Computers and Geosciences 128 (2019) 41–50

Contents lists available at ScienceDirect

Computers and Geosciences

journal homepage: www.elsevier.com/locate/cageo

Efficient training image selection for multiple-point geostatistics via analysis of contours

Mohammad Javad Abdollahifard[a,*], Mohammad Baharvand[a], Grégoire Mariéthoz[b]

[a] Electrical Engineering Department, Tafersh University, Tafresh, 39518 79611, Iran
[b] Institute of Earth Surface Dynamics, Université de Lausanne, Lausanne, Switzerland

ARTICLE INFO

Keywords:
Texture classification
High-order consistency
Training image
Sensitivity analysis

ABSTRACT

Multiple-point statistics (MPS) methods have emerged as efficient tools for environmental modelling, however their efficiency highly depends on the availability of appropriate training images (TIs). We introduce an efficient method for selecting one compatible TI among a proposed set, based on a measure of compatibility with available conditioning data. While existing approaches to do this consider all available data-events in the simulation grid, we concentrate on a limited number of data-events around the contours and edges of the image. The proposed method is evaluated with different sampling rates, based on hundreds of sample sets extracted from binary, categorical and continuous images, and compared with exhaustive data-event extraction. Our experiments show that the proposed method improves the required CPU-time by up to two orders of magnitude and at the same time leads to a slight improvement in the recognition accuracy.

1. Introduction

In recent years, training-image (TI) based geostatistical methods have gained popularity for the simulation of environmental variables. Several algorithms have been introduced in the literature to cope with different challenges in multiple-point simulation, such as their heavy computational burden, the difficulty of preserving continuity of patterns, appropriate incorporation of conditioning data, and modelling realistic variability of environmental phenomena (Abdollahifard, 2016; Abdollahifard and Faez, 2014; Abdollahifard and Nasiri, 2017; Honarkhah and Caers, 2010; Mahmud et al., 2014; Mariethoz et al., 2010; Tahmasebi et al., 2014).

Before starting the MPS simulation process, it is essential to select a representative training image. This is a challenging problem, particularly when not enough information is available about the behavior of the field, e.g. subsurface variables. The training image is a conceptual model that allows the modeler to directly incorporate desired concepts on the physical process of interest (Maharaja, 2008). As noticed by different researchers, selection of an inadequate TI may lead to realizations incompatible with either observed data or real field variations (Abdollahifard and Ahmadi, 2016; de Almeida, 2010; Pyrcz et al., 2008). The subjective essence of the TI preparation process makes it necessary to quantitatively check the consistency of the TIs with observed data.

A limited number of researchers have addressed this problem. Some authors have used indirect state data through inverse methods to check the compatibility of the training images with dynamic outputs (Khodabakhshi and Jafarpour, 2013; Suzuki and Caers, 2008). Such approaches, however, are only limited to specific applications and typically require expensive forward model runs.

Another avenue is to check the compatibility of direct static data with the training image. This usually proceeds by extracting different data-events from the grid, searching the TI to find matches for each data-event, and then making the final decision based on the matches found. Eskandaridalvand (2008) suggests to extract data around each conditioning point in a spiral order and search the TI to find patterns with similar behavior (i.e. increase or decrease in the same manner). The consistency measure is computed based on the distribution of compatible training nodes. Pérez et al. (2014) employed a similar method for data-event extraction around each grid-point. They have defined a distance function between data-event and TI patterns. If the distance is below a small threshold, the training pattern is considered as a match. The number of matches are then used for computing a compatibility index. Weighted distance functions are also used by Feng et al. (2017).

In this paper a new data-event extraction method is suggested inspired by recent advances in the field of image processing. Studies on the human visual system revealed that while surface characteristics

* Corresponding author.
E-mail address: mj.abdollahi@tafreshu.ac.ir (M.J. Abdollahifard).

https://doi.org/10.1016/j.cageo.2019.04.004
Received 22 September 2018; Received in revised form 22 February 2019; Accepted 4 April 2019
Available online 09 April 2019
0098-3004/

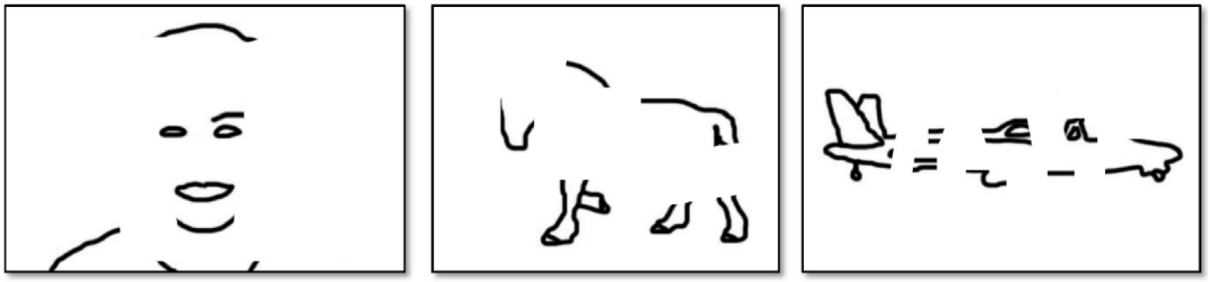

Fig. 1. People can recognize objects from partial contours (Shotton et al., 2008).

such as color, brightness and texture only play a secondary role, people primarily rely on edges and contours of the image for object recognition (Biederman and Ju, 1988; De Winter and Wagemans, 2004). It was observed that the recognition time and success rate for professionally photographed objects is identical to the simplified line drawings showing only the objects' major components. As another important observation, people are also capable of recognizing objects from partial contours, like those depicted in Fig. 1. On the other hand, computer algorithms designed based on edge and contour analysis show considerable success in object recognition (Shotton et al., 2008) and texture analysis (Ojala et al., 2002). Texture synthesis and image inpainting algorithms also show improvements in terms of continuity preservation by considering image edges during simulation (Abdollahifard, 2016; Criminisi et al., 2004). It has been shown that samples extracted around image edges are more informative and important (Abdollahifard et al., 2016).

Inspired by aforementioned facts, we have proposed an edge-based method for evaluation of multiple-point consistency of conditioning data with the TIs. Given the point conditioning data, we have developed a simple method for estimating the location of edges based on Delaunay triangulation (Guibas et al., 1992). Then, data events are extracted around edge points and TIs are sought to find matches for each data-event. The final decision regarding training image compatibility is made by comparing the number of data events matches in each TI. Ignoring flat data events, i.e. data events with no transition, improves the algorithm's speed significantly, and at the same time, leads to slight improvements in terms of recognition rate. To find matches for any given data-event, the TI is scanned exhaustively. It may be possible to further accelerate the TI selection process by employing approximate match-finding methods. However, such methods increase the speed at the expense of a reduced accuracy. Furthermore, the speedup achieved by our approach makes such optimization unnecessary for many practical purposes.

