AGU PUBLICATIONS

Water Resources Research

RESEARCH ARTICLE

10.1002/2013WR015069

Key Points:

- The first use of image quilting approach in geosciences
- Adaptations for exact conditioning and 3-D simulation
- A drastic acceleration compared to existing algorithms

Correspondence to:
K. Mahmud,
kashif.mahmud@unsw.edu.au

Citation:
Mahmud, K., G. Mariethoz, J. Caers, P. Tahmasebi, and A. Baker (2014), Simulation of Earth textures by conditional image quilting, *Water Resour. Res.*, *50*, 3088–3107, doi:10.1002/2013WR015069.

Received 19 NOV 2013
Accepted 19 MAR 2014
Accepted article online 21 MAR 2014
Published online 10 APR 2014

Simulation of Earth textures by conditional image quilting

K. Mahmud[1,2], G. Mariethoz[1,2], J. Caers[3], P. Tahmasebi[3], and A. Baker[1,2]

[1]Connected Waters Initiative Research Centre, University of New South Wales, Sydney, New South Wales, Australia, [2]National Centre for Groundwater Research and Training, School of the Environment, Flinders University, Adelaide, SA 5001, Australia, [3]Stanford Center for Reservoir Forecasting, Department of Energy Resources Engineering, Stanford University, Stanford, California, USA

Abstract Training image-based approaches for stochastic simulations have recently gained attention in surface and subsurface hydrology. This family of methods allows the creation of multiple realizations of a study domain, with a spatial continuity based on a training image (TI) that contains the variability, connectivity, and structural properties deemed realistic. A major drawback of these methods is their computational and/or memory cost, making certain applications challenging. It was found that similar methods, also based on training images or exemplars, have been proposed in computer graphics. One such method, image quilting (IQ), is introduced in this paper and adapted for hydrogeological applications. The main difficulty is that Image Quilting was originally not designed to produce conditional simulations and was restricted to 2-D images. In this paper, the original method developed in computer graphics has been modified to accommodate conditioning data and 3-D problems. This new conditional image quilting method (CIQ) is patch based, does not require constructing a pattern databases, and can be used with both categorical and continuous training images. The main concept is to optimally cut the patches such that they overlap with minimum discontinuity. The optimal cut is determined using a dynamic programming algorithm. Conditioning is accomplished by prior selection of patches that are compatible with the conditioning data. The performance of CIQ is tested for a variety of hydrogeological test cases. The results, when compared with previous multiple-point statistics (MPS) methods, indicate an improvement in CPU time by a factor of at least 50.

1. Introduction

Modeling subsurface heterogeneity is important for predicting the behavior of hydrogeological systems. For more than 50 years, it has been extensively used for the management and the estimation of uncertain water resources [*de Marsily et al.*, 2005]. A broad variety of methods have been developed in geostatistics, aimed at characterizing the spatial structure, or texture, of the variables considered (e.g., hydraulic conductivity or porosity in hydrogeology).

In most cases, spatial covariance or variogram analysis is used to quantify spatial geological continuity. Variograms define correlations between any two points in space. The main critique toward these traditional geostatistical approaches is related to an inability to represent realistically complex and connected geological structures [*Gómez-Hernández and Wen*, 1998; *Journel and Zhang*, 2006; *Sánchez-Vila et al.*, 1996; *Schlüter and Vogel*, 2011; *Western et al.*, 2001]. Object-based approaches are also commonly used in challenging geological environments [*Michael et al.*, 2010; *Ronayne et al.*, 2010], although they often lack flexibility in particular with regards to conditioning data.

An alternative was suggested by *Guardiano and Srivastava* [1993], with the multiple-point geostatistics (MPS) approach based on the assessment of the conditional probability distribution for a simulated value based on a training image (TI). MPS allows the production of images similar to those found with object-based models, with the benefit of ease in conditioning to field data, therefore overcoming some of the major difficulties of object-based models [*Renard et al.*, 2006]. The TI provides a conceptual but explicit representation from which high-order statistics [*Mustapha and Dimitrakopoulos*, 2010] can be extracted. TIs may originate from real data representative of the geology under consideration, or a large unconditional realization of another stochastic simulation technique. *Strebelle* [2002] recommends using TIs developed from expert information, outcrops, or even a geologist's sketching. Therefore, it is not necessary for a TI to be locally constrained to any data or to have similar dimensions as the area under consideration [*Caers and Zhang*, 2004].

The concept of simulating models using multiple-point statistics from a TI seems easy, straightforward, and smart for geologists [*Hu and Chugunova*, 2008]. From its original applications in reservoir modeling [e.g., *Caers et al*., 2003; *Falivene et al*., 2006; *Ronayne et al*., 2008], MPS algorithms have been used for a broad spectrum of fields relevant to hydrogeological applications [*Comunian et al*., 2011; *dell'Arciprete et al*., 2012; *He et al*., 2013; *Huysmans and Dassargues*, 2009; *Jha et al*., 2013b; *Le Coz et al*., 2011] and water resources modeling, such as remote sensing [*Boucher et al*., 2008; *Mariethoz et al*., 2012], climate modeling [*Jha et al*., 2013a], physics of fluids in porous media [*Okabe and Blunt*, 2007; *Tahmasebi and Sahimi*, 2013], even medical imaging [*Pham*, 2012]. In many applications, however, computation time is a major obstruction for MPS to become more routine. In hydrogeology, large-scale 3-D models can contain millions of nodes, and the simulation of a single realization with a continuous variable can take as much time as computing a steady-state transport simulation in this realization. This can adversely affect the performance of any inverse modeling scheme [*Alcolea and Renard*, 2010; *Hansen et al*., 2012; *Khodabakhshi and Jafarpour*, 2013; *Park et al*., 2013; *Zhou et al*., 2012]. In climate and remote sensing applications, there is usually no flow problem, therefore the CPU effort associated to these application is entirely controlled by the stochastic simulation. Being 2-D-temporal, these problems can potentially involve even larger grids than subsurface applications, for example, when downscaling climate models for a 100-year period, at 3-hourly time steps. Therefore, although current methods are flexible and produce satisfying results, there is a need to radically accelerate them. We believe the solution does not reside in faster computers, but rather in improved algorithms, as emphasized by a citation of a Report to the US President and Congress [*PCAST*, 2010]: "While improvements in hardware accounted for an approximate 1000 fold increase in calculation speed over a 15-year time-span, improvements in algorithms accounted for an over 43,000 fold increase."

