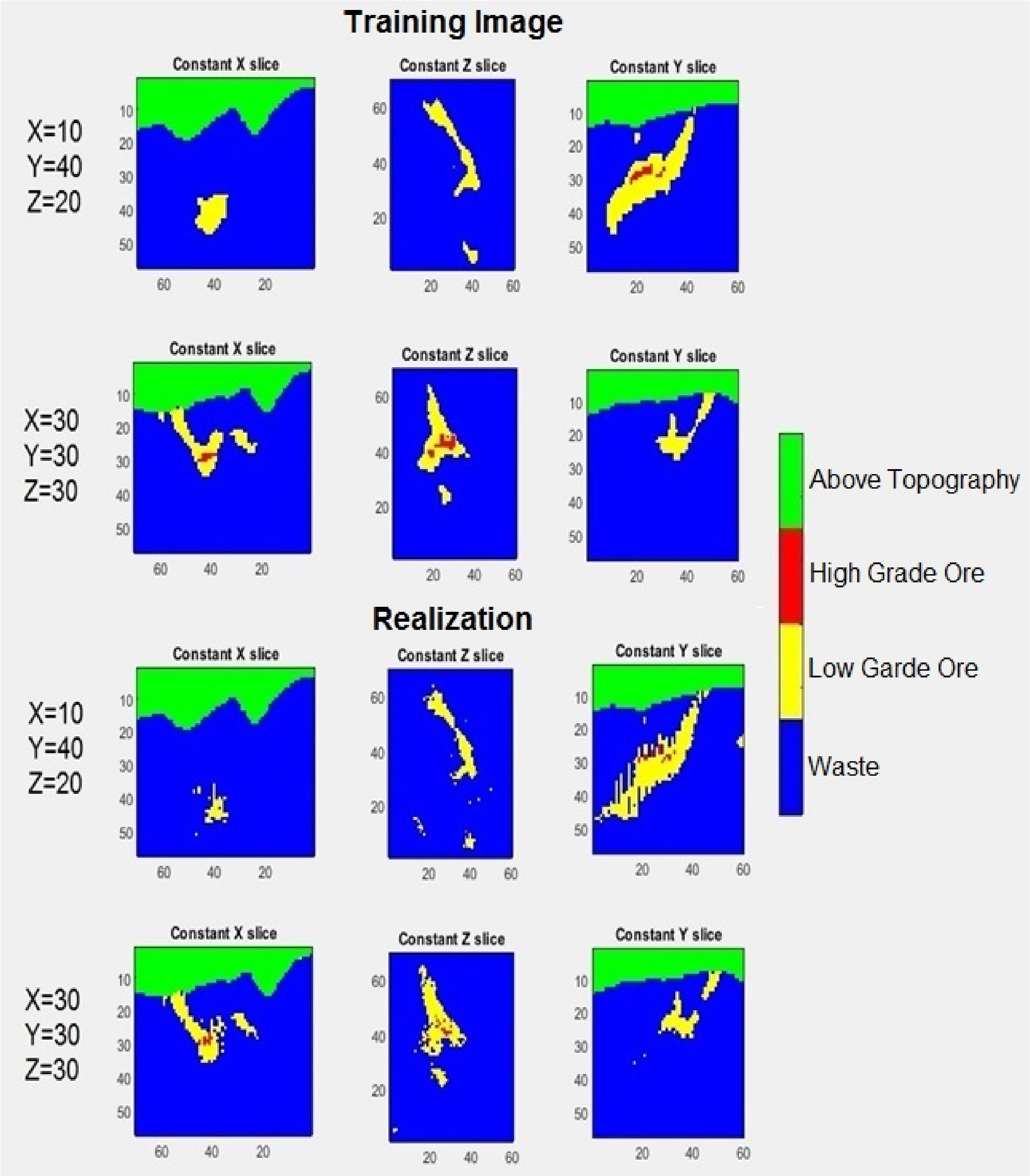

Figure 33. Three-dimensional TI's six slice views (two in each direction) for showing textures and shapes (top two rows), and the realization's six slice views (two in each same direction) for checking if/how the simulations are reproducing the TI textures and shapes (bottom two rows).

