Computers & Geosciences 52 (2013) 307–324

Contents lists available at SciVerse ScienceDirect

Computers & Geosciences

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

A practical guide to performing multiple-point statistical simulations with the Direct Sampling algorithm

Eef Meerschman [a,*], Guillaume Pirot [b], Gregoire Mariethoz [c], Julien Straubhaar [b], Marc Van Meirvenne [a], Philippe Renard [b]

[a] Research Group Soil Spatial Inventory Techniques, Department of Soil Management, Faculty of Bioscience Engineering, Ghent University, Coupure 653, Gent 9000, Belgium
[b] Centre of Hydrogeology and Geothermics, University of Neuchâtel, Rue Emile Argand 11, CH-2000 Neuchâtel, Switzerland
[c] National Centre for Groundwater Research and Training, School of Civil and Environmental Engineering, The University of New South Wales, Sydney, NSW 2052, Australia

ARTICLE INFO

Article history:
Received 23 February 2012
Received in revised form
20 September 2012
Accepted 22 September 2012
Available online 1 November 2012

Keywords:
Multiple-point statistics
Direct Sampling algorithm
Sensitivity analysis
Training image

ABSTRACT

The Direct Sampling (DS) algorithm is a recently developed multiple-point statistical simulation technique. It directly scans the training image (TI) for a given data event instead of storing the training probability values in a catalogue prior to simulation. By using distances between the given data events and the TI patterns, DS allows to simulate categorical, continuous and multivariate problems. Benefiting from the wide spectrum of potential applications of DS, requires understanding of the user-defined input parameters. Therefore, we list the most important parameters and assess their impact on the generated simulations. Real case TIs are used, including an image of ice-wedge polygons, a marble slice and snow crystals, all three as continuous and categorical images. We also use a 3D categorical TI representing a block of concrete to demonstrate the capacity of DS to generate 3D simulations. First, a quantitative sensitivity analysis is conducted on the three parameters balancing simulation quality and CPU time: the acceptance threshold t, the fraction of TI to scan f and the number of neighbors n. Next to a visual inspection of the generated simulations, the performance is analyzed in terms of speed of calculation and quality of pattern reproduction. Whereas decreasing the CPU time by influencing t and n is at the expense of simulation quality, reducing the scanned fraction of the TI allows substantial computational gains without degrading the quality as long as the TI contains enough reproducible patterns. We also illustrate the quality improvement resulting from post-processing and the potential of DS to simulate bivariate problems and to honor conditioning data. We report a comprehensive guide to performing multiple-point statistical simulations with the DS algorithm and provide recommendations on how to set the input parameters appropriately.

1. Introduction

Multiple-point statistics (MPS) covers an ensemble of sequential simulation algorithms using a training image (TI) as input data for the spatial structure of a process instead of a two-point variogram (Guardiano and Srivastava, 1993; Strebelle, 2000). A TI is a conceptual image of the expected spatial structure and is often built based on prior information. Using a TI allows extracting multiple-point statistics and hence describing more complex patterns; this is especially important when spatial connectivity plays a key role in the model application (Bianchi et al., 2011; Gómez-Hernández and Wen, 1998; Renard and Allard, 2012; Zinn and Harvey, 2003).

As is characteristic for sequential simulations, the unknown locations $\boldsymbol{x}$ of the simulation grid are visited according to a predefined (random or regular) path. For each $\boldsymbol{x}$ the simulated value is drawn from a cumulative distribution function F conditioned to a local data event $\boldsymbol{d}_n$:

$$F(z,\boldsymbol{x},\boldsymbol{d}_n) = \mathrm{Prob}\{Z(x) \le z \,|\, \boldsymbol{d}_n\} \quad (1)$$

This data event comprises the values of the known neighboring grid nodes $\boldsymbol{x}_i$, i.e., the conditioning data and the already simulated grid nodes, and their relative positions. F is built based on the central nodes of TI patterns equal or similar to $\boldsymbol{d}_n$.

The Direct Sampling (DS) algorithm is a recent MPS algorithm (Mariethoz et al., 2010)[1]. The particularities of DS consist in skipping the explicit modeling of F by directly sampling the TI

Abbreviations: DS, Direct Sampling code; MPS, multiple-point statistics; TI, training image

* Corresponding author. Tel.: +3292646042; fax: +3292646247.
E-mail address: Eef.Meerschman@UGent.be (E. Meerschman).

0098-3004/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cageo.2012.09.019

[1] It is the object of an international patent application (PCT/EP2008/009819). The code is available on demand for academic and research purposes. Requests should be sent to Renard, Mariethoz or Straubhaar.

during simulation, and in using dissimilarity distances between $\boldsymbol{d}_n$ and the TI patterns. As soon as a TI pattern is found that matches $\boldsymbol{d}_n$ exactly or as soon as the distance between the TI pattern and $\boldsymbol{d}_n$ is lower than a given threshold, the value at the central node of the TI pattern is directly pasted to $\boldsymbol{x}$. Since the TI is scanned randomly, this strategy is equal to drawing a random value from F, but increases simulation speed. Other MPS algorithms, like the widely used *snesim* (Strebelle, 2002) and *IMPALA* (Straubhaar et al., 2011) algorithm, scan the TI beforehand for all possible $\boldsymbol{d}_n$'s and store the TI probabilities in a catalogue. Therefore, they are restricted to the simulation of categorical variables and need to use a predefined template for $\boldsymbol{d}_n$. Due to its unique strategy, DS allows to simulate both categorical and continuous variables, and to handle multivariate cases, only by selecting the appropriate distances measures.

Since DS is a promising simulation technique for a wide range of applications, it is important to understand precisely its capacities and its sensitivity to the user-defined input parameters. DS is implemented in the ANSI C language and all input and output files are in an ASCII SGeMS compatible format (Remy et al., 2009). Detailed algorithm steps and further implementation details of DS can be found in Mariethoz et al. (2010) and the user manual of DS.

Using DS requires the user to define some parameters: among them, the acceptance threshold t, the maximum fraction of TI to scan f and the maximum number of points in the neighborhood n are the most important since they are balancing simulation quality and CPU time (Section 2.1). For these three parameters, a detailed sensitivity analysis is reported by generating non-conditional simulations for the entire 3D parameter space. Next to a visual inspection of the resulting simulations, we quantify the similarity between the simulations and the TI by means of simulation quality indicators (CASE 1). The same quality indicators are calculated for a 3D example (CASE 2). We also illustrate the potential of the post-processing option (CASE 3), the multivariate simulation option (CASE 4) and the data conditioning option (CASE 5) and discuss the corresponding user-defined input parameters. Table 1 summarizes the values of the parameter that we keep fixed and the range of values of the parameters that we vary.

Since Mariethoz (2010) already showed good performance of DS with as much as 54 processors, the parallelization option is not discussed here. For more information about the option to use transform-invariant distances we refer to Mariethoz and Kelly (2011).

Many previous studies have used only one TI with sinuous channels. In contrast, we include a greater variety of patterns by performing sensitivity analyses on seven TIs: an image of ice-wedge polygons (Plug and Werner, 2002), a microscopic view of a thin marble slice, an image of snow crystals, all three as categorical and continuous images, and a categorical 3D image of concrete (Fig. 1). The continuous 2D TIs are grayscale photographs with pixel values between 0 and 255; the categorical 2D TIs are derived from these by classifying them into three categories. The 3D TI is generated by sequentially simulating 2D slices constrained by conditioning data computed at the previous simulation steps (Comunian et al., 2012). The figures shown in this paper are the results for the categorical ice-wedge TI, the continuous marble TI and the 3D concrete TI. They are presented with the same color scale as the TIs in Fig. 1. The results for the other TIs can be found as supplementary electronic material available online.

2. CASE 1: Parameters balancing simulation quality and CPU time: t, f and n

2.1. Parameters t, f and n

An acceptance threshold t needs to be defined because a TI pattern matching $\boldsymbol{d}_n$ exactly is often not found, especially for continuous variables. When the distance between the TI pattern and $\boldsymbol{d}_n$ is smaller than t, the central node of the TI pattern is pasted at location $\boldsymbol{x}$. The default distances used in this paper are based on the fraction of non-matching nodes for categorical simulations and the mean squared errors for continuous simulations. The 'exponent to the distance function in the template' is

Table 1
Fixed parameters with their default values chosen for this study (sorted according to their appearance in the parameter file) and parameters that are varied with their default values and range over which they are varied (sorted according to the case number in which they are studied).

Fixed parameters	
Name	**Default**
Simulation method	MPS
Number of realizations	10
Max search distance	125 125 0 (½ size simulation grid)
Anisotropy ratios in the search window (x,y,z)	1 1 1
Transformations	0 (no transformations)
Path type	0 (random path)
Type of variable	0 for categorical, 1 for continuous
Exponent of the distance function in the template	0
Syn-processing parameters (4)	0 0 0 0 (no syn-processing)
Initial seed	1350
Parameters reduction	1 (no parameters reduction)
Parallelization	1 (serial code, no parallelization)

Varied parameters			
Name	**Default**	**Range**	**Case**
Threshold position t	0.05	0.01–0.02–0.04–0.06–0.08–0.1–0.12–0.14–0.16–0.18–0.2–0.25–0.5–0.75–0.99	1
Max fraction of TI to scan f	0.5	0.05–0.1–0.15–0.2–0.3–0.4–0.5–0.6–0.75–1	1
Max number of points in neighborhood n	50	1–5–10–15–20–30–50–80	1
Post-processing parameters			2
– Number of post-processing steps (p)			
– Post-processing factor (p_f)	0	0–1–2	
	0	0–1–3	
Number of variables to simulate jointly	1	1–2	3
Relative weight of each variable	1	0.1 0.9–0.25 0.75–0.5 0.5–0.75 0.25–0.9 0.1	3
Weight of conditioning data (δ)	1	0–1–5	4
Data conditioning	No	No–yes	4