We believe that such algorithmic developments have already taken place in computer graphics, where there is a need to rapidly generate large amounts of realistic textures for applications such as animation movies or computer games. Texture synthesis is the process of algorithmically constructing a large digital image from a small digital sample image by taking advantage of its structural content. In this field, methods have been developed producing texture which, when viewed by a human eye, appear nonrepetitive and generated by the same underlying process as the original sample image. A recent review [*Mariethoz and Lefebvre*, 2014] shows that the field of computer graphics, and in particular texture synthesis methods, pursue goals similar to MPS: to generate images with patterns similar to the training image (or exemplar), yet show stochasticity (i.e., not just produce a repetitive tiling of the same patterns).

In the field of texture synthesis, an early method was proposed by *Efros and Leung* [1999], using a nonparametric sampling to "grow" texture by enforcing statistics locally, pixel by pixel. All the neighbors are synthesized by computing the conditional distribution to all neighboring pixels, and this is done by searching the training image and searching all likely neighborhoods. The algorithm, essentially similar to the one of *Guardiano and Srivastava* [1993], yields decent outputs for an extensive variety of textures, but is computationally demanding because every pixel has to be synthesized through an exhaustive search of the input image. Instead of using the pixel as the unit of synthesis, *Efros and Freeman* [2001] proposed a 2-D method named image quilting (IQ) that considers 2-D blocks or patches. This is similar as patch-based geostatistical methods [*Arpat and Caers*, 2007; *El Ouassini et al*., 2008; *Faucher et al*., 2012; *Tahmasebi et al*., 2012a; *Zhang et al*., 2006], where texture synthesis can be compared to a jigsaw puzzle. However, major differences exist between patch-based MPS methods and image quilting. Whereas existing geostatistics patch-based methods proceed by searching for patches that fit together with minimal error, IQ modifies the shape of the patches such that the overlap error is further reduced. This is a notable difference, and we believe that using this idea for geostatistical applications bears the potential for increased pattern continuity: it effectively removes the vertical and horizontal artifacts often present with patch-based methods. However, the IQ method does not allow for conditioning. Moreover, IQ is currently only applicable to 2-D cases. Our aim in this paper is to: (1) develop a conditional version of the algorithm, named conditional image quilting (CIQ), (2) to extend it to 3-D, and (3) to test its applicability to traditional geostatistical modeling problems.

We first present an outline of the IQ method in its original implementation and outline the changes to make it amenable for subsurface modeling. In particular, we provide a stochastic selection of candidate patches, define a new way of carving the patches in 3-D, and then test the sensitivity of the parameters for the reproduction of structures observed in natural geological systems. We then propose an adaptation of the method to accommodate conditioning data, including the possibility to consider data with

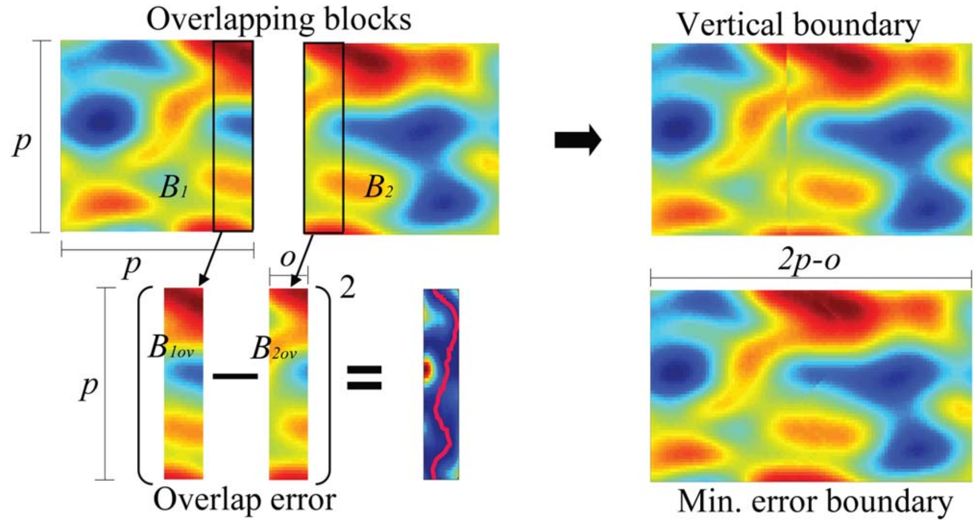


Figure 1. Minimum error boundary cut phenomena to find out the least error path between two overlapping blocks.

measurement uncertainty, which has not been considered in MPS. For cases where exact conditioning is sought, we also implement a template splitting scheme that guarantees reproduction of the hard data.

2. Methodology

2.1. Basic 2-D Unconditional Image Quilting Algorithm

Efros and Freeman [2001] originally presented a 2-D unconditional IQ algorithm to synthesize a texture by using a given example. The key idea is to generate similar texture in 2-D by considering small pieces of existing textures and then sewing these pieces together in a coherent manner. Similar to existing patch-based algorithms, IQ works by assembling patches, one by one, according to a unilateral path. However, patches are not only selected with the minimum overlap error, they are also cut such that the overlap error is minimized. In 2-D, the cutting is accomplished using Dijkstra's algorithm [*Dijkstra*, 1959], while a specific cut function is developed for 3-D cases. As a result, an unconditional IQ simulation consists of the following steps (here "patch" and "area" correspond to 2-D simulations while "block" and "volume" are their 3-D analogs):

Unconditional IQ: Inputs: training image TI, patch size or tile size p, overlap size o, number of replicates ε, and simulation grid SG.

1. Start the simulation at the top left corner of the SG by placing a square patch of size p taken from a random location in the TI.