2.5. Runtime and Code Availability

Our proposed method can be considered a fast MPS method in both two-dimensional and three-dimensional simulations, based on the information provided in Table 1. The codes were run on a PC with Windows 10 (64-bit) and with configurations including 1.70 GHz Intel(R) Xeon(R) Bronze 3106 CPU (2 processors) and 96 GB of Installed Memory (RAM). The computer was a shared system exploited simultaneously by several users. The sources codes used to produce the results published in this paper, and the associated supplementary materials are available online in the GitHub repository of the first author, https://github.com/adel-asadi/Pixel_based_MPS. It should be noted that our code is implemented in MATLAB, using parallel computing for the simulation step. Therefore, the speed could be improved in environments like Java, Python, and C++. We cannot compare the speed of our method with other algorithms that are not written and run in MATLAB. However, especially due to the dimensionality reduction and subsequent fast clustering, the runtime of the code on different cases is acceptably low, which is a promising criterion in the success of the methodology in terms of being used as an application in the industry.

Table 1. Simulation times by the training image and simulation type.

Training Image	Simulation Grid Size	Number of Hard Data	Simulation Type	Number of Realizations	Total Time (Seconds)	Time per Realization (Seconds)
Two-dimensional Categorical (Fig. 5-a)	101×101 (10201)	-	Unconditional	100	516	5.16
	101×101 (10201)	361	Conditional	100	429	4.29
	101×101 (10201)	1000	Conditional	100	936	9.36 [(1)]

Two-dimensional Continuous (Fig. 5-b)	100×128 (12800)	-	Unconditional	100	510	5.10
	100×128 (12800)	208	Conditional	100	388	3.88
	100×128 (12800)	500	Conditional	100	874	8.74 [1]
Three-dimensional TI (Fig. 16)	69×69×39 (185679)	-	Unconditional	10	24520	2452 [2]
Three-dimensional TI (Fig. 29)	70×60×57 (239400)	8759	Conditional	10	9850	985 [3]

The information provided by Table 1 also indicates the followings points as marked within the table:

1. Conditional simulation via the same training image is faster than the unconditional simulation because the less number of nodes need to be filled via the MPS simulation. However, we also tested the method with a larger number of hard data (for both of the two-dimensional TIs shown in Fig. 5), and saw an increase in the computational time, which is more than the unconditional simulation time because of the more expensive similarity detection.

2. By assigning MinPts=3, the computational time decreases significantly (542 seconds for this case), but the reconstructions can be considered suboptimal. We decided to assign MinPts=2 for quality results. However, it should be changed and tested via the trial and error for different training images and scenarios. Another observation of our experiments was that having a higher number of conditioning data to help the simulation in reproducing the desired textures and statistics, can let the user assign MinPts=3 for faster simulations while maintaining outputs with acceptable quality. It should be noted that assigning MinPts=3 decreased the number of clusters from 1947 to 374 for this case. In an experiment, removal of *ccdf* partial usage in similarity detection for MinPts=3 case, increased the timing to 792 seconds.

3. The case study three-dimensional TI had 46339 points above topography, which did not need to be simulated. The number of nodes to be simulated was 193061 (including the points which were filled using hard data). In addition, the computational time is less than the unconditional simulation for the other three-dimensional TI, because the number of clusters of patterns formed for the case study TI was 463, which is almost a quarter of 1947 for the other two-categorical three-dimensional TI.

2.6. Conclusions and Future Works

This research proposed a new pixel-based simulation algorithm using different machine learning techniques. The results showed that t-Distributed Stochastic Neighbors Embedding (t-SNE algorithm), and Density-based Spatial Clustering of Applications with Noise (DBSCAN algorithm) are efficient methods for dimensional reduction and classification of pattern database generated from a training image. In addition, we showed that the algorithm is useful to take advantage of the pixel-based MPS algorithms as the basic approach of our study. We also employed an automatic method for template size determination and a number of optimization techniques in order to fully automate our proposed MPS method. In this study, we achieved a high performance while reconstructing complex features and continuities although our method uses a pixel-based approach of geostatistical simulation. In terms of respecting hard data, the MPS method showed its ability to produce high-quality conditional simulations while also honoring the statistical properties of the training image. Continuous variables were also tested for simulation using the algorithm, and the results of continuous training image simulation were promising, similar to the categorical simulations. Another main advantage of the algorithm was its computational speed, as we achieved a relatively high-speed methodology in this work, and this criterion is very important to consider for real-life three-dimensional problems. The validation results for different scenarios showed that the algorithm is capable of solving related problems in various disciplines, including but not limited to mineral resources modeling.