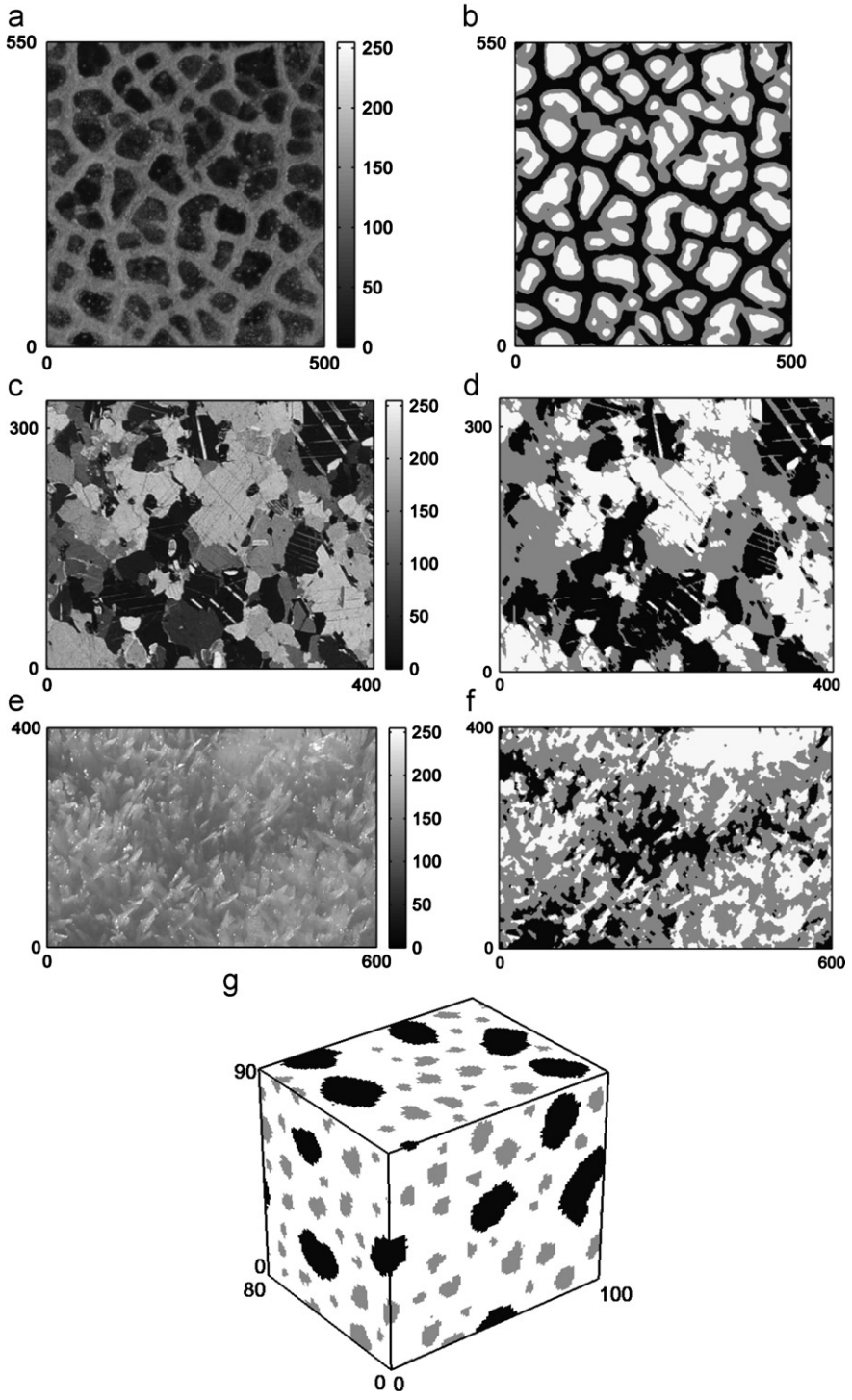

Fig. 1. The seven training images (TIs) that are used for the sensitivity analyses: (a) continuous (photograph: Plug and Werner, 2002) and (b) categorical ($k=3$) TI of ice-wedge polygons; (c) continuous and (d) categorical ($k=3$) TI of a thin marble slice; (d) continuous and (e) categorical ($k=3$) TI of snow crystals; (g) categorical ($k=3$) 3D TI of a block of concrete The x-, y- and z-axes represent the number of pixels. The results of the sensitivity analyses for TIs (b), (c) and (g) are illustrated in this paper; the results for TIs (a), (d), (e) and (f) can be found in the supplementary material available online.

set to 0, meaning that the distances are calculated without weighting the nodes in the TI pattern according to their proximity to the central node. All distances are normalized ensuring their minimum to be zero (exact match) and their maximum to be 1 (no match) (Mariethoz et al., 2010).

The maximum fraction of TI to scan f limits the number of TI patterns that are scanned for their similarity with $\boldsymbol{d}_n$: f ranges from 0 (no scan) to 1 (scan full TI if necessary). If the maximum fraction of the TI f is scanned and still no TI pattern with a distance smaller than t is found, the central node of the TI pattern with the lowest distance is pasted at location $\boldsymbol{x}$.

The neighborhood $\boldsymbol{d}_n$ is defined as the n grid nodes that are closest to $\boldsymbol{x}$ within the defined search area. This search area can be defined by setting the parameters 'maximum search distance', i.e., the radius in the x-, y- and z-direction of a rectangular search area. Generally, it is advised to use a large search area by setting the radii

to half the size of the simulation grid, corresponding to the maximum neighborhood size, except when considering non-stationary variables (Mariethoz et al., 2010), or patching occurs (discussed below). The definition of n results in $\boldsymbol{d}_n$'s covering a large part of the search area when the first unknown grid nodes are simulated, and a progressive decrease of the size of the area covered by $\boldsymbol{d}_n$ when the number of already simulated nodes increases. Consequently, DS ensures that structures of all sizes are present in the simulation, which is also the purpose of the multi-grid approach in *snesim* (Strebelle, 2002).

Fig. 2 gives an overview of the DS workflow and its three main input parameters.

It is clear that the larger n and the closer t to 0 and f to 1, the better the simulation quality will be. However, these settings will be very expensive in terms of CPU time. For all six TIs, we simulate 10 unconditional realizations for each parameter combination of 15 t values, 10f values and 8 n values (Table 1), resulting in 12,000 realizations for each TI.

2.2. CPU time

Fig. 3 shows the CPU time needed to simulate one unconditional simulation for the categorical and the continuous case. First the influence of t and n is shown for $f=0.5$ after which the influence of f is shown for different combinations of t and n.

Besides the fact that generating simulations based on the continuous TI generally takes longer, the results for the categorical and the continuous case show a similar behavior. Simulations with small t and large n require a long simulation time and decreasing f strongly reduces CPU time. Modifying one of the parameters t, f or n increases or decreases the CPU exponentially. The combined effect of relaxing all three parameters only slightly, can reduce CPU time significantly. For instance, generating one simulation for the categorical case with default parameters ($t=0.05$, $f=0.5$, $n=50$) takes 163 s. Relaxing t to 0.1 only takes 44 s, relaxing all three parameters to $t=0.1$, $f=0.3$ and $n=30$ only takes 13 s.

This behavior is related to the scanning algorithm. When t is close to 0, f close to 1 and n high, the algorithm scans the entire TI for a very good match with complex data events (large neighborhoods). This takes a lot of time. In the opposite case, the algorithm finds very quickly a TI pattern that matches the criteria and the algorithm is fast. What is striking in Fig. 3 is that the evolution between these two cases is rather abrupt for some parameter values. When the parameter values are behind such an abrupt boundary, DS is very fast whatever the parameter values, below this boundary CPU time increases.

2.3. Visual quality inspection

Fig. 4 shows the first out of 10 simulations for some combinations of t, f and n using the categorical ice-wedge TI and Fig. 5 using the continuous marble TI. The results for the other TIs can be found as supplementary material. Similar as for Fig. 3, first a sensitivity analysis on t and n is performed (for $f=0.5$), after which the effect of f is illustrated for some combinations of t and n. We select simulations with different quality levels in order to illustrate the evolution of the simulation quality. As the quality steps depend on the TI, simulations with different t and n values are illustrated for each case.

For the categorical case, running DS with $t>0.5$ or $n\leq 5$ results in noisy images. This is not surprising since then the sampling is not selective enough: any TI pattern can be accepted even if it is far away from $\boldsymbol{d}_n$. This corresponds to situations in which the algorithm is very fast. For $t\leq 0.5$ and $n>5$, the ice-wedge pattern is reasonably well reconstructed. For $t\leq 0.2$ and $n\geq 30$, the simulation quality is very good. Not only the pattern reconstruction, but also the appearance of noise and the fuzziness of the edges between different categories are influenced by t and n (CASE 3). For the categorical marble TI (Fig. 1d) similar thresholds are found (Supplementary material—Fig. b). For the categorical snow crystals TI (Fig. 1f) the simulation quality is good for $t\leq 0.1$ and $n\geq 50$ (Supplementary material—Fig. c). In contrast to the effect of t and n, f has a much smaller effect on the simulation quality. Scanning a smaller part of the TI hardly results in a quality decrease (Fig. 4b). The same conclusion can be drawn from the simulations using the other categorical TIs.

Fig. 5 shows that generating continuous simulations generally requires stricter parameters (lower t, higher n and f). Running DS with $t\geq 0.2$ results in noisy images. The simulation quality is good for $t\leq 0.1$ and $n\geq 30$. Simulations using the continuous ice-wedge TI (Fig. 1a) show important changes in visual quality for the same values of t and n (Supplementary material—Fig. a). The quality of the simulations using the continuous snow crystal TI (Fig. 1e) is generally less good: only for $t\leq 0.1$ and $n\geq 50$ the

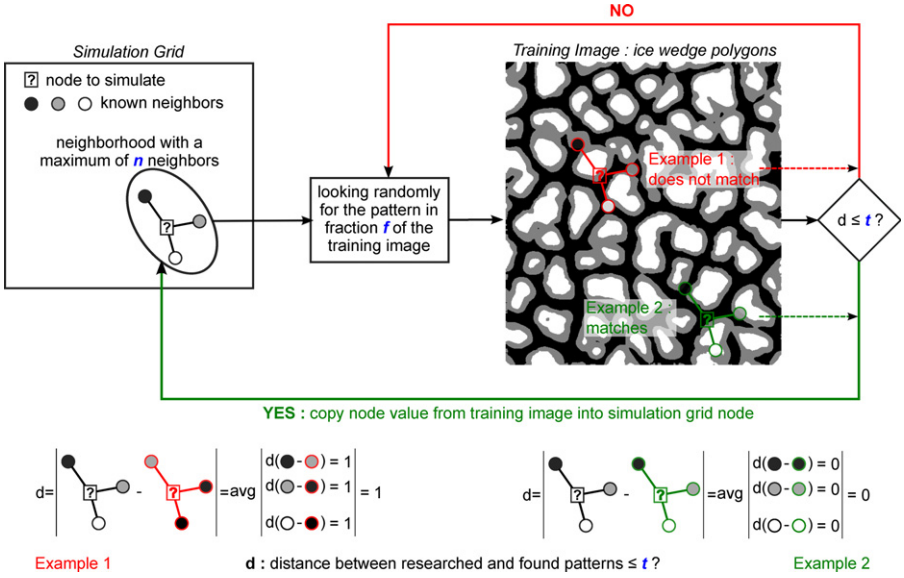

Fig. 2. Workflow of DS explaining the three main input parameters: the acceptance threshold t, the maximum fraction of TI to scan f and the maximum of neighbors n.

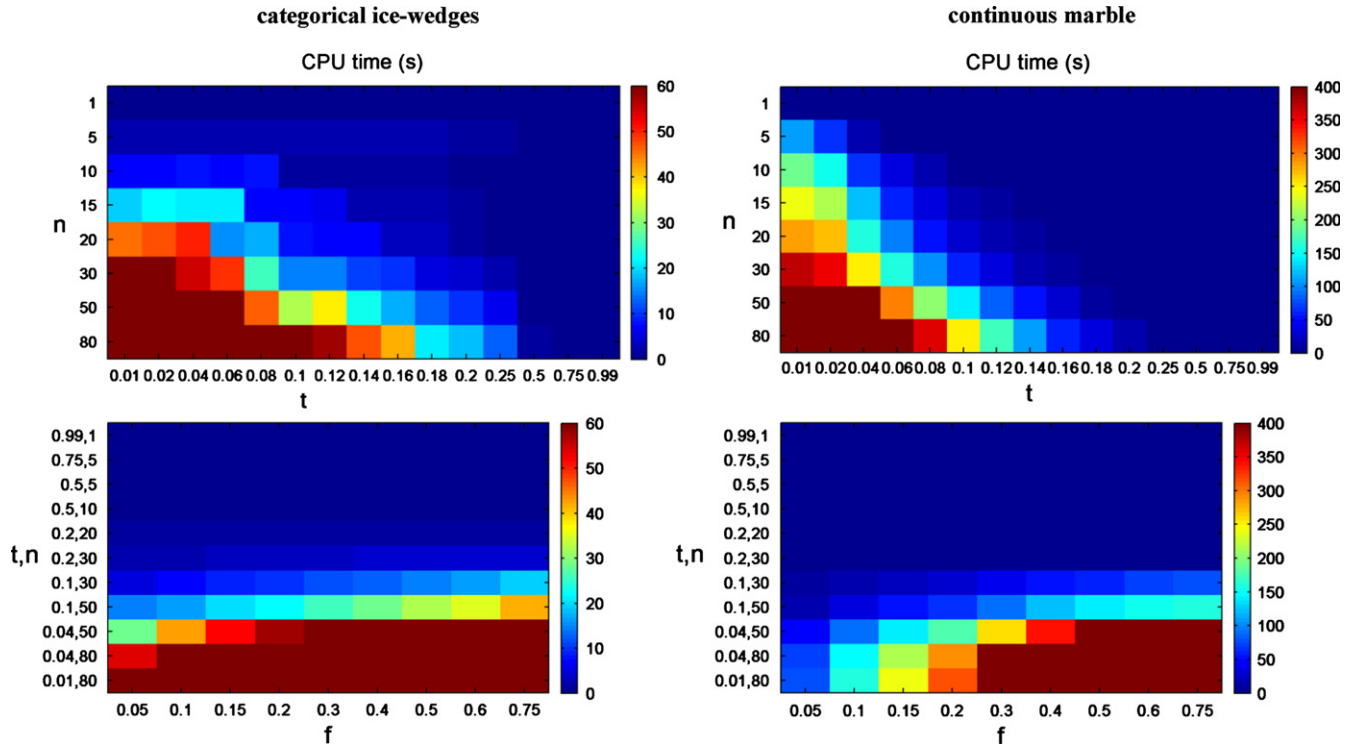

Fig. 3. (Color online) Influence of t and n (for f=0.5) (top) and f (bottom) on the CPU time required to run one unconditional simulation.

simulation quality is moderate (Supplementary material—Fig. d). For the continuous cases, it is observed that variations in f do not affect much on the simulation quality.