2. For each subsequent patch along an unilateral path:

a. Identify the overlap areas with previously simulated patches, consisting of o nodes either on the left, top or both left and top areas of the patch.

b. Perform a convolution of the TI and each overlap area. This results in the sum of square differences (denoted E_t) for each location in the TI.

c. Randomly draw a location in the TI among the ε locations with the smallest error E_t, and paste the corresponding area of the TI in the SG.

d. Define the optimal cut of the patch using Dijkstra's algorithm in 2-D or our modified version in 3-D (described below). Retain the old values on the left (top) side of the cut, and paste the new ones on the right (bottom) side of the cut (Figure 1).

In step 2c, the original image quilting algorithm by *Efros and Freeman* [2001] uses the single best matching patch in the TI. In our approach, we rank the candidates and then randomly take one patch among the best

ε positions. This approach enhances the stochasticity of the realizations by avoiding the case where, in a given situation, the same best identical patch is used, leading to exact copying of the training image (we discuss this problem, which we name verbatim copy, in the following sections). If the patch chosen is suboptimal with regard to the overlap regions, some incompatibilities may occur. In that case, the boundary cut procedure removes such inconsistencies by setting the boundary at a location that minimizes the overlap errors. Note that until step 2d, IQ shares similarities with previous patch-based methods [*El Ouassini et al.*, 2008; *Parra and Ortiz*, 2011; *Tahmasebi et al.*, 2012a], notably the use of an overlap region. However, optimally cutting the patches produces significantly improved pattern continuity, as shown in Figure 1. Without considering a minimum error boundary cut, vertical and horizontal artifacts can appear, which can result in discontinuities in the structures, and may affect flow and transport properties of the generated medium. In addition, the fact that the patches are cut and assembled in a coherent manner results in new patterns being created. With most other MPS algorithms, one is limited to only using the patterns present in the TI. This causes problems such as loss of coherence when no matching pattern is found in the TI.

2.2. Finding the Minimum Error Cut in 2-D

The optimal cut is computed by the following pseudocode [*Dijkstra*, 1959]. The procedure for a vertical cut is described. Its adaptation for a horizontal cut is evident.

2-D minimum error cut: Inputs: Two overlapping patches B_1 and B_2 with overlapping areas B_1^{ov} and B_2^{ov}.

1. Define an error surface $\mathbf{e}=(B_1^{ov}-B_2^{ov})^2$, having a rectangular size *p* by *o*.

2. Compute the cumulative minimum error along the cutting direction:

```
for each row $i$ in e (with $i=2...p$), do
    for each column $j$ in e, with $j=1...o$, do
        Calculate the cumulative minimum error E using the 3 closest pixels on the previous row (or
        2 closest pixels if on an edge):
            $\mathbf{E}_{i,j}=\mathbf{e}_{i,j}+\min(\mathbf{E}_{i-1,j-1},\mathbf{E}_{i-1,j},\mathbf{E}_{i-1,j+1})$; if $j=2\dots(o-1)$
            $\mathbf{E}_{i,j}=\mathbf{e}_{i,j}+\min(\mathbf{E}_{i-1,j},\mathbf{E}_{i-1,j+1})$;  if $j=1$
            $\mathbf{E}_{i,j}=\mathbf{e}_{i,j}+\min(\mathbf{E}_{i-1,j-1},\mathbf{E}_{i-1,j})$;  if $j=o$
    end
end
```

3. Identify the coordinate *k* corresponding to the entry with smallest value on the last row of **E**. This location corresponds to the arrival point of a path of minimum cost through the error surface.

4. Trace back the minimum values for each row *i*, going backward (with $i = p$-1...1) and each time identify the cutting path as $\min(\mathbf{E}_{i,k-1},\mathbf{E}_{i,k},\mathbf{E}_{i,k+1})$.

2.3. Finding the Minimum Error Cut in 3-D

Dynamic programming aims at finding a shortest 1-D path. A 1-D path is sufficient to cut in two parts a 2-D image, however in 3-D a cut surface is necessary. Applying the same algorithm for a 3-D block would result in a 1-D minimum cost path: a hairline running through the block, which would be of no help for cutting the block. We, therefore, modified the dynamic programming algorithm described in section 2.2 to obtain a continuous 2-D surface that cuts the overlapping volume in two sections. This 2-D cut needs to correspond to a minimum error surface: that is, the cut surface must encompass as many 1-D minimum cost hairlines as possible. Such a cut is obtained by the procedure described below:

3-D Minimum error cut: Inputs: Two overlapping blocks B_1 and B_2 of size p^3 with overlapping areas B_1^{ov} and B_2^{ov}.

1. Define an error volume $\mathbf{e}=(B_1^{ov}-B_2^{ov})^2$, having a size *p* by *p* by *o*.

2. Compute the cumulative minimum error along the cutting direction:

```
for each layer $k$ in e (with $k=2...p$), do
    Layer $k$ has ($p$*$o$) voxels. Denote the coordinates of a voxel on this layer [$i$,$j$], with $i=1...p$ and
    $j=1...o$.
    for each voxel [i,j] in layer k, do
        Calculate the cumulative minimum error E using the 9 closest voxels on the previous layer $k-1$:
        $\mathbf{E}_{i,j,k}=\mathbf{e}_{i,j,k}+\min(\mathbf{E}_{i-1,j-1,k-1},\mathbf{E}_{i-1,j,k-1},\mathbf{E}_{i-1,j+1,k-1},\mathbf{E}_{i,j-1,k-1},\mathbf{E}_{i,j,k-1},\mathbf{E}_{i,j+1,k-1},\mathbf{E}_{i+1,j-1,k-1},\mathbf{E}_{i+1,j,k-1},\mathbf{E}_{i+1,j+1,k-1})$
        If $\mathbf{e}_{i,j}$ is located on an edge, correspondingly less than 9 neighbors are available on the previous layer.
      end
end
```

3. Consider only the last layer of **E**. Each **E** value on this last layer represents the cost of the shortest 1-D path throughout the block. We isolate this last layer and perform a 2-D cut (section 2.2) to find the minimum error cut in $\mathbf{E}_{i,j,p}$. This 2-D cut encompasses the maximum number of arrival points of 1-D shortest paths throughout the block.