Although the proposed approach is computationally fast, there is still some scope for further improvement. The t-SNE is the most time-consuming step in our proposed method. The DBSCAN clustering is fast and efficient enough but inappropriate for a high-dimensional dataset. Therefore, implementing a fast clustering algorithm for high-dimensional datasets instead of dimensionality reduction and subsequent clustering, as proposed in this study, can significantly improve the computational time. If this could be achieved, the proposed method will be significantly faster while being efficient. Another path towards the development of the proposed method will be the implementation and testing of the algorithm via non-stationary simulations, multivariate modeling, block conditioning, simulations post-processing, optimization and enhancement of the reconstructions, dealing with Spatio-temporal datasets, and other widespread MPS applications in different fields of studies, using various training images and datasets for hard or soft conditioning to the simulation process.

3. Concluding Remarks

The goal of this research was to introduce a novel pixel-based multiple-point geostatistical simulation algorithm to increase the efficiency of such models. Reduction in computational time, honoring hard conditioning data, and improvement in reproducing the training image patterns and statistics are important measures of evaluating the performance of the algorithm. To achieve that, we proposed the implementation of several advanced machine learning algorithms for dimensionality reduction, clustering of patterns database, and automation of the (MPS) algorithm. PCA and t-SNE for mapping high-dimensional patterns database, and DBSCAN for subsequent clustering of mapped data, showed their efficiency in this study. The proposed pixel-based simulation approach proved its performance in different conditions via various validation tests performed on the generated realizations of categorical and continuous variables, using 2D and 3D training images, and via conditional and unconditional simulations. The proposed algorithm is fast, based on the recorded run-times; however, the stochastic embedding part of the process is consuming the most of the simulation time. Thus, a recommendation for future study can be the implementation of a faster mapping technique. Besides, we recommend adding other extensions such as non-stationary and multivariate modeling, in addition to testing the method on various applications.

4. References

Abdollahifard MJ (2016) Fast multiple-point simulation using a data-driven path and an efficient gradient-based search. Computers & Geosciences Journal (Elsevier), 86 (2016) 64–74. http://dx.doi.org/10.1016/j.cageo.2015.10.010.

Abdollahifard MJ, Faez K (2014) Fast direct sampling for multiple-point stochastic simulation. *Arab J Geosci* **7,** 1927–1939 (2014). https://doi.org/10.1007/s12517-013-0850-4.

Abdollahifard MJ, Mariéthoz G, Ghavim M (2019) Quantitative evaluation of multiple-point simulations using image segmentation and texture descriptors. *Comput Geosci* **23,** 1349–1368 (2019). https://doi.org/10.1007/s10596-019-09901-z.

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

Arpat B, Caers J (2004) A Multiple-scale, Pattern-Based Approach to Sequential Simulation. In Leuangthong, O. & Deutsch, C. V. (Eds) Quantitative Geology and Geostatistics, 14, 255-264.

Arpat GB, Caers J (2007) Conditional Simulation with Patterns. *Math Geol* **39,** 177–203 (2007). https://doi.org/10.1007/s11004-006-9075-3.

Akbari Z, Unland R (2016) Automated Determination of the Input Parameter of DBSCAN Based on Outlier Detection. In: Iliadis L., Maglogiannis I. (eds) Artificial Intelligence Applications and Innovations. AIAI 2016. IFIP Advances in Information and Communication Technology, vol 475. Springer, Cham.

Boucher A (2009) Considering complex training images with search tree partitioning. Comput. Geosci. 35, 1151–1158 (2009).

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

Chatterjee S, Dimitrakopoulos R, Mustapha H(2012-b), Dimensional Reduction of Pattern-Based Simulation Using Wavelet Analysis, Math. Geosci., 44(3), 343-374.

Dagasan Y, Renard P, Straubhaar J, Erten O, Topal E (2018) Automatic Parameter Tuning of Multiple-Point Statistical Simulations for Lateritic Bauxite Deposits. Minerals 2018, 8, 220.

Efros A. Freeman WT (2001) Image Quilting for Texture Synthesis and Transfer. Proceedings of the ACM SIGGRAPH Conference on Computer Graphics. 341-346.

Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. Simoudis, Evangelos; Han, Jiawei; Fayyad, Usama M. (eds.). In proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press. pp. 226–231. CiteSeerX 10.1.1.121.9220. ISBN 1-57735-004-9.

Friedman JH, Bentely J, Finkel RA (1977) An Algorithm for Finding Best Matches in Logarithmic Expected Time. ACM Transactions on Mathematical Software 3, no. 3 (1977): 209–226.

Gravey M, Mariethoz G (2019) Quantile Sampling: a robust and simplified pixel-based 2 multiple-point simulation approach. Geoscientific Model Development Discussions, December 2019, https://doi.org/10.5194/gmd-2019-211.

Guardiano FB, Srivastava RM (1993) Multivariate Geostatistics: Beyond Bivariate Moments. In: Soares A. (eds) Geostatistics Tróia '92. Quantitative Geology and Geostatistics, vol 5. Springer, Dordrecht. https://doi.org/10.1007/978-94-011-1739-5_12.

Honarkhah M, Caers J (2010) Stochastic Simulation of Patterns Using Distance-Based Pattern Modeling, *Geosci.*, *42*(5), 487-517, https://doi:10.1007/s11004-010-9276-7.

Hotelling H (1933) Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology, 24, 417–441, and 498–520.

Huysmans M, Dassargues A, (2001) Direct multiple-point geostatistical simulation of edge properties for modeling thin irregularly shaped surfaces. Math. Geosci. 43, 521 (2011).

Jain AK (2010) Data clustering: 50 years beyond K-means. Pattern Recognit Lett 31:651–666.

Jolliffe IT (2002) Principal Component Analysis. 2nd ed., Springer, 2002.

Journel AG, (2003) Multiple-point Geostatistics: A State of the Art. Unpublished Stanford Center for Reservoir Forecasting paper.

Li X, Mariethoz G, Lu D, Linde N, (2016) Patch-based iterative conditional geostatistical simulation using graph cuts. Water Resour. Res. 52, 6297–6320 (2016).

LJP van der Maaten (2009) Learning a Parametric Embedding by Preserving Local Structure. In Proceedings of the Twelfth International Conference on Artificial Intelligence & Statistics (AI-STATS), JMLR W&CP 5:384-391.

van der Maaten LJP (2014) Accelerating t-SNE using Tree-Based Algorithms, Journal of Machine Learning Research 15 (2014) 1-21.

van der Maaten LJP, Hinton GE (2008) Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research 9(Nov):2579-2605, 2008.

MacQueen JB (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of 5th Berkeley symposium on mathematical statistics and probability, vol 1, pp 281–297.

Mahmud K, Mariethoz G, Caers J, Tahmasebi P, Baker A (2014) Simulation of Earth textures by conditional image quilting, Water Resources Research Journal, 50, 3088–3107, https://doi:10.1002/2013WR015069.

Mariethoz G, Caers J (2015) Multiple-point Geostatistics: Stochastic Modeling with Training Images, First Edition, by Gregoire Mariethoz and Jef Caers, © 2014 John Wiley & Sons, Ltd.

Mariethoz G, Straubhaar J, Renard P, Chugunova T, Biver P (2015) Constraining distance-based multipoint simulations to proportions and trends. Environ. Model Softw. 72, 184–197 (2015).

Mariethoz G, Renard P, Straubhaar J (2010) The direct sampling method to perform multiple-point simulations, Water Resour. Res., 46(W11536), 10.1029/2008WR007621.

Mead A (1992) Review of the Development of Multidimensional Scaling Methods. Journal of the Royal Statistical Society. Series D (The Statistician). 41 (1): 27–39.

Melnikova Y, Zunino A, Lange K, Cordua KS, Mosegaard K (2015) History matching through a smooth formulation of multiple-point statistics. Math. Geosci. 2015, 47, 397–416.

Minniakhmetov I, Dimitrakopoulos R, Godoy M, (2018) High-Order Spatial Simulation Using Legendre-Like Orthogonal Splines. Mathematical Geosciences Journal 50, 753–780 (2018). https://doi.org/10.1007/s11004-018-9741-2.

Mustapha H, Dimitrakopoulos R (2010) A new approach for geological pattern recognition using high-order spatial cumulants. Computers & Geosciences, 36 (2010) 313–334, https://doi:10.1016/j.cageo.2009.04.015.