Especially for the continuous cases, it can be seen that some simulations are almost exact copies of parts of the TI. This phenomenon is called 'patching'. It is caused by copying each time the central node of the same best matching pattern. The issue of patching will be discussed further in the paper (Section 2.5).

2.4. Simulation quality indicators

For each unconditional simulation we calculate several quality indicators by comparing the histogram, variogram and connectivity function of the TI and the simulations. The connectivity function $\tau(\boldsymbol{h})$ for a category s is defined as the probability that two points are connected by a continuous path of adjacent cells all belonging to s, conditioned to the fact that the two points belong to s (Boisvert et al., 2007; Emery and Ortiz, 2011; Renard et al., 2011; Renard and Allard, 2012):

$$\tau(\boldsymbol{h}) = \mathrm{Prob}\{\boldsymbol{x} \leftrightarrow \boldsymbol{x}+\boldsymbol{h} \,|\, s(\boldsymbol{x})=s, s(\boldsymbol{x}+\boldsymbol{h})=s\} \tag{2}$$

Fig. 6 compares the histogram, variogram and connectivity function of the categorical ice-wedge TI (Fig. 1b) with these of a good simulation (t=0.01, f=0.5, n=80) and a bad simulation (t=0.5, f=0.5, n=15 for categorical and t=0.2, f=0.5, n=5 for continuous). Both the indicator variogram values $\gamma^k(\boldsymbol{h})$ and the connectivity function $\tau^k(\boldsymbol{h})$ for each category k are calculated for 20 lag classes $\boldsymbol{h}$ with a lag width of 5. The histograms (proportions of the three categories) are represented for both simulations. The indicator variograms and the connectivity functions are only reproduced for the good simulation, except for the intermediate material (grey), where the bad simulation partially reproduces the TI statistics.

Fig. 7 illustrates the same for the continuous case. Here the standard variogram $\gamma(\boldsymbol{h})$ is calculated instead of the indicator variogram. To calculate the connectivity functions, the TI and the simulations are first divided into three categories based on two thresholds representing connectivity jumps in the TI (Renard and Allard, 2012). Similarly to Fig. 6, the histograms (represented as the cdf) are represented for both the good and the bad simulation, whereas the variogram and the connectivity functions are only reproduced by the good simulation.

To quantify the dissimilarity between the simulations' statistics and those of the TI, we calculate three error indices for each simulation: a histogram error ε_{hist}, variogram error ε_{var} and connectivity error ε_{conn}. For the categorical simulations, ε^k_{hist} is defined as the absolute value of the difference between the proportion of k in the simulation grid f^k_{sim} and in the TI f^k_{TI}. For the continuous simulations, ε_{hist} is calculated as the Kullback–Leibler divergence D_{KL} (Kullback and Leibler, 1951):

$$\varepsilon_{hist} = D(P||Q) = \sum_i P(i)\log\frac{P(i)}{Q(i)} \tag{3}$$

with $P(i)$ the probability distribution in the simulations and $Q(i)$ the probability distribution in the TI.

The variogram error ε_{var} for the continuous simulations is based on the weighted average difference between the variogram values of the simulations $\gamma_{sim}(\boldsymbol{h})$ and the TI $\gamma_{TI}(\boldsymbol{h})$ for 20 lag classes $\boldsymbol{h}_d$, and is calculated as

$$\varepsilon_{\mathrm{var}} = \frac{\sum_{d=1}^{20} (1/\boldsymbol{h}_d)\,|\gamma_{sim}(\boldsymbol{h}_d) - \gamma_{TI}(\boldsymbol{h}_d)|/(\mathrm{var}_{sim})}{\sum_{d=1}^{20} (1/\boldsymbol{h}_d)} \tag{4}$$

with var_{sim} the simulation variance used to standardize the absolute errors, so they range between 0 and 1. The term $1/\boldsymbol{h}_d$ is included to give larger weights to errors corresponding to small variogram lags. The variogram error $\varepsilon^k_{\mathrm{var}}$ for the categorical case is calculated similarly using the indicator variogram values.

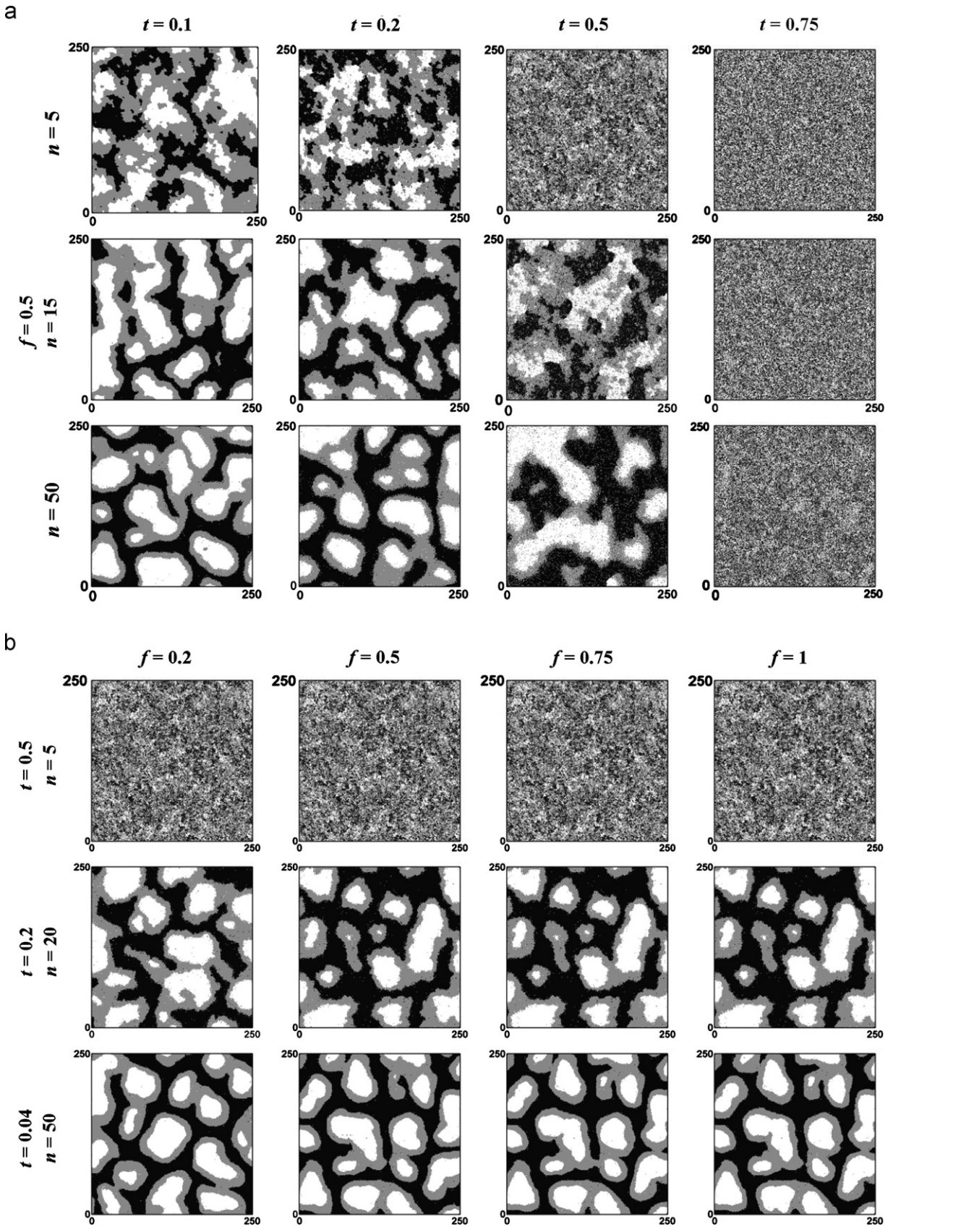

Fig. 4. (a) First out of 10 unconditional simulations illustrating the effect of parameters t and n with f=0.5 and (b) first out of 10 unconditional simulations illustrating the effect of f for constant t and n based on the categorical ice-wedge TI (Fig. 1b).