4. Propagate this 2-D cut throughout the block by going backward (with $k = p\text{-}1\ldots1$), each time identifying the cutting path as $\min(\mathbf{E}_{i-1,j-1,k-1},\mathbf{E}_{i-1,j,k-1},\mathbf{E}_{i-1,j+1,k-1},\mathbf{E}_{i,j-1,k-1},\mathbf{E}_{i,j,k-1},\mathbf{E}_{i,j+1,k-1},\mathbf{E}_{i+1,j-1,k-1},\mathbf{E}_{i+1,j,k-1},\mathbf{E}_{i+1,j+1,k-1})$

Note that only the 9 closest voxels of the previous layer (k-1) are considered for calculating the minimum error cut, while the neighboring voxels of layer k are not considered. This could potentially lead to discontinuities in the cut surface, resulting in artifacts. We address this issue by forcing a continuous cut surface. This is accomplished by not allowing the cut of a layer to be more than one voxel away from the cut of the previous layer.

In both 2-D and 3-D cases, the most important parameters are patch size *p*, the overlap size *o* and the number ε of replicates considered. The minimum boundary cut occurs in the overlap regions, and therefore if those are larger (larger *o* value), there is greater latitude for cutting the patches and creating new patterns, different from the ones present in the TI. In MPS, the pattern reproduction in the realization is dependent on the parameterization of the simulation algorithm. Similar analyses to identify the optimal parameters of training image-based stochastic methods have been carried for other algorithms [*Liu*, 2006; *Meerschman et al.*, 2013]. Similarly, in section 4 of this paper, we present a detailed sensitivity analysis of the parameters of our approach.

2.4. Influence of Minimum Error Boundary Cut

Optimally cutting the patches produce significantly improved pattern continuity, as shown in real 2-D and 3-D examples (Figure 2). The 2-D TI shown in Figure 2 represents a binary image of a reservoir consisting of long connected sinuous channels. This model has been used extensively in the past to test algorithms such as SNESIM [*Strebelle*, 2002], FILTERSIM [*Zhang et al*., 2006], DISPAT [*Honarkhah and Caers*, 2010], CCSIM [*Tahmasebi et al*., 2012a], DS [*Mariethoz et al*., 2010], as well as for validating extensions of the MPS approach such as imposing connectivity constraints [*Renard et al*., 2011] or modeling specific edge properties [*Huysmans and Dassargues*, 2011]. For the 3-D case, we have used a TI generated with the FFT-MA (fast fourier transform-moving average) method [*Goovaerts*, 1997; *Le Ravalec-Dupin et al*., 2000] using a Gaussian variogram model. Using such variogram creates smoothly varying and therefore challenging test training images. The parameters used for these tests are mentioned in Figure 2. If a vertical boundary cut, numerous artifacts will appear in the realizations (Figure 2b). This can result in discontinuity of the structures, and may not represent the actual pattern of the TIs. By performing the minimum error boundary cut, the discontinuities no longer occur in the generated realizations (Figure 2c). Note that in the case of discrete variables (such as in the upper example of Figure 2), there may be several paths presenting exactly the same cost. In order to randomly choose one of these paths of equal cost, we add a uniform random noise of a very small magnitude to the error surface **e**. This results in minute variations in cost of paths that would otherwise be equivalent, and then the resulting minimum cost path is chosen.

2.5. Patch Size Randomization

Figure 3 illustrates the ensemble average of 100 realizations obtained with 2-D IQ. The average displayed in Figure 3b clearly shows an artifact near the patch boundaries. This issue is common to many patch-based