Mustapha H, Dimitrakopoulos R (2011) HOSIM: A high-order stochastic simulation algorithm for generating three-dimensional complex geological patterns. Computers & Geosciences 37 (2011) 1242–1253. https://doi:10.1016/j.cageo.2010.09.007.

Mustapha H, Chatterjee S, Dimitrakopoulos R (2014) CDFSIM: Efficient Stochastic Simulation Through Decomposition of Cumulative Distribution Functions of Transformed

Spatial Patterns. *Math Geosci* **46,** 95–123 (2014). https://doi.org/10.1007/s11004-013-9490-1.

Parra A, Ortiz JM (2011) Adapting a texture synthesis algorithm for conditional multiple point geostatistical simulation. Stoch. Env. Res. Risk A. 25, 1101–1111 (2011).

Rezaee H, Marcotte D, Tahmasebi P, Saucier A (2015) Multiple-point geostatistical simulation using enriched pattern databases. Stoch. Env. Res. Risk A. 29, 893–913 (2015).

Sander J, Ester M, Kriegel HP, Xu X (1998) Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. Data Mining and Knowledge Discovery. Berlin: Springer-Verlag. 2 (2): 169–194. https://doi:10.1023/A:1009745219419.

Schubert E, Sander J, Ester M, Kriegel HP; Xu X (2017) DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. ACM Trans. Database Syst. 42 (3): 19:1–19:21. https://doi:10.1145/3068335. ISSN 0362-5915.

Straubhaar J, Renard P, Mariethoz G, Froidevaux R, Besson O (2011) An improved parallel multiple-point algorithm using a list approach. Math Geosci 43(3):305 328. https://doi.org/10.1007/s11004-011-9328-7.

Straubhaar J, Walgenwitz A, Renard P (2013) Parallel Multiple-Point Statistics Algorithm Based on List and Tree Structures. Math. Geosci. J. (2013), 45:131–147. https://doi:10.1007/s11004-012-9437-y.

Strebelle S (2002) Conditional Simulation of Complex Geological Structures Using Multiple-Point Statistics, Geol., 34(1), 1-22.

Strebelle S (2012) Multiple-Point Geostatistics: from Theory to Practice, Ninth International Geostatistics Congress, Oslo, Norway June 11 – 15, 2012.

Tahmasebi P (2018) Multiple-point statistics, Handbook of Mathematical Geosciences: Fifty Years of IAMG, Edited by B. S. Daya Sagar, Qiuming Cheng and Frits Agterberg, Springer, Netherlands.

Tahmasebi P, Hezarkhani A, Sahimi M (2012) Multiple-point geostatistical modeling based on the cross-correlation functions, Computational Geosciences Journal, https://doi:10.1007/s10596-0129287-1.

Tahmasebi P, Sahimi M, Caers J (2014) MS-CCSIM: Accelerating pattern-based geostatistical simulation of categorical variables using a multi-scale search in Fourier space, Comp. & Geosci., 67(0), 75-88.

Tahmasebi P, Sahimi M (2015) Geostatistical Simulation and Reconstruction of Porous Media by a Cross-Correlation Function and Integration of Hard and Soft Data. Transport in Porous Media 107(3): 871-905.

van der Maaten, LPJ (2013) *Barnes-Hut-SNE*. arXiv:1301.3342 [cs.LG], 2013.

Wu J, Zhang T, Journel A (2008) Fast FILTERSIM simulation with scorebased distance. Math Geosci 40(7):773–788.

Yang L, Hou W, Cui C, Cui J (2016) GOSIM: A multi-scale iterative multiple-point statistics algorithm with global optimization. Computers & Geosciences, 89 (2016) 57–70. http://dx.doi.org/10.1016/j.cageo.2015.12.020.

Yao L, Dimitrakopoulos R, Gamache M (2018) A New Computational Model of High-Order Stochastic Simulation Based on Spatial Legendre Moments. Math. Geosci. J. 50, 929–960 (2018). https://doi.org/10.1007/s11004-018-9744-z.

Zhang T, Switzer P, Journel J (2004) Sequential Conditional Simulation Using Classification of Local Training Patterns. In Leuangthong, O. & Deutsch, C. V. (Eds) Quantitative Geology and Geostatistics, 14, 265-274.

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