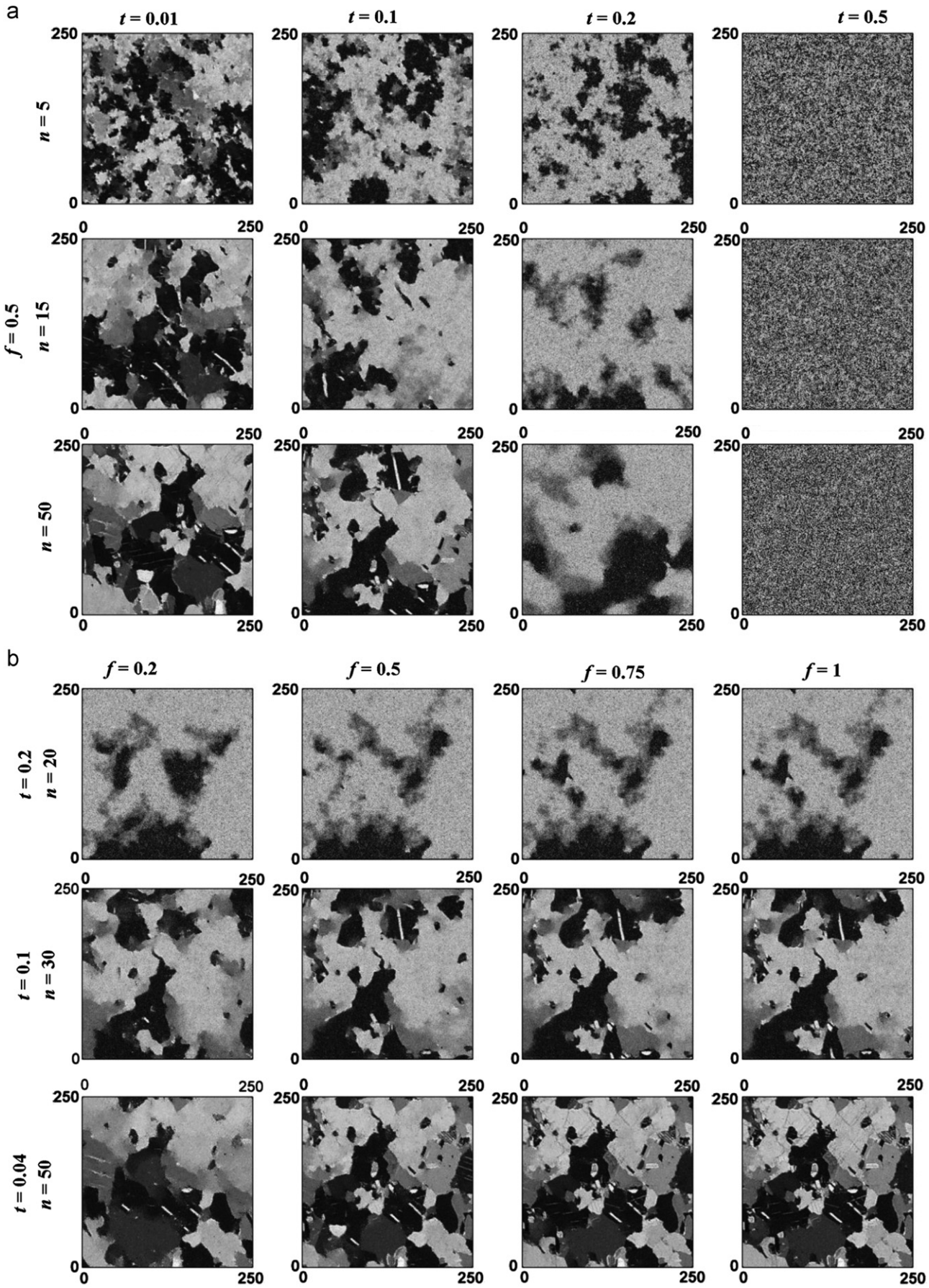

Fig. 5. (a) First out of 10 unconditional simulations illustrating the effect of parameters t and n with $f=0.5$ and (b) first out of 10 unconditional simulations illustrating the effect of f for constant t and n based on the continuous marble TI (Fig. 1c).

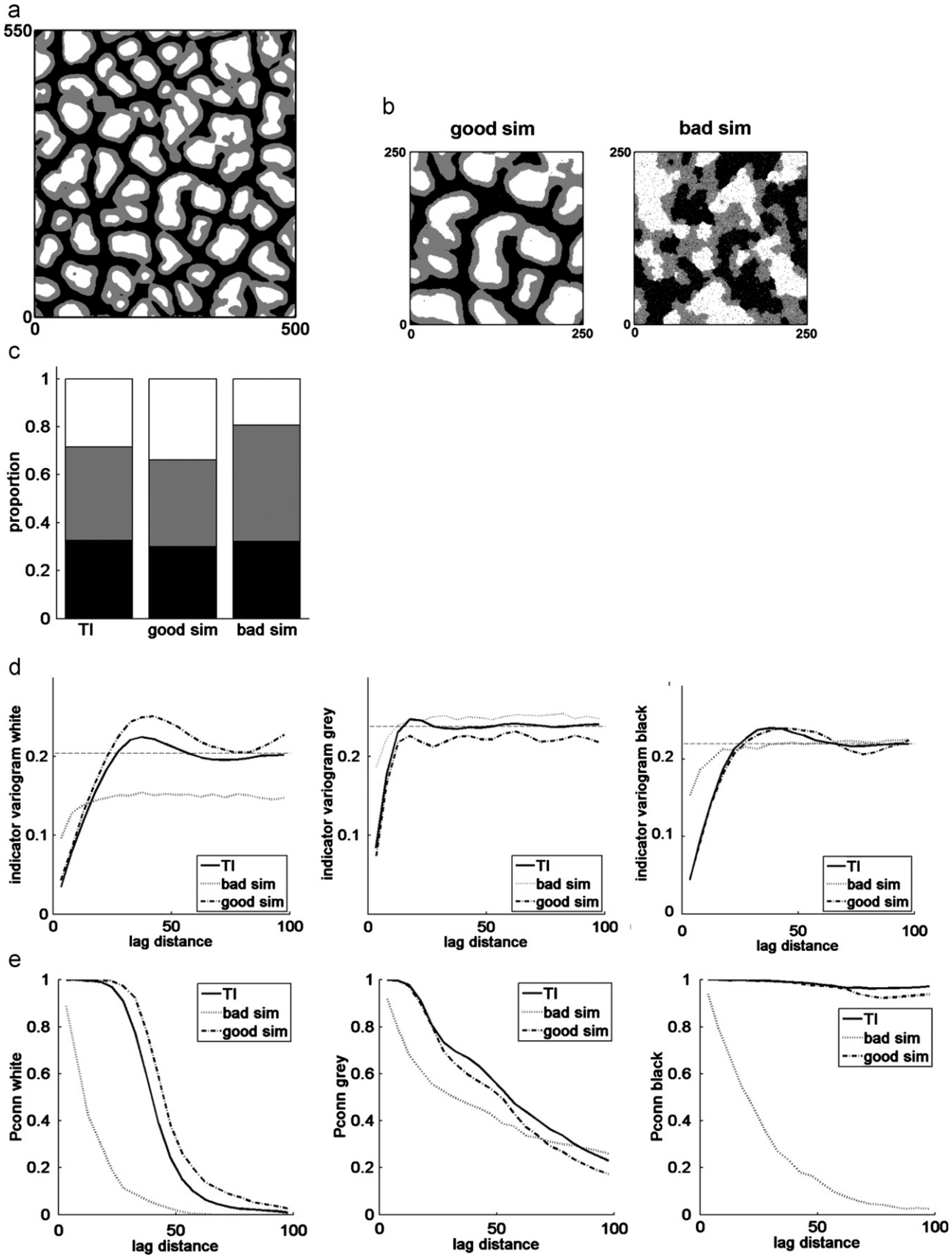

Fig. 6. Reproduction of (a) the categorical ice-wedge TI statistics of (b) a good ($t=0.01$, $f=0.5$, $n=80$) and a bad simulation ($t=0.5$, $f=0.5$, $n=15$) with the reproduction of (c) the histogram, (d) the indicator variograms (the dotted lines represent the TI indicator variance) and (e) the connectivity functions.

The connectivity error ε_{conn} is calculated as

$$\varepsilon_{conn} = \frac{\sum_{d=1}^{20} |\tau_{sim}^{k}(\boldsymbol{h}_d) - \tau_{TI}^{k}(\boldsymbol{h}_d)|}{20} \quad (5)$$

and also ranges between 0 and 1.

2.5. Results and discussion

Figs. 8 and 9 show the results of the simulation quality indicators for the categorical and the continuous case. The first part of the figures illustrates the effect of t and n, the second part the effect of f.

As was already concluded from Figs. 6 and 7 ε_{hist} behaves differently than ε_{var} and ε_{conn}. The histogram is generally well reproduced for all simulations. Noisy images reproduce the histogram the best, which is especially clear for the continuous case. This can be explained by considering the extreme case of $t=1$. With such a setting, DS randomly samples values from the TI, resulting in a perfect reproduction of the marginal distribution ($\varepsilon_{hist}=0$), but no reproduction of spatial continuity (very large ε_{var} and ε_{conn}). For intermediate combinations of t and n, ε_{hist} is generally larger. In the areas with good simulation quality ($t \leq 0.2$ and $n \geq 30$) ε_{hist} behaves differently for the categorical and the continuous case. For the categorical case low t and high n guarantee both low ε_{hist} and good simulation quality (Fig. 4, Supplementary material—Figs f and g). For the continuous case, the high quality simulations ($t \leq 0.2$ and $n \geq 30$) have higher ε_{hist}. This counterintuitive result can be explained as follows: with low t and high n, the simulation has to honor very strong spatial constraints. When the structures are made of objects that are large with respect to the domain size, respecting such spatial constraints means to respect the connectivity of facies and the objects size, even if it contradicts the target pdf. Because of a slight non-stationarity in the TI, the simulation can then follow the pdf of one specific part of the TI that is different than the rest. This can result

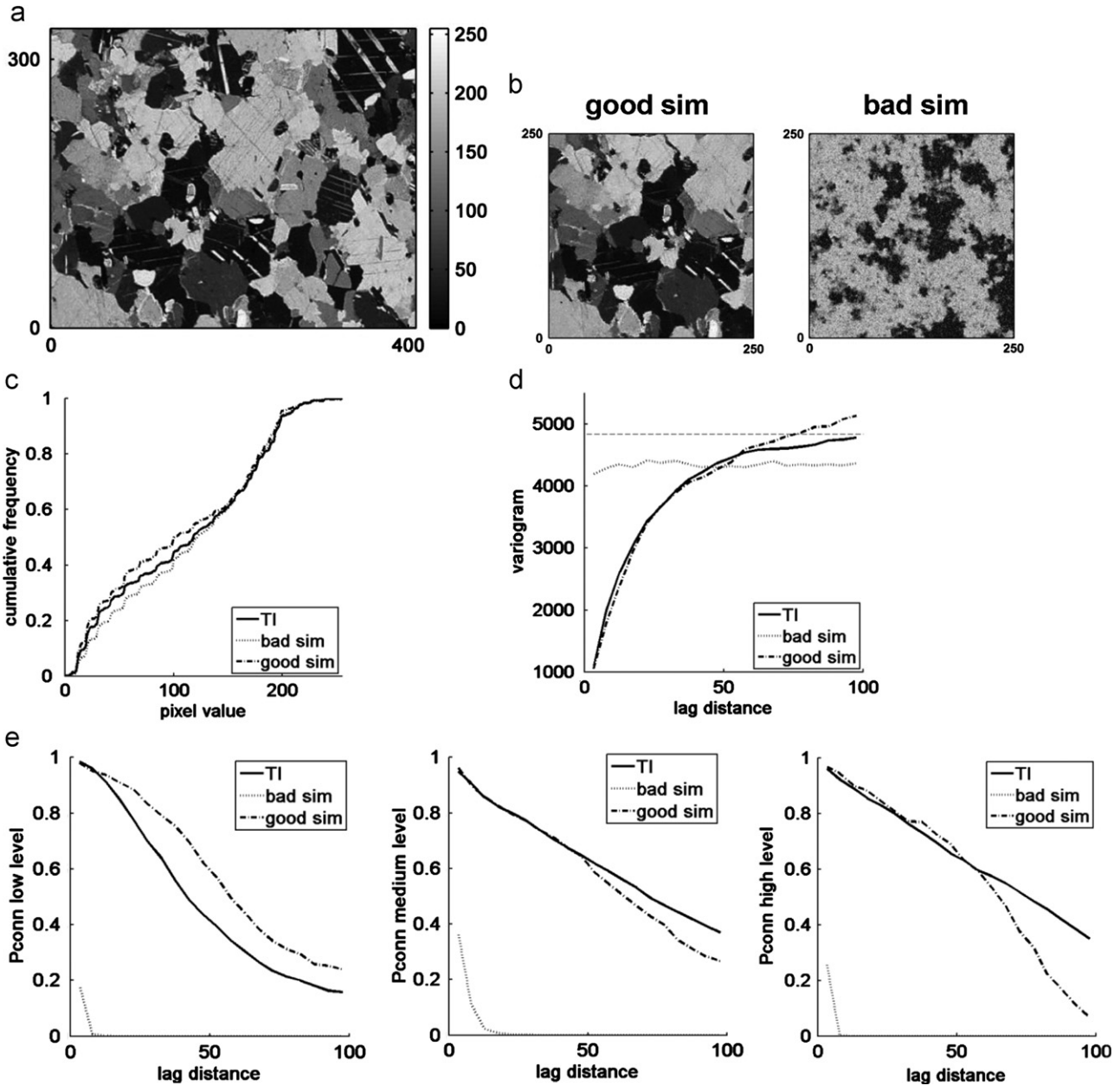

Fig. 7. Reproduction of (a) the continuous marble TI statistics of (b) a good ($t=0.01$, $f=0.5$, $n=80$) and a bad simulation ($t=0.2$, $f=0.5$, $n=5$) with the reproduction of (c) the histogram, (d) the variogram (the dotted line represents the TI variance) and (e) the connectivity functions.

in significant variability in the pdfs of the simulations. This is the opposite as the case of $t=1$, where the TI distribution is honored because there is no constraint on the spatial continuity. For the continuous ice-wedge TI (Fig. 1a) and the snow crystal TI (Fig. 1e), the histogram is well reproduced in the high quality simulations (Supplementary material—Figs e and h). Since certain applications require the histogram to be reproduced, this issue could be further addressed by the DS developers.