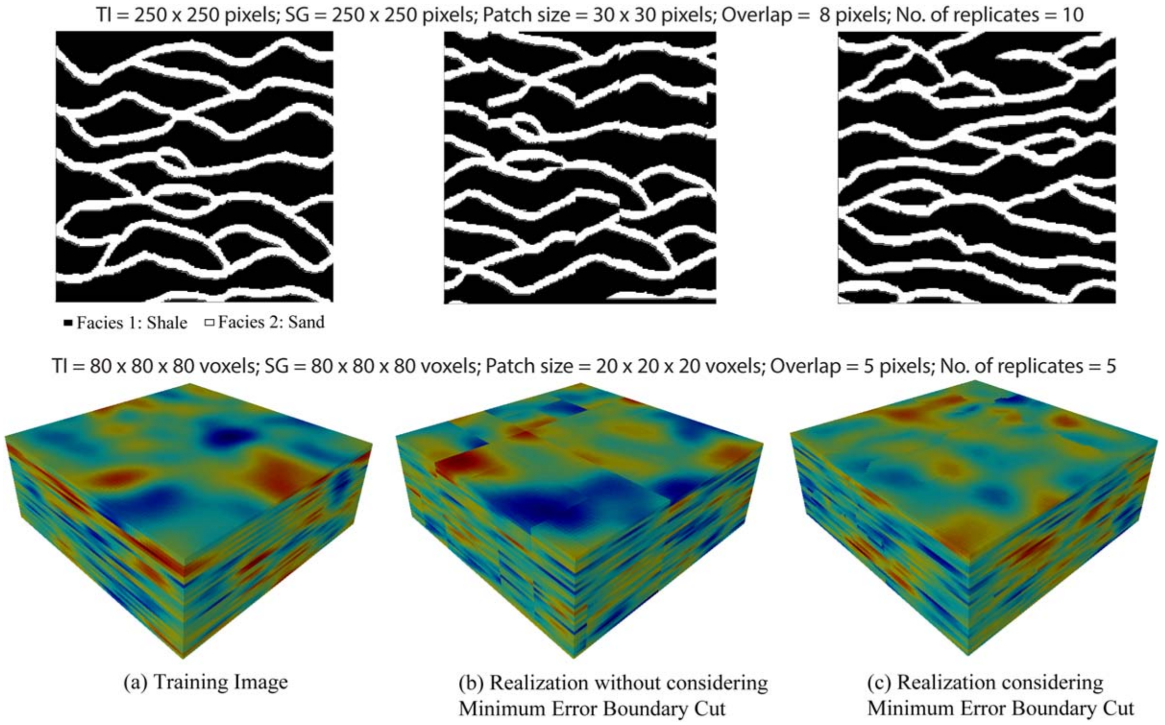

TI = 250 x 250 pixels; SG = 250 x 250 pixels; Patch size = 30 x 30 pixels; Overlap = 8 pixels; No. of replicates = 10
■ Facies 1: Shale
□ Facies 2: Sand
TI = 80 x 80 x 80 voxels; SG = 80 x 80 x 80 voxels; Patch size = 20 x 20 x 20 voxels; Overlap = 5 pixels; No. of replicates = 5
(a) Training Image
(b) Realization without considering Minimum Error Boundary Cut
(c) Realization considering Minimum Error Boundary Cut

Figure 2. Comparison between realizations with and without considering the minimum error boundary cut.

simulation methods [*Honarkhah*, 2011]. If many realizations are generated with identical patch size, the overlap areas between patches will be at the same location for all realizations. New patterns are generated in these overlap areas, either because there is an overlap error (such as with FILTERSIM or SIMPAT) or in our case, because of the cut procedure. Hence, the overlap regions may typically display patterns different from the rest of the realizations. The result is a possible local bias in the ensemble average at these locations. To deal with this issue, *Honarkhah* [2011] proposes to randomly shift by some small amount the grid origin for each realization. Here we achieve a similar result by randomizing the patch size for each realization in a small interval within a range of ±10% of the user-defined value. This causes the overlap areas to be at a

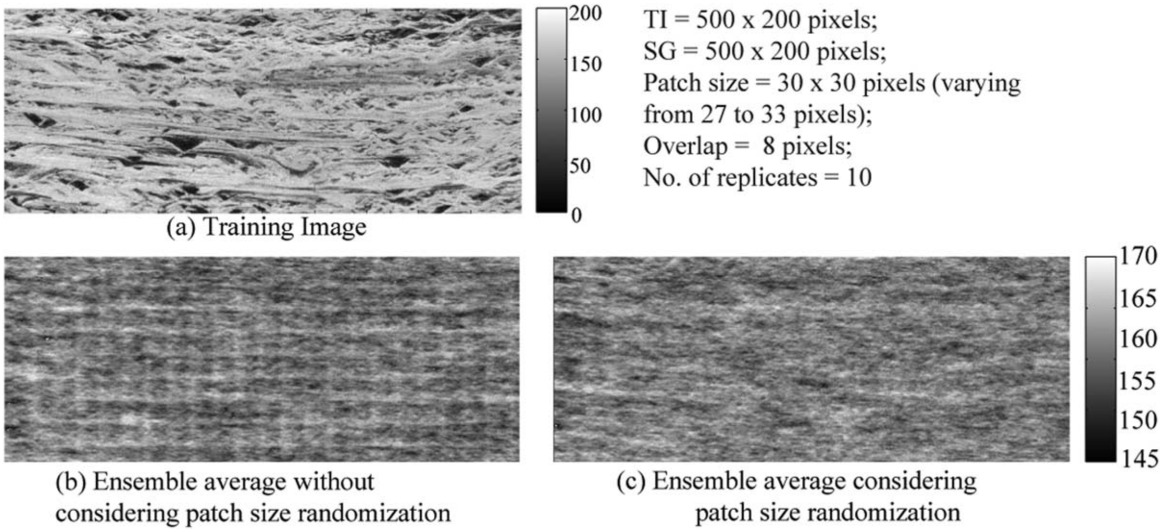


Figure 3. Comparison between ensemble averages with and without patch size randomization.

different position for each realization. Figure 3c shows the effect of this patch size randomization on the ensemble average; a comparison is made between the ensemble average with and without patch size randomization. The training image (Figure 3a) is further used and described in section 5.3. The parameters used for this analysis are listed in the figure.

3. Conditional Image Quilting (CIQ)

The method originally proposed by *Efros and Freeman* [2001] is a 2-D unconditional texture synthesis method. We described above how it can be extended to 3-D, now we address the challenge to adapt it for accommodating conditioning hard data. Constraining models to data is not straightforward and several methods, such as Boolean simulation, present difficulties in this regard [*Garcıa-Morales*, 2003]. Here we only consider the integration of hard data (i.e., no soft constraints), and present two ways of accomplishing this conditioning: the first one, by patch selection, is straightforward but only achieves approximate conditioning. We argue that it can be used when the data do not have to be matched precisely, for example, in the presence of uncertain measurements. It is, however, not applicable in the canonical case where the data have to be matched exactly. For this case, we implemented a dynamic template splitting approach that achieves exact conditioning by reducing the patch size. Both approaches are presented below.

3.1. Conditioning by Patch Selection Method

We propose to condition by modifying the convolution step used to select candidate patches from the TI (step 2b of the IQ algorithm). Instead of selecting candidate patches solely based on their overlap with previously simulated areas, we also consider compatibility with conditioning data. With this selection method, the TI is searched for blocks that agree both with their neighbors along the section of overlaps and at the same time all hard data available within the block. The error E_t relative to a block thus becomes

$$E_t = \frac{1-w}{O_{size}} E_q + \frac{w}{d} E_c. \tag{1}$$

The total error E_t is calculated by considering both the overlap error E_q and the conditioning error E_c. The importance of each term is determined by the weighting parameter w, which lies in the interval [0 1]. For both terms to be comparable, each has to be normalized by the number of pixels considered. Hence, O_{size} is the number of pixels or voxels in each overlapping region and d is the number of data points in the patch.