2. Methodology

In this paper, the key idea is to take into account the content of data-events and use only those data-events that convey helpful information. In the MPS literature, a *data-event* refers to a set of informed nodes extracted from the simulation grid (SG) in a neighborhood with limited spatial extent. In MPS simulation methods, data-events are often extracted by sliding fixed size templates on the simulation grid (Honarkhah and Caers, 2010; Kalantari and Abdollahifard, 2016; Mahmud et al., 2014; Pourfard et al., 2017; Sharifzadehlari et al., 2018). To handle both short and long-range variations within fixed size templates, simulation grids are usually processed hierarchically in multi-resolution pyramids. Mariethoz et al. (2010) suggest variable size data-events depending on the data density around each grid point. The n closest data are included in the data-event, provided that their spatial distance from the center is less than a predefined radius, r. Using a similar approach for TI-selection, Pérez et al. (2014) explore each neighborhood in a spiral order, to either collect a fixed number of data points or reach a maximum size.

In different MPS simulation and TI-selection methods, the contents of data-events often has no effect on the data-event extraction process. For simulation algorithms, it has been shown that prioritizing data-events based on their contents and giving higher importance to data-events with strong edges would result in improvements in pattern connectivity (Abdollahifard, 2016). In this paper, we focus on more informative data-events, which are drawn around edges and contours of the image, ignoring data-events in flat regions. Assuming that conditioning data are distributed sparsely in the field, the adjacency between them is defined using a Delaunay triangulation, and contour localization is performed by comparing adjacent pixels' values. A thresholding method is proposed for continuous variables, relying only on given conditioning data to highlight significant transitions. After approximate edge localization, data-events are only extracted along the edges and training image compatibility is computed based on comparing data-event with the given TIs.

2.1. Approximate contour localization

We assume that the conditioning data are given in a grid I in the form $\{(\boldsymbol{p}_i, z_i)|i = 1, ...,N_s\}$, where $\boldsymbol{p}_i \in \mathscr{R}^d$ denotes the location of point data and z_i denotes their corresponding values ($d \leq 3$). Note that our proposed contour localization works identically for any type of grid, including non-structured grids. For binary variables, $z \in \{0,1\}$, for categorical variables with n_C facies, $z \in \{0,1, ...,n_C - 1\}$, and for continuous variables $z \in \mathbb{R}_{[0,1]}$. Contours of an image are defined as the border between regions with different values. The contours of all variable types, including binary, categorical and continuous, can be identified by processing a set of binary grids as follows:

Binary Variables: The contours are defined as the transitioning locus between the two valid values, 0 and 1.

Categorical variables: Categorical grids with n_C categories can be expressed by n_C binary grids. The value of n-th binary grid is one at points belonging to the n-th category and zero at points belonging to other categories. For the n-th binary grid a set of contours, denoted by γ_n, is defined as the locus between the n-th category and the remaining categories ($n = 1, ...,n_C$).

Continuous variables: Because continuous variables can take an infinite number of values, their contours are intractable in their original form. By using a set of thresholds, continuous variables are partitioned into categories and only the contours between the identified categories are considered in the remaining steps of the algorithm. Thresholds are automatically drawn from the data by emphasizing on their sharp transitions (see section 2.3).

In a complete image, a pixel is assumed on the edge if its value is (significantly) different from the value of some of its adjacent pixels. However, since the point-data are usually distributed sparsely in our application, it is impossible to precisely localize the edges. To define adjacency, the points, $\boldsymbol{p}_i$s, are triangulated and the resulting graph is denoted by G. Two points, $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$, are considered adjacent if they are connected by a link, l_{ij}, in the graph.

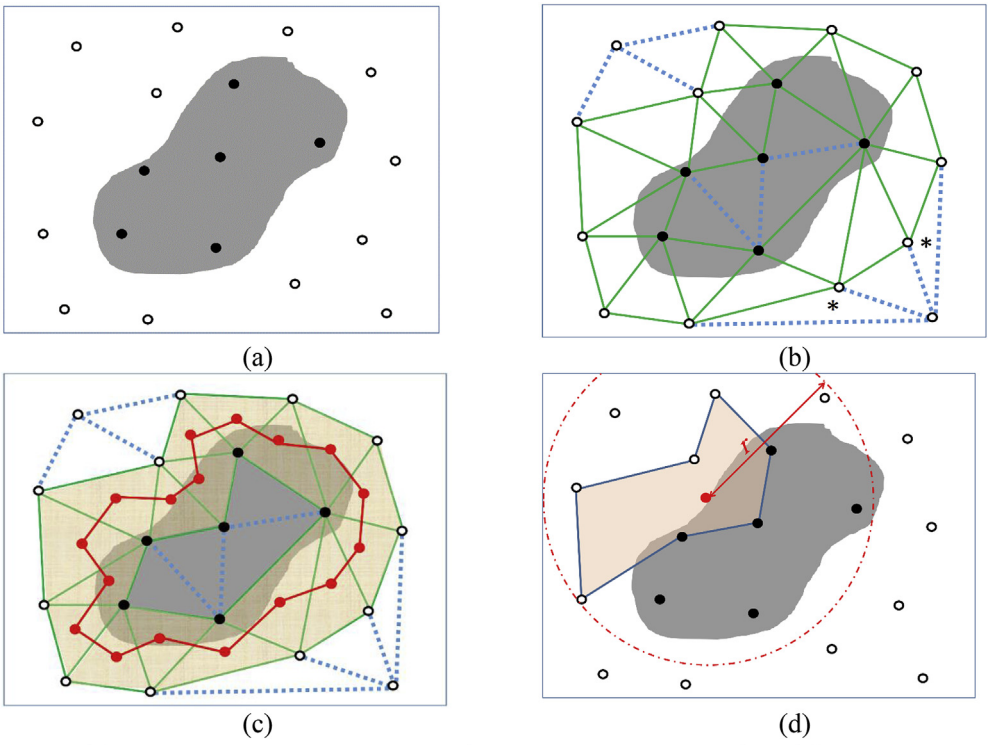

Fig. 2. (a) Some conditioning-data extracted from a hypothetical binary field, (b) Delaunay triangulation. G' (depicted with solid green lines) is a subgraph of G containing only transitioning triangles. G' Encompasses the image contour. (c) The dual graph G'' depicted with solid red lines. (d) An example data-event having $N = 5$ triangles. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

Based on the above discussion, without loss of generality, we assume that the samples are drawn from a binary field, as depicted in Fig. 2 a. The link, l_{ij}, is called transitioning if its endpoints have different values ($v_i \neq v_j$). At some point along any transitioning link, at least one transition has happened. In other words, the contours are delimited by transitioning links. A triangle in the graph, G, is called transitioning if it contains at least one transitioning link. Since the variable is binary, any transitioning triangle will have exactly two transitioning links. The pruned graph G' is formed from G by keeping only transitioning triangles, as depicted in Fig. 2 b by solid green links. The contour of the image lies completely inside G' (Fig. 2 c).