In contrast, ε_{var} and ε_{conn} increase for larger t and smaller n, which is a more intuitive behavior. Both errors show similar quality jumps as were derived from the visual inspection and therefore behave as stable simulation quality indicators.

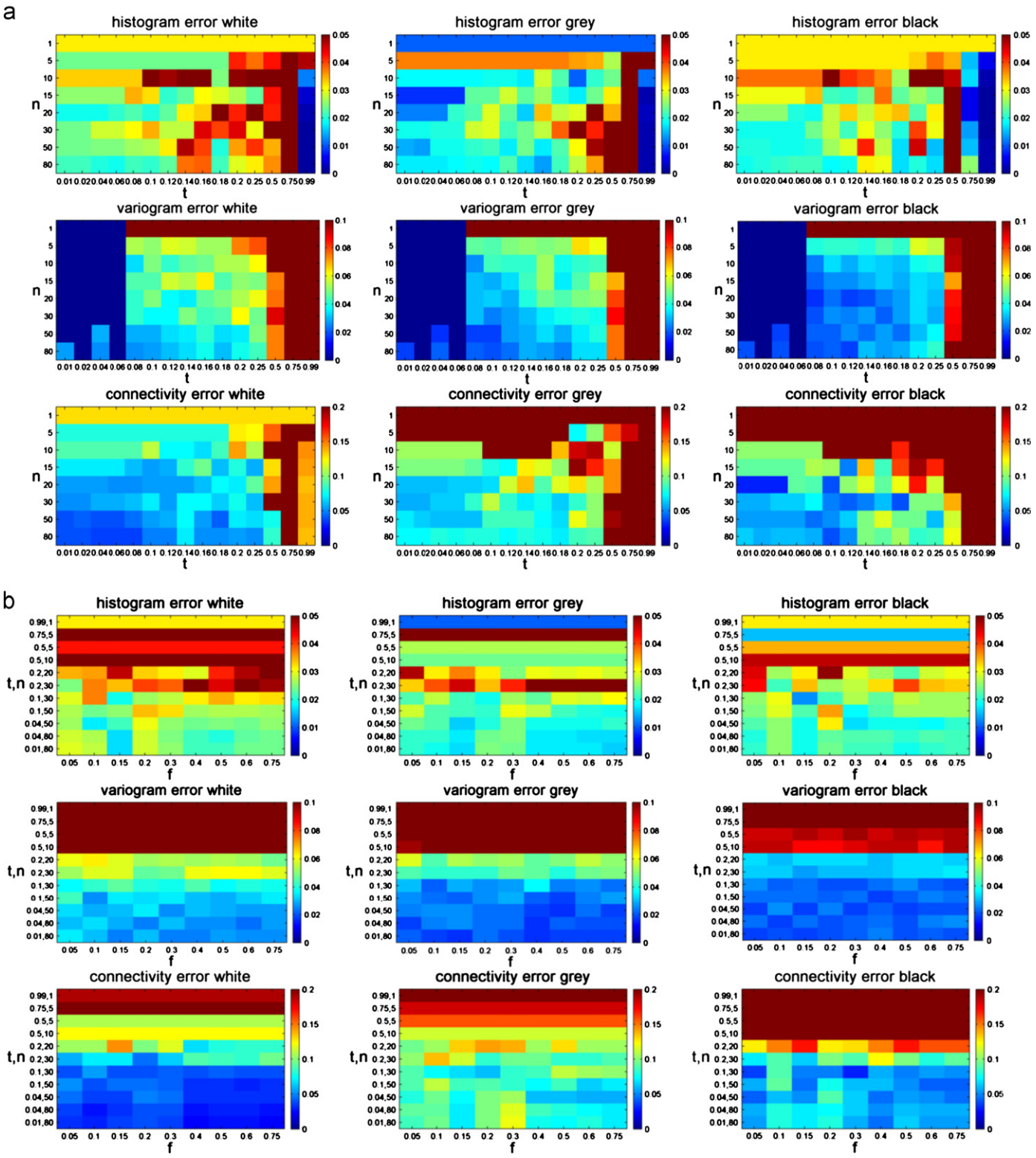

Fig. 8. (Color online) Influence of (a) t and n (for $f=0.5$) and (b) f on the quality indicators based on the categorical ice-wedge TI (Fig. 1b).

These results allow us to derive some rules of thumb. Running DS using a categorical TI should result in good simulations for $t \leq 0.2$ and $n \geq 30$. Selecting $t \geq 0.5$ and $n < 15$ should be avoided. For continuous TI, it is advised to use $t \leq 0.1$, $n \geq 30$ and to avoid $t \geq 0.2$ and $n \leq 15$. The quality of intermediate combinations is hard to predict. It is obvious that the simulation quality steps strongly depend on the TI. The greater the pattern repeatability in the TI, the better the simulation quality will be.

Fig. 8b and Fig. 9b lead again to the conclusion that f does not have a strong influence on the simulation quality. This is confirmed by the other TIs (Supplementary material—Figs e–h). Only for certain situations, like for $f < 0.2$ in the continuous case (see results for ε_{var}), the pattern reproduction degrades with lower f since the probability of finding a matching TI pattern is lower. Note also that using a small f value for TIs that contain insufficient diversity (Mirowski et al., 2009), might aggravate the statistical scarcity and lead to poor results. But generally decreasing f results in large computation gains without a substantial decrease in simulation quality, which is an important conclusion for an efficient use of DS.

It is interesting to juxtapose the CPU time (Fig. 3) with the corresponding quality indicators (Figs. 8 and 9). This reveals where the interesting boundaries are located in terms of quality over CPU time ratio. For instance, for the categorical case the quality is moderate from $n \geq 15$ and $t \leq 0.18$ ($f=0.5$) (Fig. 8a), whereas the CPU time really increases from $n \geq 30$ and $t \leq 0.1$ (Fig. 3a). In between these boundaries, the simulation quality is good, as is confirmed by the visual inspection. In case CPU time is a limiting factor, users are recommended to investigate the quality over CPU time ratio for different parameter combinations running trial simulations on a small grid.

It is good practice to run DS initially with $f=0.5$, t between 0.05 and 0.2 and n between 20 and 50. From this, the parameters need to be fine-tuned for every particular situation, knowing that decreasing t and increasing n and f should result in better simulation quality. However, one should keep in mind that using parameters which guarantee very good simulations has two drawbacks. First, these configurations will require large CPU times. Second, there is a risk of generating simulations which are all exact copies of (part of) the TI (patching effect or verbatim copy). This risk is higher when the TI does not show enough pattern repeatability (which is more often the case for continuous TIs) and when there are no conditioning data (CASE 5). Strategies to avoid patching are choosing $f < 1$, thus avoiding to pick each time the same best matching mode, slightly relaxing t and n, or using a smaller 'maximum search distance'.

3. CASE 2: 3D simulation

Similarly to two dimensions (CASE 1), DS can generate 3D simulations. To demonstrate this, we perform a limited sensitivity

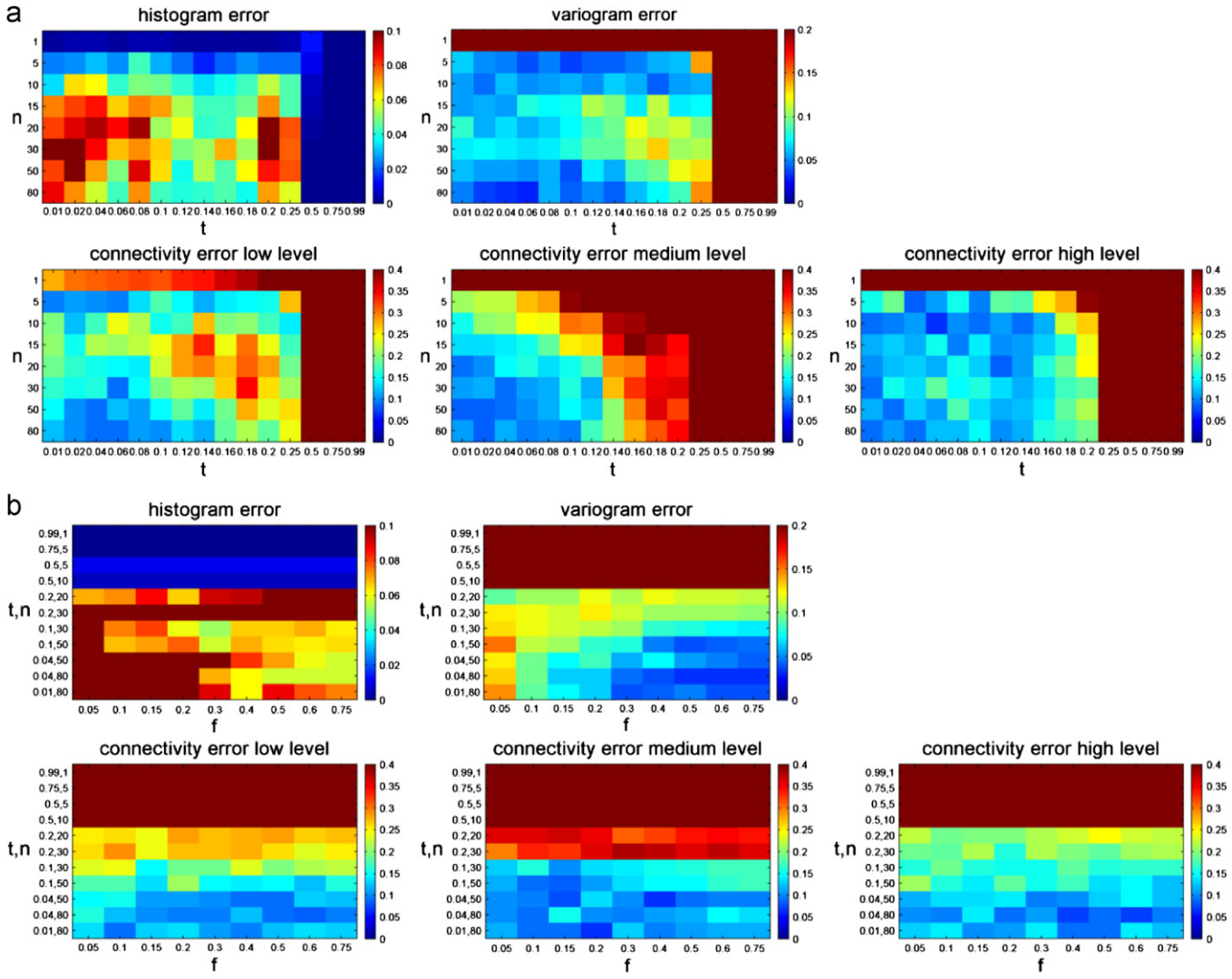

Fig. 9. (Color online) Influence of (a) t and n (for $f=0.5$) and (b) f on the quality indicators based on the continuous marble TI (Fig. 1c).