From equation (1), it is clear that w determines the trade-off made between accuracy of conditioning and consideration of the overlapping areas. Hence, accurate conditioning comes at the price of possible artifacts between patches. In the case of a continuous variable with real values, it is not possible to condition exactly due to the finite size of the TI (and hence limited pattern set). In that case we condition approximately with an error tolerance. The approach we propose in the context of patch selection conditioning is to adjust w by trial-and-error, although this could also be accomplished by a 1-D optimization. w is calibrated to a given error variance, as will be demonstrated in the example section.

3.2. Conditioning by Template Splitting

In order for conditioning to be exact, we introduce an extension of the adaptive iterative template splitting approach [*Tahmasebi et al*., 2012a]. The main idea is that it is easier to honor the hard data with a smaller template size. This is particularly the case when several conditioning data points fall in a single template, and when the pattern formed by these data has no equivalent in the TI. In cases where there is no suitable block that exactly matches the data, the template is split in four (or in eight in a 3-D case). The resulting smaller templates will each contain less data, and therefore will be easier to condition. The operation is carried out recursively, until eventually the templates are so small that they can only contain a single datum.

Figure 4 shows the steps of the template splitting procedure. The simulation proceeds as in the unconditional case until we get a problematic data configuration with more than one conditioning point. In this example, the first patch has six data points. If we perform image quilting as before, it will be difficult, or even impossible, to exactly honor all data points. To obtain exact conditioning, we split the patch in four equal subtemplates with overlapping areas shown in the Figure 4b. This corresponds to a nested version of

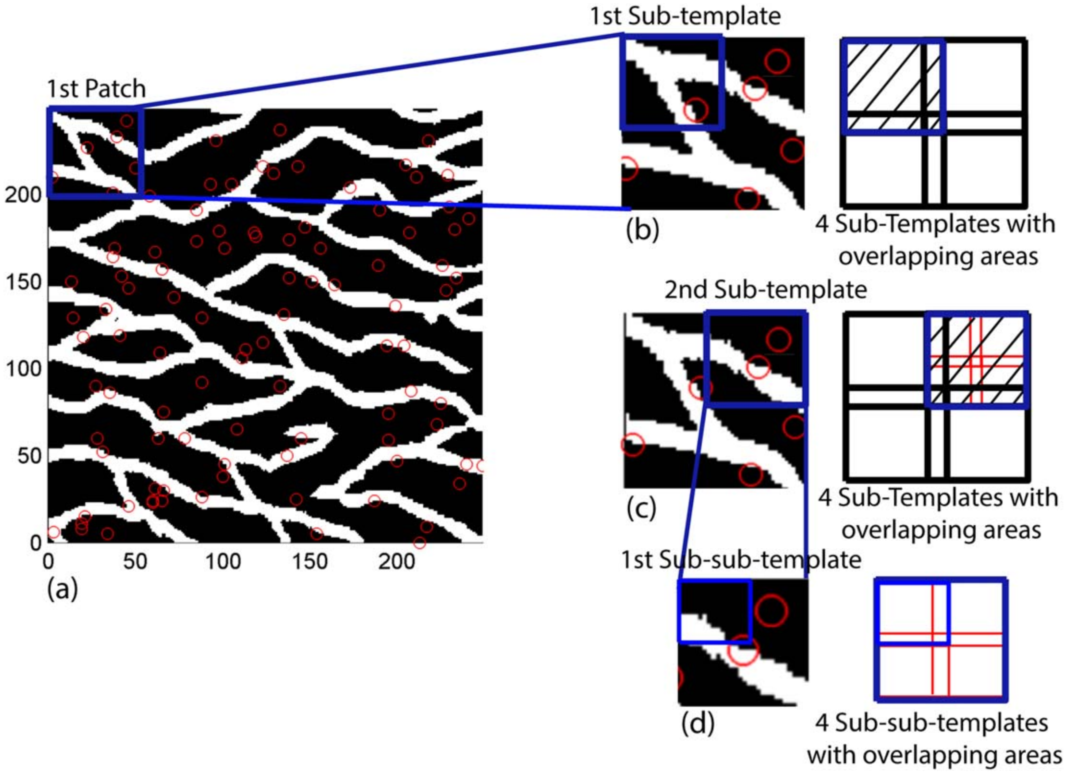

1st Sub-template
1st Patch
200
150
100
50
0
0 50 100 150 200
(a)
(b)
4 Sub-Templates with overlapping areas
2nd Sub-template
(c)
4 Sub-Templates with overlapping areas
1st Sub-sub-template
(d)
4 Sub-sub-templates with overlapping areas

Figure 4. Template splitting concept for a TI with conditioning data points.

image quilting, except that now we have to consider both the overlap with the previously simulated patches and the overlap within the different subtemplates. The splitting results in less conditioning points falling in each subtemplate. If this reduction is not sufficient to obtain exact conditioning, further splitting is carried out (Figure 4d) until all points within the different subtemplates are exactly conditioned.

4. Parameterization and Computational Performance

In this section, we apply our method on the TI shown in Figure 5a, consisting of gray scale values of a satellite image of the Lena Delta in (Russia). The size of the TI is 362 × 362 pixels. Using this application example, we perform a sensitivity analysis of IQ parameters and compare our approach with the established direct sampling method [*Mariethoz et al.*, 2010].

4.1. Parameters Sensitivity Analysis

The parameters of the IQ are p (patch size), o (overlap size), and ε (number of replicates). We study the effect and the sensitivity of these parameters on unconditional realizations, considering CPU cost and quality of the realizations obtained. Sensitivity analysis is carried out by studying deviations from the following set of standard parameters: $p = 50$ pixels, $o = 15$ pixels, $\varepsilon = 10$. One can also find other sets of standard parameters. The size of the output realizations is 362 × 362 pixels.

4.1.1. Sensitivity to Patch Size

The patch size is the main parameter controlling both the speed of IQ and the quality of the results. Figure 5 shows different realizations obtained by varying the patch size with constant $o = 15$ pixels and $\varepsilon = 10$.

A large patch allows reproducing the large-scale features and leads to faster simulation times. The use of a large patch results in literal copying of TI regions in the simulated model (termed verbatim copy, see Figure 5h), therefore causing low variability between the generated realizations. Additionally, the usage of an overly large patch size will result in more difficult conditioning on denser datasets (as discussed in section 5.3).