The dual graph of G', denoted by G'', is a graph whose nodes correspond to the faces (triangles) of G'. Two nodes are connected by a link in G'' if their corresponding triangles in G' share a transitioning link. The dual graph is depicted with solid red lines in Fig. 2 c. Each node in G'' has two connected links, except near the borders of the image where no more triangles exist. Then, G'' is composed of disjoint loops and/or open curves.

2.2. Data-event extraction

As mentioned before, our main contribution is to propose a new method for extracting more informative data-events with a data-driven support. One data-event is extracted around each node of the dual graph. On either side of the node, we find $(N-1)/2$ connected nodes if available, forming a N-node path (N is an odd integer). The vertices of N corresponding triangles in G' are considered as data-event, provided that they lie within a predefined radius, r, from the center. Hence, data-events include maximum of $N+2$ nodes. As depicted in Fig. 2 d, the data-event support has an irregular shape aligned along the image's contour.

For categorical variables, the contour approximation stage is independently repeated for each category by forming a different pruned graph and dual graph based on the corresponding binary grid. Then, data-events are extracted around each contour independently. Note that one of the contours is redundant, i.e. completely included in union of other contours. Hence, in the data-event extraction phase we ignore one of the contours and work only based on the remaining ones. We have tested a range of ideas for data-event extraction concluding that the above-mentioned method is the most efficient one. In our tests, data-events were drawn around the same nodes, but the difference was in the support of data-events. Apart from the proposed method, we also tested square and disk-shaped data-events with and without giving higher weights to nodes connected to transitioning links.

While the triangles tend to be nearly equilateral in the middle of the grid, points near the borders of the image are prone to form obtuse triangles having long sides (see for example the triangles marked by stars in Fig. 2 b). As noted before, if two nodes are connected by a link they will be considered adjacent. Assuming that two far away points are adjacent does not make sense and leads to too rough contour localization and very large-extent data-events. To prevent this, the graph is simplified by removing too long links on the boundary of the graph.

Although extending the method to 3D variables is not examined in this paper, it can proceed in a similar way. After partitioning the 3D domain in a set of tetrahedrons using the Delaunay method, the transitioning tetrahedrons can be identified as tetrahedrons having at least one transitioning link. These tetrahedrons are considered as nodes of the dual graph. Two nodes in the dual graph are connected with a link if their corresponding tetrahedrons share a transitioning triangle. Since the grid is assumed binary, each node of the dual graph has three or four connected neighbors and the dual graph is composed of a set of closed surfaces. The data-events can be extracted on these surfaces.

For geological applications, direct data are usually extracted from deep boreholes in 3D fields. In such cases, while high-resolution data in available along the holes, different holes are located distantly. In other words the samples are distributed very inhomogeneously in different directions. This can have an adverse effect on the triangulation. A possible solution to this problem could be to stretch the field in one

direction so that the density becomes the same in all directions.

2.3. Thresholding for continuous variables

Continuous variables can take many different values. In order to detect only strong edges, we clustered the image values to n_C categories using $n_C - 1$ thresholds.

To select the thresholds, at first we associate a value, z_{ij}, and an approximate derivative, d_{ij}, with each link, l_{ij}, of the graph G. The value is computed as the average of the endpoints values:

$$z_{ij} = \frac{z_i + z_j}{2}. \tag{1}$$

The derivative is also defined as:

$$d_{ij} = \frac{|z_i - z_j|}{\|\boldsymbol{p}_i - \boldsymbol{p}_j\|}. \tag{2}$$

where $\|\boldsymbol{p}_i - \boldsymbol{p}_j\|$ is the Euclidean spatial distance between the two endpoints. We form a function by adding for each link a Gaussian function centered at the link value and weighted by the derivative magnitude:

$$h(z) = \sum_{l_{ij} \in G} d_{ij} e^{-\frac{(z - z_{ij})^2}{2\sigma_{ij}^2}}. \tag{3}$$

where $\sigma_{ij} = |z_i - z_j|/4$. Local maxima of the above function are considered as thresholds, after removing too close maxima and maxima with values less than 10% of the global maximum. Obviously, links with strong derivatives (fast transitions) have more impact on h and their values are more likely to be selected as thresholds. Using the selected thresholds, each point is assigned to a cluster depending on its value.

It should be noted that the formed categorical grid is used only for (strong) contour localization, and data-events are formed from the original values taken in the continuous grid.

2.4. Compatibility evaluation

In this paper, we have adopted the compatibility evaluation method of Pérez et al. (2014), which is briefly explained in the following. Assuming M candidate training images, $\boldsymbol{TI} = \{TI_1. \ldots. TI_M\}$, we need to designate the most compatible one with the given conditioning data in the grid I. Let us denote the k-th conditioning data event with $\boldsymbol{CE}_k$ ($k = 1. \ldots K$). The TIs are searched for matching data-events. A pattern is considered as a match for $\boldsymbol{CE}_k$ if its distance is below a threshold ε. The relative compatibility of the k-th event with the m-th TI is defined as:

$$p_{k,m} = \frac{\mathscr{M}_{k,m}}{\sum_{m=1}^{M} \mathscr{M}_{k,m}}. \tag{4}$$

where $\mathscr{M}_{k.m}$ denotes the number of matches for $\boldsymbol{CE}_k$ in TI_m. The relative compatibility of all conditioning data in I with the m-th TI is defined as:

$$C_m = \frac{\sum_{k=1}^{K} p_{k,m}}{\sum_{m=1}^{M} \sum_{k=1}^{K} p_{k,m}}. \tag{5}$$

Before proceeding to the experiments, we attempt to clarify the process of the proposed method and highlight its advantages through a simple example. Fig. 3 a shows a sparsely distributed sample set from a binary filed. Our goal is to decide the compatibility of the TIs shown in Fig. 3 c and d with the given samples. Visually, it seems that the tear-shaped object shown in TI_2 is more compatible with the samples.

To quantify compatibility, the proposed method proceeds by extracting data-events around transitioning edges of the Delaunay triangulation graph. Two of such data-events are shown enclosed in red and green polygons in Fig. 3 b ($N = 5$ and $r = \infty$). The third data-event, enclosed in the blue polygon, is extracted in a textureless area (where all pixels have of the same black color). Such a data-event will not be considered in the proposed process and here we consider it only for comparison. Let us evaluate the compatibilities only based on these three data-events. Central pixels of patterns in the TIs that exactly match these data-events are marked with its associated color in Fig. 3 c and d. The match counts, compatibility factors of data-events with the TIs, and accumulative compatibility factors are reported in Fig. 3 e. While compatibilities of the first two data-events are consistent with our subjective judgment, the last one's indication is different. As shown in this example, data-events in textureless areas may not carry meaningful information. Ignoring them can reduce the computational cost.

3. Results and discussion

In this section, the proposed method is applied to binary, categorical and continuous fields. We have compared our method with exhaustive and fast search methods in terms of recognition accuracy and computational time. In the following two sub-sections, the important aspects of alternative methods and the strategies used for a fair comparison are delineated.