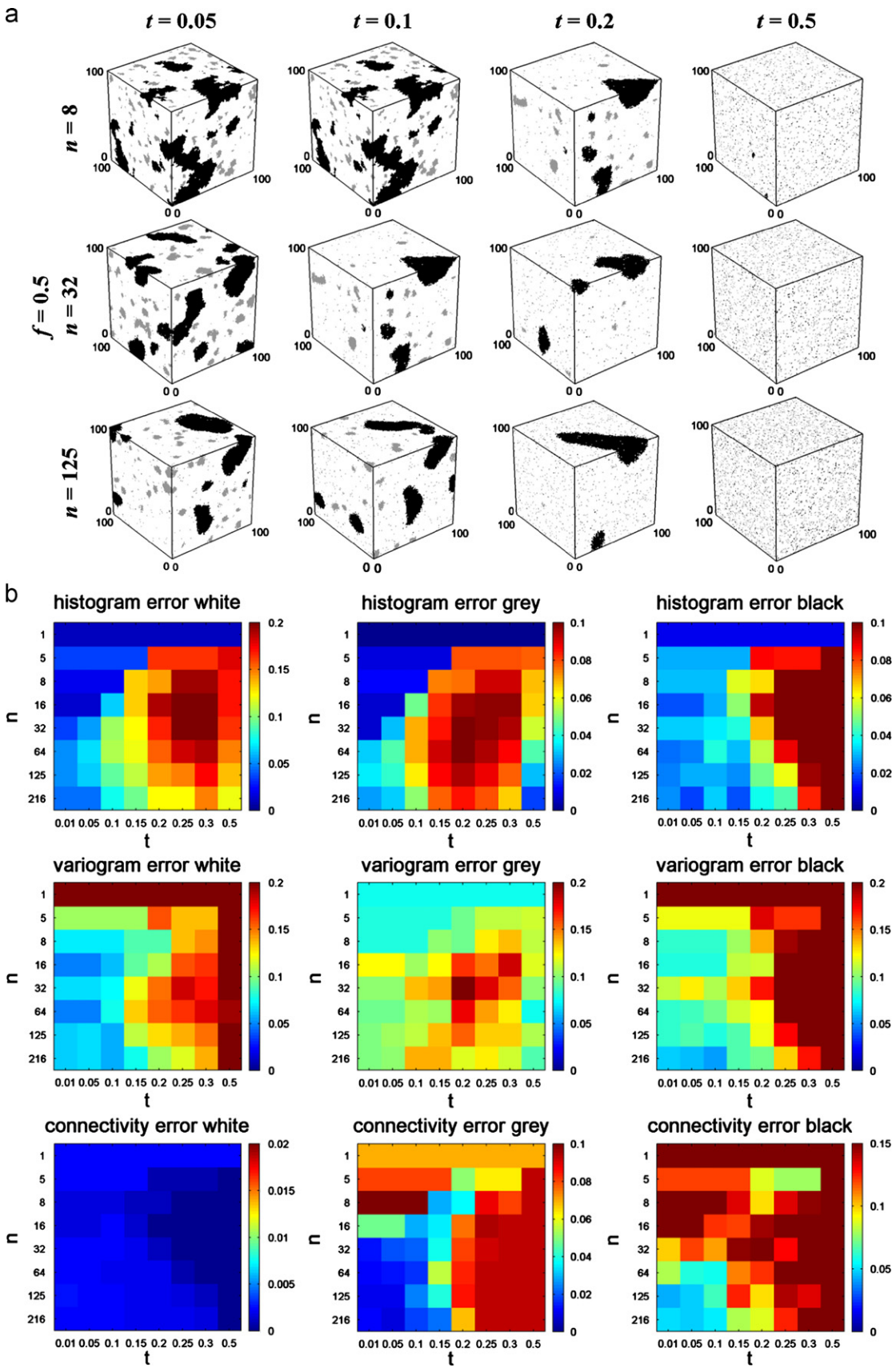

Fig. 10. (Color online) 3D example with (a) first out of 10 unconditional simulations illustrating the effect of parameters t and n with $f=0.5$ and (b) influence of t and n (for $f=0.5$) on the quality indicators based on the 3D concrete TI (Fig. 1g).

analysis using the 3D concrete TI (Fig. 1g). We generate 10 unconditional simulations for each combination of eight t [0.01–0.05–0.1–0.15–0.2–0.25–0.3–0.5] and eight n [1–5–8–16–32–64–125–216] values, using a fixed value of 0.5 for f. The other parameters are set as indicated in Table 1, with exception of the maximum search distance that is defined as half of the search grid in three directions (x, y and z).

The CPU time as a function of t and n (not shown here) behaves very similar as for 2D (Fig. 3). For instance, generating one simulation with $t=0.1$ and $n=32$ takes 194 s, with $t=0.05$ and $n=32$ 1998 s and with $t=0.05$ and $n=64$ 6804 s.

Fig. 10b shows the simulation quality indicators (see Section 2.2.4) as a function of t and n. Overall, the results are analogous to those of CASE 1. The simulation quality generally improves with increasing n and decreasing t with a quality jump for $t=0.2$ and $n=32$, as can be seen from $\varepsilon_{conn}^{grey}$ and $\varepsilon_{conn}^{black}$, and the unconditional simulations shown in Fig. 10a. Again, ε_{hist} is the smallest for noisy simulations with very small n.

Since the white category represents the background volume, $\varepsilon_{conn}^{white}$ is very small for all parameter combinations and hence not informative. With parameters producing noisy simulations ($t \geq 0.3$ and $n \leq 8$), ε_{var}^{grey} is again lower. This can be explained by the small range of the grey indicator variogram, causing ε_{var}^{grey} to be small for pure nugget variograms reproducing the sill correctly.

4. CASE 3: Post-processing for noise removal

To further improve the simulation quality and more specifically to remove noise, DS includes a post-processing option. After having simulated all the unknown grid nodes, it is possible to resimulate each node with an entirely informed neighborhood. Two post-processing parameters need to be defined: the number of post-processing steps p and the post-processing factor p_f. The latter is the factor by which f and n are divided aiming to save CPU time in the additional post-processing steps (Mariethoz, 2009). For example, if $p=2$ and $p_f=3$, all nodes are resimulated two times with parameters f and n 3 times lower than their original values. Fig. 11 illustrates the noise removal effect of post-processing for increasing t and varying p and p_f for the categorical ice-wedge TI (Fig. 1b).

The post-processing step proves to be valuable especially for intermediate t values (0.1 and 0.2), since the noise can be considered

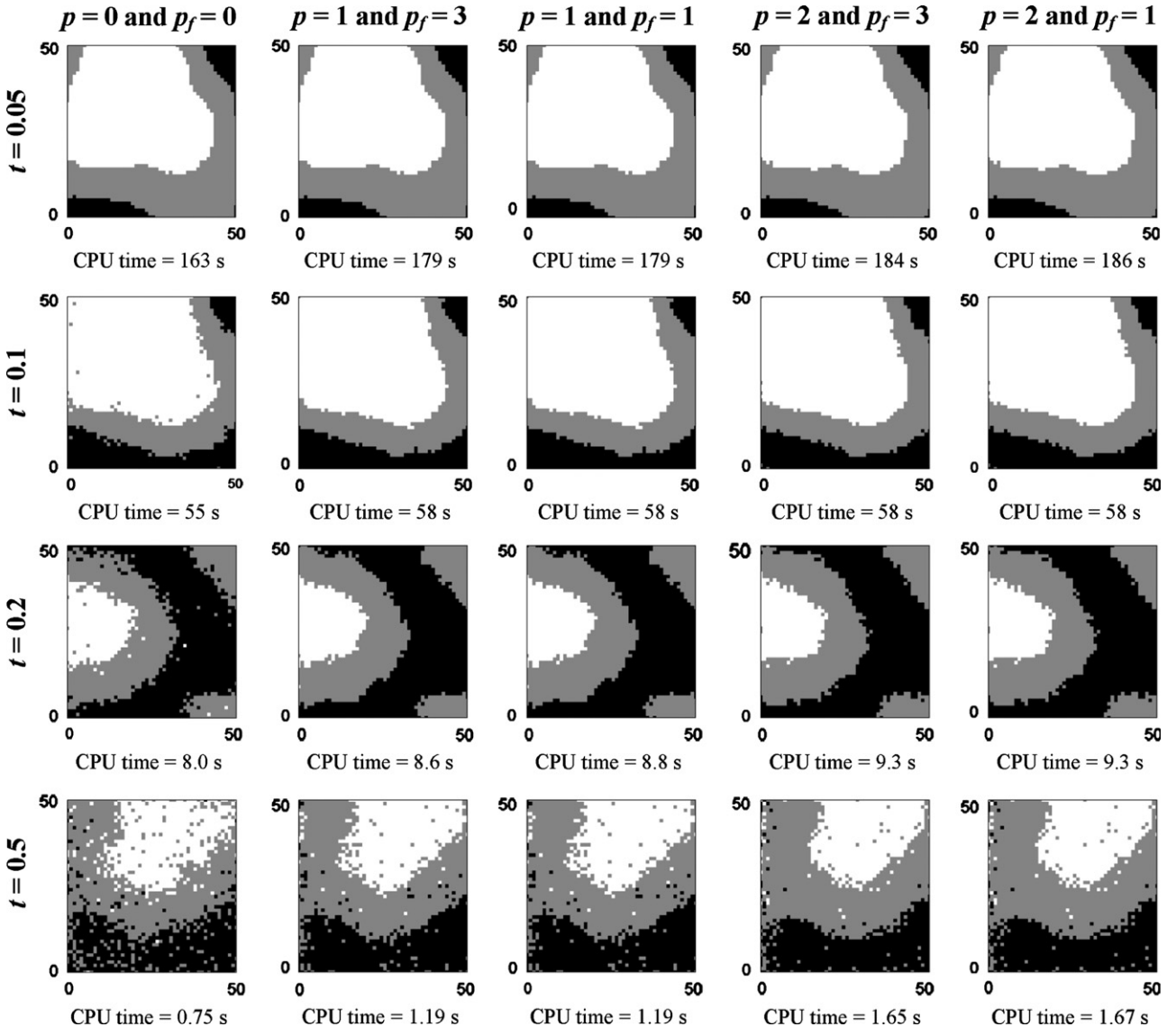

Fig. 11. Illustration of the noise removal effect of post-processing using the categorical ice-wedge TI (Fig. 1b) for increasing t, and sensitivity analysis for the number of post-processing steps (p) and the post-processing factor (p_f) showing the lower left corner of the simulation grid.

as entirely removed without substantially increasing CPU time. The simulations obtained with intermediate t after post-processing are similar to these obtained with small t, except for the boundaries which are less sharp. Furthermore, post-processing allows for a significant reduction in CPU time. With $t=0.1$ and one post-processing step, one obtains in 58 s realizations similar to when using $t=0.05$ without post-processing, which takes 163 s. For small t (0.05) the post-processing step is not necessary since the simulation quality is already good without it. For high t (0.5) it is insufficient since post-processing only removes noise and does not improve structures at larger scale. Repeating the post-processing step does not result in significant quality improvements, and whether or not f and n are decreased in the post-processing step neither decreases the CPU time nor improves the simulation quality.

The effect of a post-processing step is less substantial for the continuous case than for the categorical case and the CPU cost is much higher (not shown here). Hence, for continuous cases, the quality loss of selecting a high t, cannot be corrected with one or more additional post-processing steps.

Since p and p_f have to be chosen in advance, it can be considered as good practice to add one additional post-processing step when simulating categorical variables. When noise appears, it will be reduced and the extra CPU time needed is relatively low. For p_f a value of 1 can be selected, since adjusting p_f does not seem to have an effect. If the simulations still contain noise after post-processing, it is however advised to decrease t instead of adapting p and p_f.