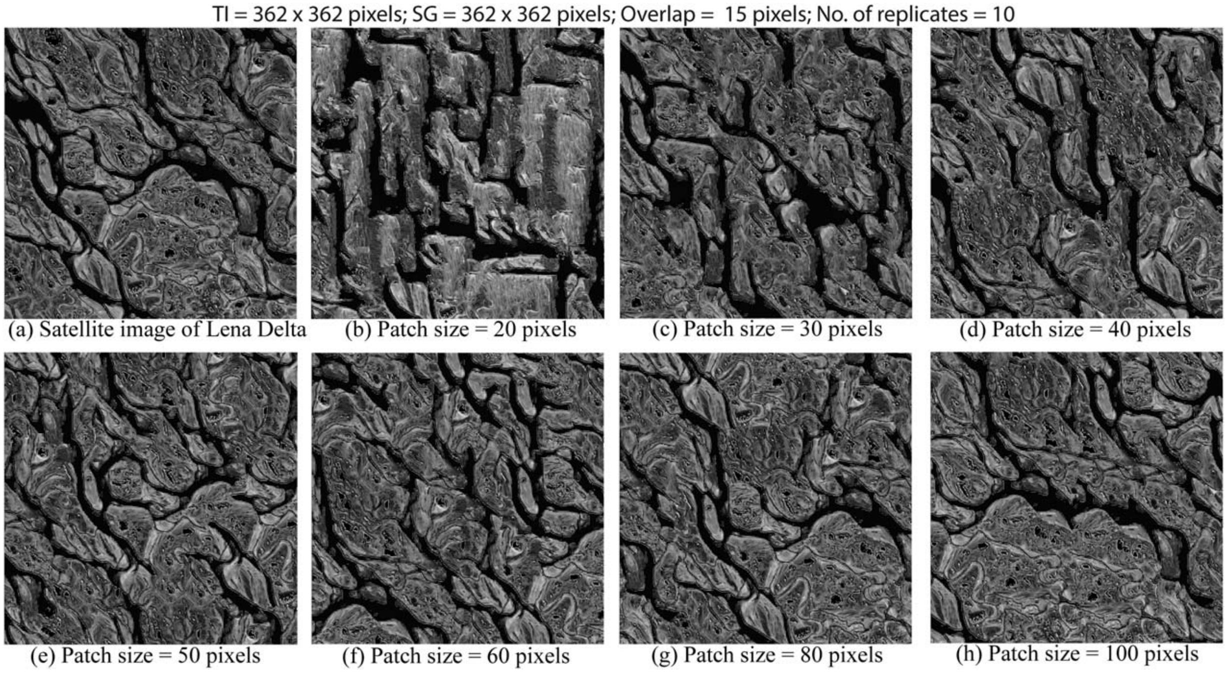

TI = 362 x 362 pixels; SG = 362 x 362 pixels; Overlap = 15 pixels; No. of replicates = 10
(a) Satellite image of Lena Delta
(b) Patch size = 20 pixels
(c) Patch size = 30 pixels
(d) Patch size = 40 pixels
(e) Patch size = 50 pixels
(f) Patch size = 60 pixels
(g) Patch size = 80 pixels
(h) Patch size = 100 pixels

Figure 5. Training image (Landsat 7 image, USGS/EROS and NASA Landsat Project) and realizations obtained using different patch sizes.

On the other hand, using very small patches may lead to the loss of large-scale structures, including connected features creating main flow paths in flow and transport problems (Figure 5b). In the case of Figure 5, artifacts in the connected structures are present with patch sizes under 40 pixels. Conversely, very large patches necessarily result in the occurrence of verbatim copy (Figure 5h). As a consequence, we recommend using the largest patch size not incurring verbatim copy, corresponding to 50 pixels in our test case. A rigorous way of determining the optimal patch size in different cases could be to use the entropy plots method proposed by *Honarkhah and Caers* [2010].

4.1.2. Sensitivity to Overlap Size

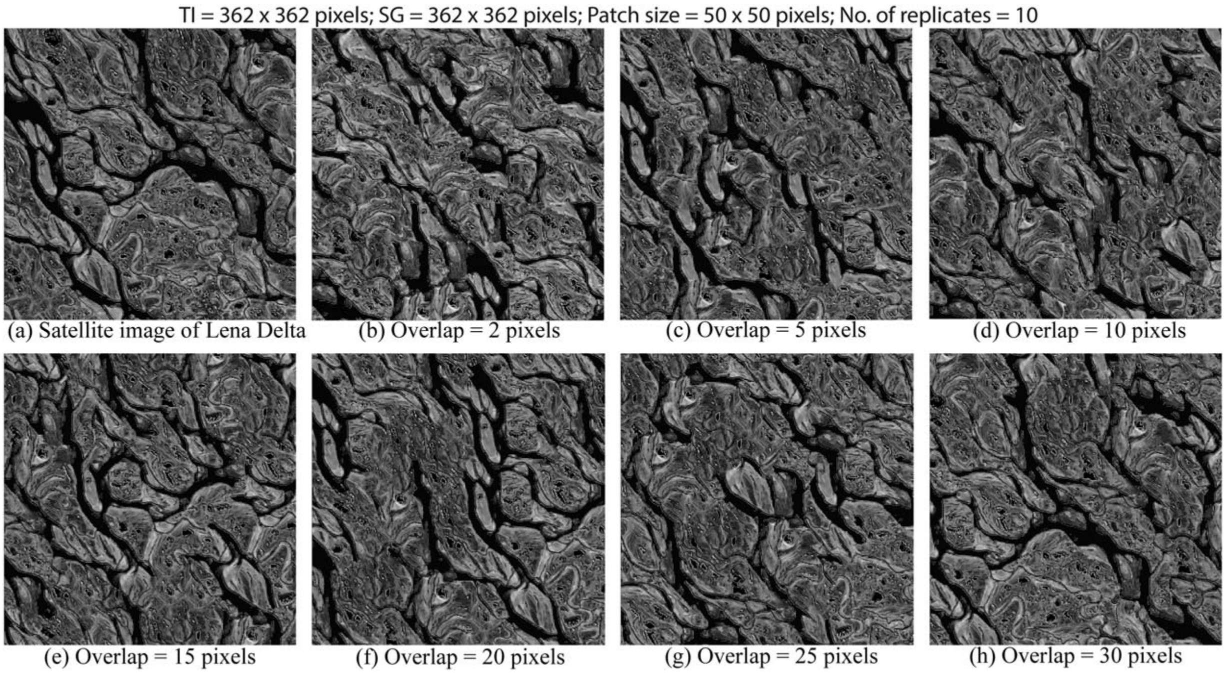
Figure 6 shows different realizations obtained by varying the amount of overlap between patches with constant $p = 50$ pixels and $\varepsilon = 10$. Using very large overlap regions leads to a verbatim copy of the training image because the constraints on the overlap region become so strong that all candidate patterns come from the same area of the TI (Figure 6h). Furthermore, the choice of smaller o regions may result in models that do not represent the continuity of the TI. Based on these results and the CPU times, we found that the optimal overlap size is between 1/3 and 1/4 of the patch size.