3.1. Alternative methods used for comparison

In all of our experiments, we have compared our method with an *exhaustive* method that extracts all possible data-events in the sampling grid. Except for the data-event extraction phase, all other phases of both methods are implemented identically. Since the number of neighboring nodes in the dual graph N, cannot be defined in the exhaustive method, we set it to infinity in our method. In other words, in both methods the points are delimited only by their distance from the center, r, not by their count. However as will be shown, the results reported for a range of r show that for lower sampling rates the best results are obtained using larger radiuses and vice versa. This justifies the use of the parameter N in practice.

For the case of binary images, two faster alternatives for the exhaustive method are also examined. The first strategy, termed as *partial 1%*, is to select a random subset containing 1% of all possible data-events and ignore the remaining data-events during compatibility evaluation. This strategy reduces the computational cost to a level comparable to our proposed method. However, it has a negative effect on the accuracy, as will be reported shortly.

Another alternative suggested by Pérez et al. (2014) is to search the TIs using a direct sampling search strategy (Mariethoz et al., 2010). This approach is termed here *DS-based* method. For each conditioning data-event, instead of completely scanning all TIs, the TIs are scanned in a random path which goes through different TIs. After finding the first match, the search is stopped and the counter corresponding to the TI in which the match is found is incremented. After doing this for all data-events, the compatibility is computed as follows:

$$C_m = \frac{L_m}{\sum_{m=1}^{M} L_m}, \tag{6}$$

where L_m is the counter associated with the m-th TI. This strategy leads to a significant reduction in the number of TI positions to be scanned, but its random search path has a negative effect on the speed.

The template-matching phase of the proposed, exhaustive and partial methods are all accomplished efficiently without any explicit loops using MATLAB built-in functions. The DS-based method, however, must inevitably be implemented via loops. Since loops are intrinsically slower than built-in functions in MATLAB, we normalized the CPU-time of the DS-based method to the time of a loop-based exhaustive implementation. Multiplying the obtained value by the time of the exhaustive implementation, an extrapolated time is computed for the DS-based method in a hypothetic programming language that can perform loop-based search as fast as built-in functions of MATLAB in a time

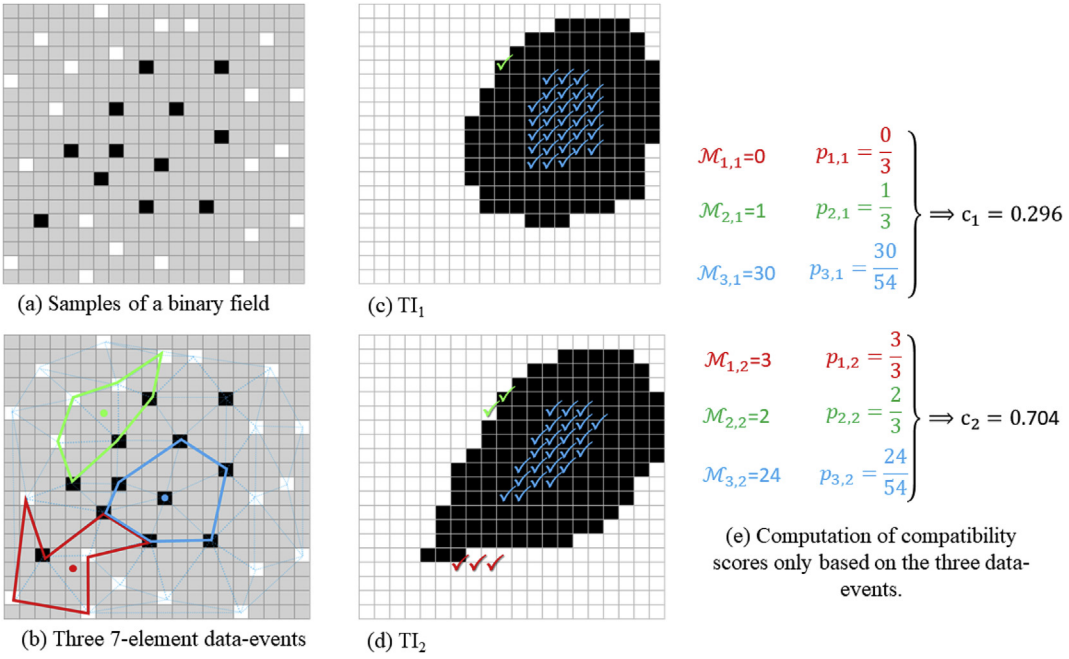

Fig. 3. (a) Is a hypothetical sample set extracted from a binary field. (b) Only three data-events enclosed in solid colored polygons are used for compatibility evaluation. The red and green ones are extracted based on the proposed method near the edges and the blue one is an ordinary data-event extracted from a textureless area. (c) and (d) are two hypothetical TIs. Visually, the second one (fig. d) seems more compatible with the observed samples. The central pixels of matched patterns are marked with the same color used for data-events. (e) Shows the compatibility computation process. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

proportional solely to the required calculations.

3.2. Evaluation strategy

For evaluation of methods, the following strategy is used. A set of images are considered as possible TIs. For the m-th TI, an image with similar behavior, denoted by I_m, is used as the reference field from which conditioning data are extracted. Except for the continuous case, all training images and reference fields used in the same test have the same facies proportions so that they cannot be distinguished based on one-point statistics. N_s samples are extracted from each field using stratified sampling, i.e. by dividing the field to N_s rectangular regions with similar size and drawing a random sample from each region. Then, the compatibility factor of the sampled image is computed with all TIs using equation (5). Obviously, the samples drawn from I_m should be more compatible with TI_m. Let us denote with $C_{m,n}$ the compatibility of samples drawn from I_m with TI_n. Then $C_{m,m}$ should be larger than $C_{m,n}$ for all $n \neq m$. If the computed factors satisfy this, we interpret it as a correct decision. For each I_m, 100 different sets of samples are extracted all with the same number of samples, N_s, but drawn from different random locations. The decision making accuracy, A, is defined as the ratio between the number of true decisions to the number of all decisions made. All experiments are performed in MATLAB environment on a laptop computer with Intel(R) Core(TM) i7-6700HQ CPU @2.6 GHz, and 12 GB of RAM.

3.3. Binary variables

The first set of images used in our tests is depicted in Fig. 4. The images are synthesized using TiGenerator in SGeMS (Maharaja, 2008) with different parameter settings. The images are taken from Pérez et al. (2014) and the reader can find further details on their generation therein. In each test, 300 sample sets are extracted using stratified sampling (100 from each I_m). The distance threshold, ε, is set to zero. The experiments are repeated on the same samplings for both proposed and exhaustive method with different rs. The tests are performed for different number of samples $N_s = 36,\ 81,\ 225$ and 400.