5. CASE 4: Multivariate simulation

Among the MPS methods, only DS has demonstrated its potential to simulate m variables simultaneously based on m TIs. These variables can be continuous, categorical or a mixture of both since for each a different distance measure can be chosen (distance type parameter set to 0 for categorical and 1 for continuous). Several implementations have been tested. The one of Mariethoz et al. (2010) is used in this paper. First, a path is defined that goes randomly through all the unknown grid nodes for each variable $\boldsymbol{x}_m$. When one variable is simulated at one location, the other variable at the same location can be simulated later in the path. For each $\boldsymbol{x}_m$ a multivariate $\boldsymbol{d}_n$ is built containing the neighboring data for the m variables, which do not have to be collocated. For each variable the maximum number of neighbors n_m can be defined separately. Based on a weighted average of the m selected distance measures, the multivariate TI pattern, centered at the same node for each TI, is chosen that is similar to the multivariate $\boldsymbol{d}_n$. The weights used to define the multivariate distance measure w_m are user-defined. DS automatically normalizes their sum to one. The value at the central node of this multivariate TI pattern is then pasted at $\boldsymbol{x}_m$. If conditioning data are given for all or some of the m variables, they will be honored (Mariethoz et al., 2010) as shown in CASE 5.

A potential application is a situation where one variable is (partially) known and the other(s) are to be simulated (the collocated simulation paradigm). DS becomes especially interesting when the relationship between the variables is known via the training data set but not expressed as a simple mathematical relation. Applications can be found in Mariethoz et al. (2010), (in press) and Meerschman and Van Meirvenne (2011). As an illustration we show five unconditional bivariate simulations using the categorical and continuous ice-wedge TI (Fig. 1a and b) as bivariate TI and perform a sensitivity analysis on the weights given to both variables (Fig. 12). For the other parameters we use the default values as given in Table 1: both n_{cat} and n_{cont} are 50.

Fig. 12 shows that not only the spatial texture of each TI is reproduced, but also the multiple-point dependence between the TIs. The weights given to each variable strongly influence the continuous variable. The larger w_{cont}, the better the quality of the continuous variable. The quality of the continuous variable

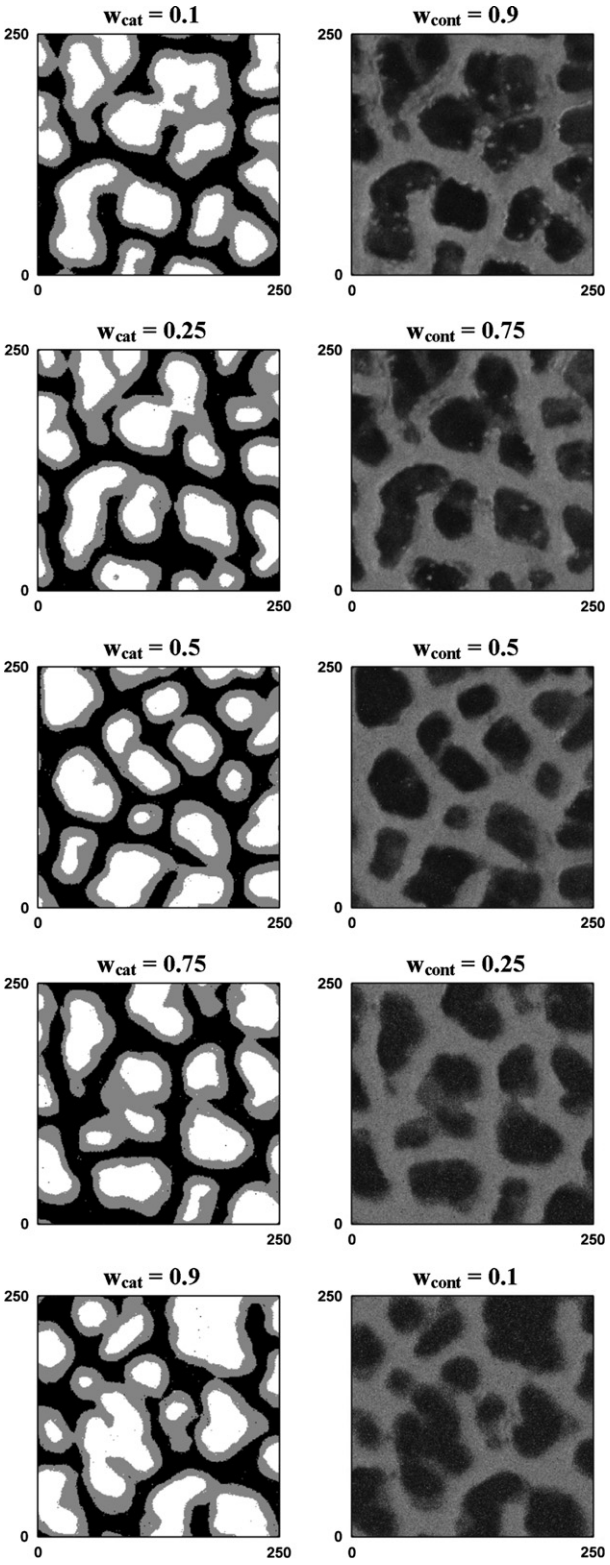

Fig. 12. Illustration of the multivariate simulation option: five unconditional bivariate simulations using Fig. 1a and b as bivariate TI, and sensitivity analysis for the weights given to the two variables (w_{cat} and w_{cont}). The left column represents the categorical variable for each simulation, and the right column represents the corresponding continuous variable.

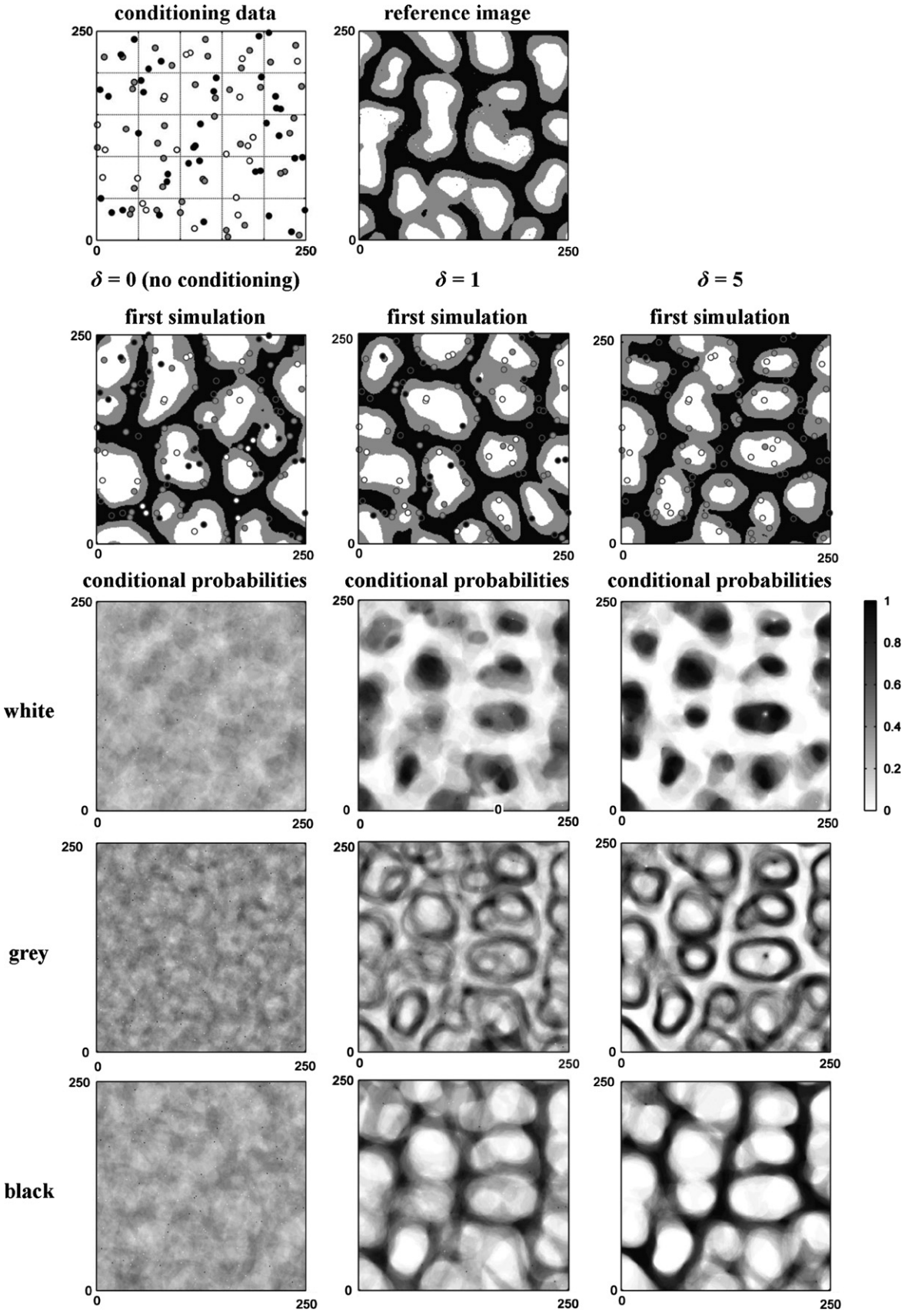

Fig. 13. Illustration of data conditioning for the categorical ice-wedge TI (Fig. 1b) based on 100 conditioning data. For $\delta=0$, $\delta=1$ and $\delta=5$ the first simulation is shown together with the conditional probabilities for each category summarizing 50 simulations.