The obtained results are shown in Table 1. t_{av} is the average CPU time required for each decision making. As expected, the proposed method is much faster than the exhaustive method. The best results in terms of accuracy are shown in boldface. By comparing the average times of these results, we can observe that the proposed method is faster than the exhaustive method by factors in the range of 25–170. The reason is that the proposed method only considers a very limited set of data-events. The ratio of the number of data-events used for decision making to the number of all grid points (either informed or uninformed) is denoted by p_{dev} and reported in Table 1. p_{dev} does not depend on parameters (r and N) and ranges from 0.31% to 3.55% for our algorithm in this experiment. While the number of data-events drawn from the grid depends on the number of samples in the proposed method, the exhaustive method extracts nearly the same number of data-events independent of the sampling density.

One may worry about the preprocessing time, including the time required for finding Delaunay triangulation, pruning the graph, forming dual graph and identifying adjacent nodes in it. Our measurements show that in this example, the ratio of the preprocessing time to the total implementation time of the proposed algorithm ranges from 2.2% to 4%. This indicates that the preprocessing time is almost negligible and match-finding is still the most computationally demanding phase.

For $N_s = 36$ and 225, our best accuracy is better than the exhaustive method and for $N_s = 400$ our method presents the maximum accuracy of 100% for a wider range of changes in radius r. Only for $N_s = 81$ our results are worse than with the exhaustive search. Another important observation is that for lower sampling rates, the best results are obtained using larger radiuses. In other words, for being sufficiently distinctive, a data-event should include a minimum number of data points. Obviously, for lower sampling rates, such data have larger spatial extents.

For this set of images, partial and DS-based methods are also tested. As shown in Table 1, by using the *partial* method, the CPU-time improves significantly and becomes comparable to our results. This method, however, leads to a considerable accuracy reduction. The

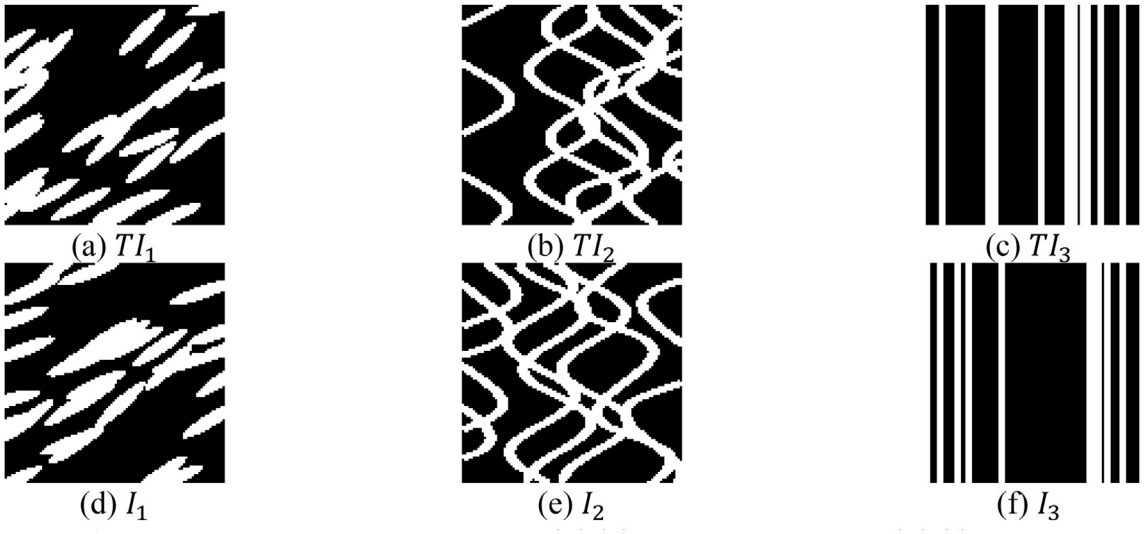

Fig. 4. A set of synthetic 100 × 100 binary images, (a)–(c) are used as TIs and (d)–(f) are used as sampling fields. The images are taken from (Pérez et al., 2014).

Table 1
Comparison of the accuracy and required CPU-time for decision making about samples drawn from binary images of Fig. 4 d-f.

	r	36 samples		81 samples		225 samples		400 samples	
		A(%)	t_{av}(s)	A (%)	t_{av}(s)	A (%)	t_{av}(s)	A (%)	t_{av}(s)
Our method	7					98.67	0.25	100	0.44
	9	57.33	0.06	78.33	0.10	99.00	0.25	100	0.48
	12	57.00	0.06	82.00	0.10	99.00	0.28	100	0.58
	15	58.33	0.06	80.33	0.11	98.67	0.34	100	0.71
	20	57.33	0.06	74.33	0.13	95.67	0.46	100	1.01
	25	57.33	0.06	67.67	0.17				
p_{dev}(%):		0.31		0.77		2.17		3.55	
Exhaustive	7					97.67	10.36	100	10.29
	9	46.33	4.61	80.00	12.11	98.00	10.42	100	10.38
	12	54.67	7.26	84.00	12.53	98.00	11.83	100	12.81
	15	57.33	10.17	81.67	14.46	88.33	13.15	71.33	14.67
	20	57.00	13.97	71.33	17.30				
p_{dev}(%):		100		100		100		100	
Partial 1%	7					95.00	0.10	99.67	0.10
	9	42.00	0.06	71.67	0.10	97.66	0.10	100	0.10
	12	47.00	0.08	78.67	0.11	95.67	0.13	99.67	0.13
	15	53.33	0.11	77.33	0.13	83.67	0.14	69.67	0.15
	20	54.00	0.14	65.33	0.15				
p_{dev}(%):		1		1		1		1	

Table 2
The recognition rate for the DS-based search on the binary images of Fig. 4. t_{ds} shows the average time for DS-based search and t_{el} is the average time of exhaustive loop-based search. t_{dsnl} is the extrapolated DS time.

	r	36 samples			81 samples			225 samples			400 samples		
		A(%)	t_{ds}/t_{el}	t_{dsnl}	A (%)	t_{ds}/t_{el}	t_{dsnl}	A (%)	t_{ds}/t_{el}	t_{dsnl}	A (%)	t_{ds}/t_{el}	t_{dsnl}
DS-based	7	34.67	2.64/609					97.33	5.49/774	0.07			
	9	41.00	3.78/742	0.02	75.33	4.78/786	0.07	98.00	11.75/865	0.14	100.0	250.1/1020	2.55
	12	45.67	4.62/809	0.04	81.33	5.39/852	0.08	98.00	173.6/923	2.23			
	15	55.00	4.80/887	0.06	81.67	12.25/937	0.19						
	20	58.33	6.23/970	0.09	71.67	169/1003	2.91						
	25	56.00	24.0/1065										

results of DS-based search are reported in Table 2. In this table, t_{ds} denotes the CPU-time of the DS-based method, and t_{el} and t_{enl} show the time required for exhaustive implementations with and without explicit loops, respectively. The extrapolated DS time is calculated as $t_{dsnl} = t_{ds}t_{enl}/t_{el}$.

Considering only the best results highlighted by boldface numbers, the DS-based method results in speed-up factors of 4–150 times. For large data-events (larger r) the average time of DS search increases dramatically because it becomes more difficult to find a match. By comparing the accuracy of the DS-based method with that of the exhaustive method reported in Table 1, we observe that for lower r, where many matches can be found, relying only on the first match

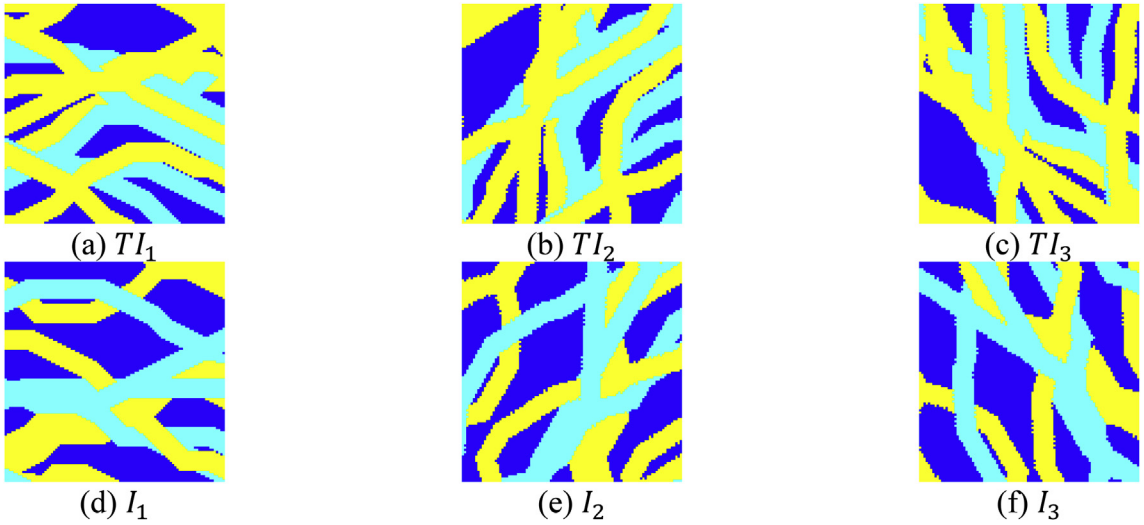

Fig. 5. Categorical 100 × 100 images used in the second test. The images of second and third columns are obtained by rotating their corresponding image in column one by 60 and 120° respectively.

significantly reduces the accuracy. However, for larger r, the accuracy of both methods are comparable. The extrapolated DS time is comparable to the CPU-time of our proposed method, but the latter outperforms in terms of recognition accuracy.

In this experiment and the following ones, we have used a large number of sample-sets. Processing such amount of sets with different methods and different settings is very time-consuming. Hence, we have only included the tests that seem essential for understanding trends and behaviors and avoided redundant ones. This explains that some cells are empty in the tables presenting the results.

3.4. Categorical variables

Another test is conducted on the categorical images shown in Fig. 5. The images show fixed width meandering channels with different orientations. In fact, TI_2 and I_2 are obtained by rotating TI_1 and I_1 by 60°, and TI_3 and I_3 are also obtained by applying a 120° rotation. Once again for each N_s, 100 sample sets are extracted from each I_m. ε is set to zero in all experiments. Then the compatibility of samples with the given TIs is evaluated using both the proposed method and the exhaustive method, and the results are shown in Table 3. The trends are similar to that of the binary case. While our results are comparable with the exhaustive method in terms of accuracy, our method has performed 15 to 105 times faster.

3.5. Continuous variables

The next test is performed on 20 sub-images extracted from a large satellite image of Lena Delta in Russia, as shown in Fig. 6. Although not all sub-images have similar histograms, sub-images with similar histograms are included to make it impossible to decide based only on one-point statistics. Once again, for each N_s 100 sample sets are extracted from each I_m (total of 1000 sample set) and the compatibility of each set with the TIs is evaluated using both the proposed and the exhaustive method. In this experiment, ε is set to 5%, allowing matches to have slight deviations from the data-events.

The results are shown in Table 4. Increasing the number of TIs and sampling fields leads to a significant increase in the run time. Particularly for the exhaustive case, computing each entry of the table requires 5–8 h of run. The previous experiments show that the best results of both methods are obtained with similar settings for r and hence, here we have computed the accuracy for exhaustive method only for the radius that maximizes the accuracy of our method. Our algorithm performs faster by a factor of 20–125. For $N_s = 36$, while the best

Table 3
Comparison of the accuracy and CPU-time of the proposed method versus exhaustive method for categorical images of Fig. 5.

	r	36 samples		81 samples		225 samples		400 samples	
		A(%)	t_{av}(s)	A (%)	t_{av}(s)	A (%)	t_{av}(s)	A (%)	t_{av}(s)
Our method	5	52.33	0.09	82.00	0.16	99.67	0.37	100	0.58
	7	52.33	0.09	82.33	0.16	99.67	0.40	100	0.64
	9	52.00	0.09	81.67	0.16	100.0	0.43	100	0.72
	12	53.00	0.09	79.00	0.16	99.67	0.49	100	0.87
	15	54.00	0.09	82.67	0.18	94.00	0.69	96.67	1.06
	20	58.67	0.09	77.67	0.22	85.00	0.94	90.33	1.27
	25	52.67	0.10	73.00	0.27	84.00	1.03		
p_{dev}(%):		0.68		1.53		2.99		3.15	
Exhaustive	5							100	9.12
	7					100	9.11	100	9.25
	9			81.67	8.06	99.00	9.52	100	9.66
	12	55.00	7.23	83.33	10.32	88.00	10.56		
	15	58.00	9.45	80.67	13.67				
	20	57.00	11.97	70.00	15.18				
	25	54.67	15.57						
p_{dev}(%):		100		100		100		100	