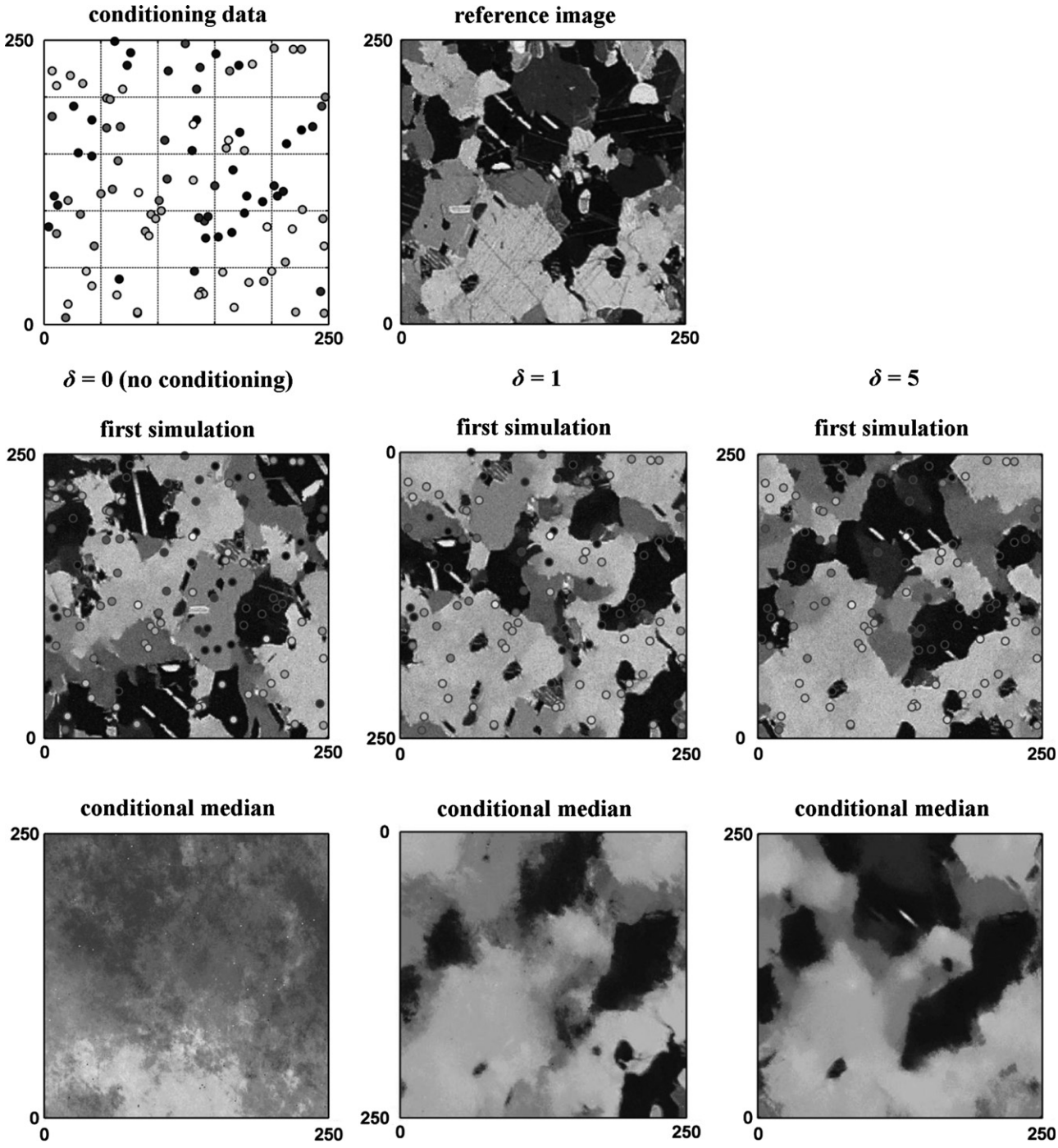

Fig. 14. Illustration of data conditioning for the continuous marble TI (Fig. 1c) based on 100 conditioning data. For $\delta=0$, $\delta=1$ and $\delta=5$ the first simulation is shown together with the conditional median for each category summarizing 50 simulations.

decreases for smaller w_{cont}. In such cases, the bivariate relationship between both TIs is well respected, but the spatial continuity of the continuous variables is not strongly imposed. The quality of the categorical simulations is less affected by the choice of the weights. Note that for large w_{cont} the continuous simulation is almost an exact copy of the continuous TI (Fig. 1a), which is again an example of the patching effect described in CASE 1.

These results and other numerical experiments (not shown here) suggest that it is often beneficial for the quality of the simulation of continuous variables to co-simulate a categorical variable that helps reproducing the continuity of the structures. This is a generally accepted technique in image processing in which the categorical variable is called 'feature map' (Lefebvre and Hoppe, 2006; Zhang et al., 2003).

6. CASE 5: Data conditioning

DS always honors conditioning data by assigning them to the closest grid node prior to simulation. Hence, local accuracy is guaranteed (the pixels at the data locations will have the correct values) but the simulated structures need to be coherent with the conditioning data. Therefore, one needs to check whether the fixed grid nodes are included in the spatial pattern or whether

they appear as noise. The parameter that can be used to enforce pattern consistency in the neighborhood of the conditioning data is the data conditioning weight δ. This parameter is used in the distance computation to weight differently data event nodes that correspond to conditioning data. If $\delta=1$, all the nodes in $\boldsymbol{d}_n$ have the same importance. For $\delta>1$ higher weights are given to the data event nodes that are conditioning data, while for $\delta<1$ they are given lower weights (Mariethoz et al., 2010; Zhang et al., 2006).

For both the categorical and continuous cases, one of the best unconditional simulations ($t=0.01$, $f=1$, $n=80$) is used as reference image. To avoid using reference images that are copies of part of the TI due to patching, we first mirrored both simulations horizontally and vertically, before sampling 100 conditioning data from each according to a stratified random sampling scheme (Figs. 13 and 14). Using the default parameters for t, f and n (Table 1), we run 50 simulations using these conditioning data and the corresponding TI. To remove noise, one post-processing step is performed with $p_f=1$ (CASE 3). Conditioning data nodes are not resimulated during post-processing. For $\delta=0$, $\delta=1$ and $\delta=5$, the first conditional simulation is shown together with the conditional probabilities for category k in the categorical case (Fig. 13), and the median over the 50 simulations for the continuous case (Fig. 14).

It can be concluded that δ is an important parameter when conditioning data are available. For $\delta=0$ the 50 simulations can be considered as unconditional simulations, since the conditioning data grid nodes are ignored in $\boldsymbol{d}_n$. The simulation patterns are not at all consistent with the conditioning data and the large variation between the simulations results in non-informative summarizing images. For $\delta=1$ the simulations show patterns that are more or less consistent with the conditioning data. The remaining inconsistencies disappear with $\delta=5$, resulting in summarizing images that closely resemble the reference images. The better results for $\delta=5$ are due to the high quality of the conditioning data, which perfectly represent the reference image without measurement errors. Generally, we advise to set δ to a value higher than or equal to 1. The lower the expected uncertainty of the conditioning data, the higher δ can be chosen.

Note that for this example the conditioning data are sampled from a field with a spatial pattern that is very similar to the TI. When one expects that the underlying spatial pattern of the conditioning data deviates more from the TI, the use of transform-invariant distances can be beneficial. This option of DS increases the number of TI patterns by randomly scaling or rotating the patterns found in the TI (Mariethoz and Kelly, 2011).

7. Conclusions

This paper has reported the first comprehensive sensitivity analysis for the DS algorithm, aiming to encourage users to benefit more efficiently from the potential of DS and its wide spectrum of applications. Given these results we provide the following general guidelines.

For categorical TIs, choosing $t\leq 0.2$ and $n\geq 30$ will generally result in high quality simulations. Smaller t and larger n result in better simulation quality and a lower level of noise. However, this choice will also depend on the available CPU time. Furthermore, for small t and large n the user should check if there is still sufficient variability between the simulations. For continuous TIs, we advise to select $t\leq 0.1$ and $n\geq 30$. For continuous cases, the selection of t and n is a delicate balance between ensuring good simulation quality and still guaranteeing sufficient variability between the simulations (avoiding patching). A good strategy to reduce both CPU time and the risk of patching is setting $f<1$, and reducing the maximum search distance to a third the domain size or less, thus scanning a different fraction of the TI for each unknown grid node.

For categorical simulations in particular, it is advised to always add one post-processing step for noise removal. If the final simulations still contain (too much) noise, improvements should be sought by adapting t and n.

Simulating bivariate images is a very new and promising technique first offered by the DS algorithm. With the illustrative example in this paper we have shown that the weights given to each variable clearly affect simulation quality. In case of continuous variable simulation, it is beneficial to add an auxiliary categorical variable that is co-simulated with a relative small weight. This generally improves the simulation of the continuous variable.

When conditioning data are available, it is interesting to put the weights given to the conditioning data (parameter δ) higher than the weights given to the already simulated nodes. This results in simulated patterns more consistent with the conditioning data.

Acknowledgments

The work presented in this paper was financially supported by the Flemish Fund for Scientific Research (FWO-Vlaanderen), by the Swiss National Science Foundation under the contract CRSI22_122249/1, and by the National Centre for Groundwater Research and Training.

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.cageo.2012.09.019.

References

Bianchi, M., Zheng, C., Wilson, C., Tick, G.R., Liu, G., Gorelick, S.M., 2011. Spatial connectivity in a highly heterogeneous aquifer: from cores to preferential flow paths. Water Resources Research 47, W05524.

Boisvert, J.B., Pyrcz, M.J., Deutsch, C.V., 2007. Multiple-point statistics for training image selection. Natural Resources Research 16, 313–321.

Comunian, A., Renard, P., Straubhaar, J., 2012. 3D multiple-point statistics simulation using 2D training images. Computers & Geosciences 40, 49–65.

Emery, X., Ortiz, J.M., 2011. A comparison of random field models beyond bivariate distributions. Mathematical Geosciences 43, 183–202.

Gómez-Hernández, J.J., Wen, X.H., 1998. To be or not to be multi-Gaussian? A reflection on stochastic hydrogeology. Advances in Water Resources 21, 47–61.

Guardiano, F.B., Srivastava, R.M., 1993. Multivariate geostatistics: beyond bivariate moments. In: Soares, A. (Ed.), Geostatistics-Troia, vol. 1. Kluwer Academic Publishers, Dordrecht, pp. 133–144.

Kullback, S., Leibler, R.A., 1951. On information and sufficiency. Annals of Mathematical Statistics 22, 79–86.

Lefebvre, S., Hoppe, H., 2006. Appearance-space texture synthesis. ACM Transactions on Graphics 25, 541–548.

Mariethoz, G., 2009. Geological Stochastic Imaging for Aquifer Characterization. Ph.D. Dissertation, University of Neuchâtel, Neuchâtel, Switzerland, 225 pp.

Mariethoz, G., 2010. A general parallelization strategy for random path based geostatistical simulation methods. Computers & Geosciences 36, 953–958.

Mariethoz, G., Kelly, B.F.J., 2011. Modeling complex geological structures with elementary training images and transform-invariant distances. Water Resources Research 47, W07527.

Mariethoz, G., McCabe, M.F., Renard, P. Spatio-temporal reconstruction of gaps in multivariate fields using the Direct Sampling approach. Water Resources Research, http://dx.doi.org/10.1029/2012WR012115, in press.

Mariethoz, G., Renard, P., Straubhaar, J., 2010. The Direct Sampling method to perform multiple-point geostatistical simulations. Water Resources Research 46, W11536.

Meerschman, E., Van Meirvenne, M., 2011. Using bivariate multiple-point geostatistics and proximal soil sensor data to map fossil ice-wedge polygons In: Jakšík, O., Klement, A., Borůvka, L. (Eds.), Pedometrics 2011. Innovations in Pedometrics: Book of Abstracts. Czech University of Life Sciences, Prague, pp. 51.

Mirowski, P.W., Tetzlaff, D.M., Davies, R.C., McCormick, D.S., Williams, N., Signer, C., 2009. Stationarity scores on training images for multipoint geostatistics. Mathematical Geosciences 41, 447–474.

Plug, L.J., Werner, B.T., 2002. Nonlinear dynamics of ice-wedge networks and resulting sensitivity to severe cooling events. Nature 417, 929–933.

Remy, N., Boucher, A., Wu, J., 2009. Applied Geostatistics with SGeMS: A User's Guide. Cambridge University Press, New York, NY 284 pp.

Renard, P., Allard, D., 2012. Connectivity metrics for subsurface flow and transport. Advances in Water Resources , http://dx.doi.org/10.1016/j.advwatres.2011.12.001.

Renard, P., Straubhaar, J., Caers, J., Mariethoz, G., 2011. Conditioning facies simulations with connectivity data. Mathematical Geosciences 43, 879–903.

Straubhaar, J., Renard, P., Mariethoz, G., Froidevaux, R., Besson, O., 2011. An improved parallel multiple-point algorithm using a list approach. Mathematical Geosciences 43, 305–328.

Strebelle, S., 2000. Sequential Simulation Drawing Structures from Training Images. Ph.D. Dissertation, Stanford University, Stanford, CA, 187 pp.

Strebelle, S., 2002. Conditional simulation of complex geological structures using multiple-point statistics. Mathematical Geology 34, 1–21.

Zhang, J., Zhou, K., Velho, L., Guo, B., Shum, H.Y., 2003. Synthesis of progressively-variant textures on arbitrary surfaces. ACM Transactions on Graphics 22, 295–302.

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

Zinn, B., Harvey, C.F., 2003. When good statistical models of aquifer heterogeneity go bad: a comparison of flow, dispersion, and mass transfer in connected and multivariate Gaussian hydraulic conductivity fields. Water Resources Research 39, 1051.