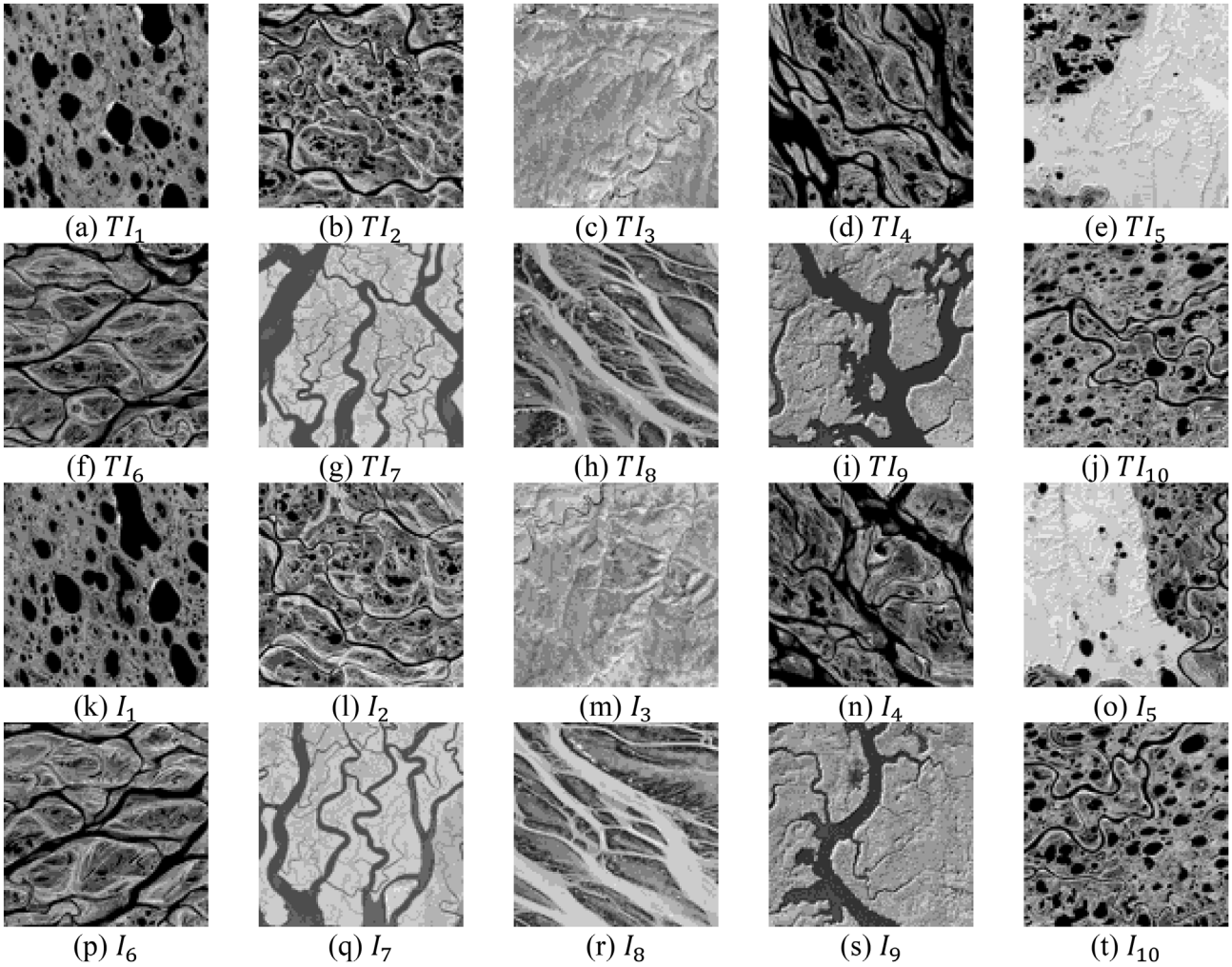

Fig. 6. 100 × 100 continuous images used in the third test. Images are extracted from a large satellite image of the Lena Delta, Russia. For all m, TI_m and I_m are extracted from neighboring regions in the original image.

performance corresponds to $r = 15$ and 20 in the binary and categorical examples, here $r = 12$ leads to the best results. This is because of higher curvatures of the images contours in the continuous case.

As mentioned before, our method automatically partitions each sample set to a number of categories. Among 1000 sample sets used in this experiment, 598 are divided into two categories, 296 into three, 90 into four, 13 into five, and the remaining 3 into six.

3.6. Nonstationary variables

Here, we have tested the proposed method on the set of nonstationary images shown in Fig. 7. The images are obtained with Direct Sampling and the method of elementary images. The elementary training image consists in horizontal lines. Rotation and affinity fields are designed such that the realization are symmetric (Mariethoz and Kelly, 2011). The images are then divided into two halves. The left halves are used as TIs and the right halves are used as sampling fields

Table 4
Comparison of the accuracy and CPU-time of the proposed method versus exhaustive method for continuous images of Fig. 6.

	r	36 samples		81 samples		225 samples		400 samples	
		A(%)	t_{av}(s)	A (%)	t_{av}(s)	A (%)	t_{av}(s)	A (%)	t_{av}(s)
Our method	5							86.50	1.45
	7					86.40	0.72	89.70	1.48
	9	56.40	0.14	68.50	0.29	87.00	0.77	88.80	1.57
	12	56.90	0.15	68.90	0.30	83.60	0.86	75.20	1.82
	15	56.80	0.15	68.30	0.33	76.00	1.02	62.60	2.18
p_{dev}(%):		0.49		1.14		1.98		2.77	
Ex.	7							87.90	28.72
	9					87.70	24.15		
	12	52.50	18.80	67.90	23.07				
p_{dev}(%):		100		100		100		100	

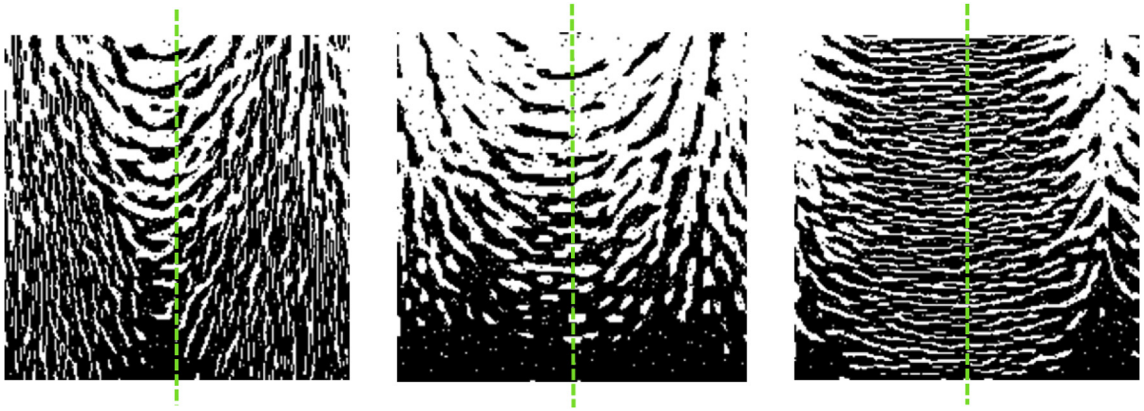

Fig. 7. The images of size 200×200 used to test our method in non-stationary cases. These images are divided in two halves. The left halves are used as TIs and the right halves are used as sampling fields after being mirrored around the dashed lines.

Table 5
Comparison of the accuracy and CPU-time of the proposed method versus exhaustive method for non-stationary images of Fig. 7.

	r	128 samples		200 samples		450 samples		800 samples	
		A(%)	t_{av}(s)	A(%)	t_{av}(s)	A (%)	t_{av}(s)	A (%)	t_{av}(s)
Our method	15			48.33	0.32	63.33	0.85	75.33	2.04
	20	43.67	0.19	54.67	0.37	73.33	0.98	90.67	2.51
	25	53.67	0.24	62.67	0.42	78.33	1.14	88.00	2.65
	30	57.67	0.26	66.33	0.47	75.67	1.29		
	40	58.67	0.30	60.67	0.57				
	50	53.33	0.36	55.67	0.64				
p_{dev}(%):		0.63		1.01		2.41		4.32	
Ex.	20					57.00	41.62	60.67	47.15
	25			62.00	43.47	61.67	50.37	70.33	62.12
	30	60.00	46.79	59.00	53.23	72.67	62.39	76.67	81.85
	40	61.67	65.09						
p_{dev}(%):		100		100		100		100	

after being mirrored around the dashed lines. Since the general behavior of all three images is by construction similar, the TI selection problem is more challenging in this example. The recognition rate of both the proposed method and the exhaustive method are reported in Table 5 for $N_s = 128$, 200, 450 and 800. Once again the proposed method shows higher speed and better recognition rate.

4. Conclusion

In this paper, we have proposed an efficient method for training image selection in multiple-point geostatistics. Instead of extracting all possible data-events, data are drawn only around strong edges of the image, resulting in a significant speedup of up to two orders of magnitude. To define edges in a grid with scattered static data, the adjacency is defined based on Delaunay triangulation. Data-events are extracted from subsequent transitioning links i.e. the links with highly different values at the endpoints. The proposed method is evaluated on hundreds of sample sets extracted from binary, categorical and continuous images and compared with exhaustive data-event extraction. Evaluation is performed for different sampling rates and parameter settings. In most cases, the proposed method outperforms the exhaustive method in terms of recognition accuracy.

The experiments reveal that for obtaining the best results, the spatial extent of data-events should be considered larger for lower sampling rates. Furthermore, the best choice for the spatial extent of data-events depends on the behavior of the fields. While, smaller extents are sufficient for images with highly curved contours, recognizing images with contours having lower curvatures requires larger data-events. It should be noted that the proposed method is performant when the TIs cannot be distinguished based on one-point statistics (facies proportions or histograms). Otherwise, elimination of data that are far from image contours may even lead to performance reduction. Extending the method to scattered data in 3D fields is straightforward. However, adjustments may be required for handling data extracted densely along boreholes.

5. Computer code availability

Computer code is freely available for academic purposes at: https://github.com/abdollahifard/Contour_Analysis-for-TI-Selection.

Name of code: CATIMPS (Contour Analysis for TI Selection in MPS), Developer: Mohammad Javad Abdollahifard (e-mail: mj.abdollahi@tafreshu.ac.ir), Year first available: 2018, Programming Language: MATLAB (Release 2013a or higher).

Authorship Statement

M.J. Abdollahifard proposed the methodology, implemented the code and wrote the paper. M. Baharvand tested a variety of alternative methods and did the experiments. G. Mariethoz critically reviewed the paper, provided data and suggested additional experiments and comparisons.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.cageo.2019.04.004.

References

Abdollahifard, M.J., 2016. Fast multiple-point simulation using a data-driven path and an efficient gradient-based search. Comput. Geosci. 86, 64–74.

Abdollahifard, M.J., Ahmadi, S., 2016. Reconstruction of binary geological images using analytical edge and object models. Comput. Geosci. 89, 239–251.

Abdollahifard, M.J., Faez, K., 2014. Fast direct sampling for multiple-point stochastic simulation. Arab. J. Geosci. 7, 1927–1939.

Abdollahifard, M.J., Mariethoz, G., Pourfard, M., 2016. Improving in situ data acquisition using training images and a Bayesian mixture model. Comput. Geosci. 91, 49–63.

Abdollahifard, M.J., Nasiri, B., 2017. Exploiting transformation-domain sparsity for fast query in multiple-point geostatistics. Comput. Geosci. 21, 289–299.

Biederman, I., Ju, G., 1988. Surface versus edge-based determinants of visual recognition. Cogn. Psychol. 20, 38–64.

Criminisi, A., Pérez, P., Toyama, K., 2004. Region filling and object removal by exemplar-based image inpainting. IEEE Trans. Image Process. 13, 1200–1212.

de Almeida, J.A., 2010. Stochastic simulation methods for characterization of lithoclasses in carbonate reservoirs. Earth Sci. Rev. 101, 250–270.

De Winter, J., Wagemans, J., 2004. Contour-based object identification and segmentation: stimuli, norms and data, and software tools. Behav. Res. Methods Instrum. Comput. 36, 604–624.

Eskandaridalvand, K., 2008. Growthsim: a Complete Framework for Integrating Static and Dynamic Data into Reservoir Models. University of Texas at Austin.

Feng, W., Wu, S., Yin, Y., Zhang, J., Zhang, K., 2017. A training image evaluation and selection method based on minimum data event distance for multiple-point geostatistics. Comput. Geosci. 104, 35–53.

Guibas, L.J., Knuth, D.E., Sharir, M., 1992. Randomized incremental construction of Delaunay and Voronoi diagrams. Algorithmica 7, 381–413.

Honarkhah, M., Caers, J., 2010. Stochastic simulation of patterns using distance-based pattern modeling. Math. Geosci. 42, 487–517.

Kalantari, S., Abdollahifard, M.J., 2016. Optimization-based multiple-point geostatistics: a sparse way. Comput. Geosci. 95, 85–98.

Khodabakhshi, M., Jafarpour, B., 2013. A Bayesian mixture-modeling approach for flow-conditioned multiple-point statistical facies simulation from uncertain training images. Water Resour. Res. 49, 328–342.

Maharaja, A., 2008. TiGenerator: object-based training image generator. Comput. Geosci. 34, 1753–1761.

Mahmud, K., Mariethoz, G., Caers, J., Tahmasebi, P., Baker, A., 2014. Simulation of Earth textures by conditional image quilting. Water Resour. Res. 50, 3088–3107.

Mariethoz, G., Kelly, B.F., 2011. Modeling complex geological structures with elementary training images and transform-invariant distances. Water Resour. Res. 47.

Mariethoz, G., Renard, P., Straubhaar, J., 2010. The direct sampling method to perform multiple-point geostatistical simulations. Water Resour. Res. 46.

Ojala, T., Pietikainen, M., Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. 24, 971–987.

Pérez, C., Mariethoz, G., Ortiz, J.M., 2014. Verifying the high-order consistency of training images with data for multiple-point geostatistics. Comput. Geosci. 70, 190–205.

Pourfard, M., Abdollahifard, M.J., Faez, K., Motamedi, S.A., Hosseinian, T., 2017. PCTO-SIM: multiple-point geostatistical modeling using parallel conditional texture optimization. Comput. Geosci. 102, 116–138.

Pyrcz, M.J., Boisvert, J.B., Deutsch, C.V., 2008. A library of training images for fluvial and deepwater reservoirs and associated code. Comput. Geosci. 34, 542–560.

Sharifzadehlari, M., Fathianpour, N., Renard, P., Amirfattahi, R., 2018. Random partitioning and adaptive filters for multiple-point stochastic simulation. Stoch. Environ. Res. Risk Assess. 32, 1375–1396.

Shotton, J., Blake, A., Cipolla, R., 2008. Multiscale categorical object recognition using contour fragments. IEEE Trans. Pattern Anal. Mach. Intell. 30, 1270–1281.

Suzuki, S., Caers, J., 2008. A distance-based prior model parameterization for constraining solutions of spatial inverse problems. Math. Geosci. 40, 445–469.

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. Comput. Geosci. 67, 75–88.