4.1.3. Sensitivity to Number of Replicates

Figure 7 shows different realizations obtained by varying the number of replicates ε. Other parameters are fixed to $p = 50$ pixels and $o = 15$ pixels. We find that the "verbatim copy" of the training image occurs when the number of replicates is too low (typically 1), as shown in Figure 7b. In that case, the simulation algorithm duplicates the TI patch by patch, each time following the best candidate in the TI. As the number of replicates increases, the variability between realizations and TI reflects the space of possible patch arrangements and verbatim copy no longer occurs. In general, the occurrence of verbatim copy is avoided by using large ε (larger than 5), and avoiding very small o or very large p. Using a large number of replicates is suboptimal as the structures appear globally less continuous due to the acceptance of patches with a relatively high mismatch (Figures 7g and 7h).

4.2. Comparison with Direct Sampling

We compare IQ with Direct Sampling (DS) for a realistic case, and show that IQ produces at least a similar degree of patterns reproduction, but at a fraction of the computational cost. The DS is a pixel-based MPS algorithm that we use for comparison with IQ because it can also be used with both continuous and



TI = 362 x 362 pixels; SG = 362 x 362 pixels; Patch size = 50 x 50 pixels; No. of replicates = 10

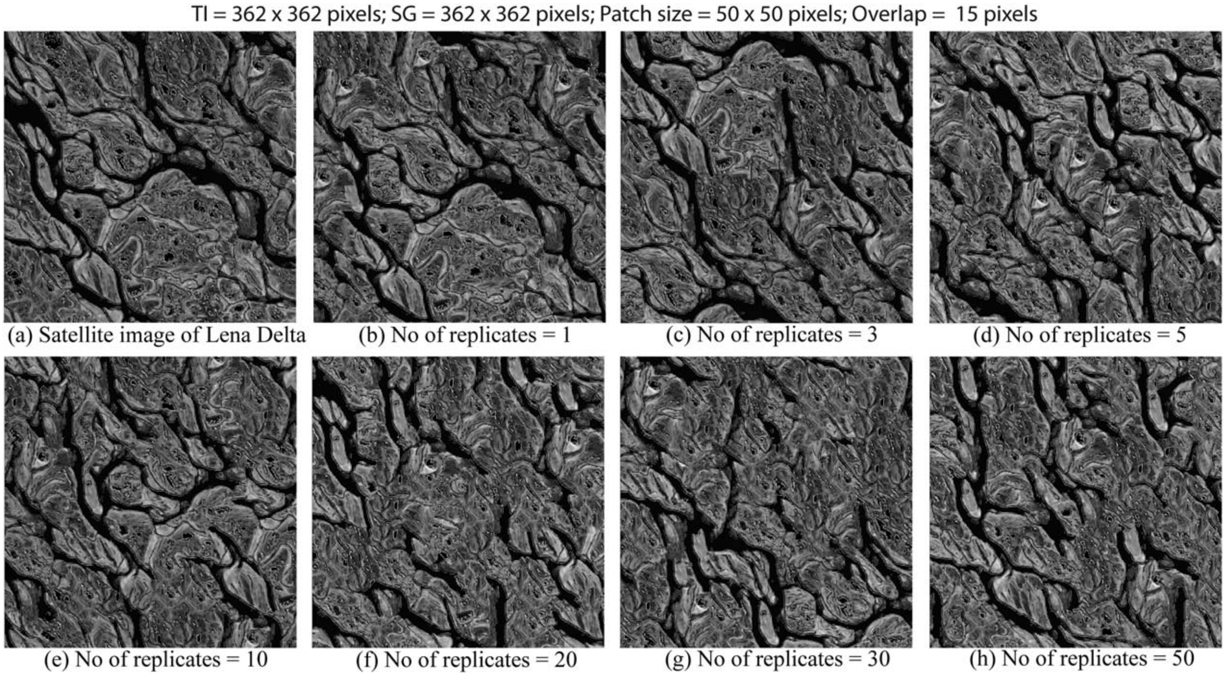
(a) Satellite image of Lena Delta (b) Overlap = 2 pixels (c) Overlap = 5 pixels (d) Overlap = 10 pixels

(e) Overlap = 15 pixels (f) Overlap = 20 pixels (g) Overlap = 25 pixels (h) Overlap = 30 pixels

Figure 6. Different realizations due to the variation in overlapping amount to allow between patches.

categorical variables [*Mariethoz et al*., 2010]. The method is based on a sampling technique introduced by [*Shannon*, 1948], which consists in scanning the training image for a given conditioning data event. At the first occurrence of the data event searched for, the scan is interrupted and the corresponding data event is accepted. Such an interrupted scan is statistically equivalent to scanning the entire TI.

The Lena training image (in gray scale levels) was used to compare IQ and DS. The comparison is based on histograms and variograms of the generated realizations. The parameters used for both cases are listed in

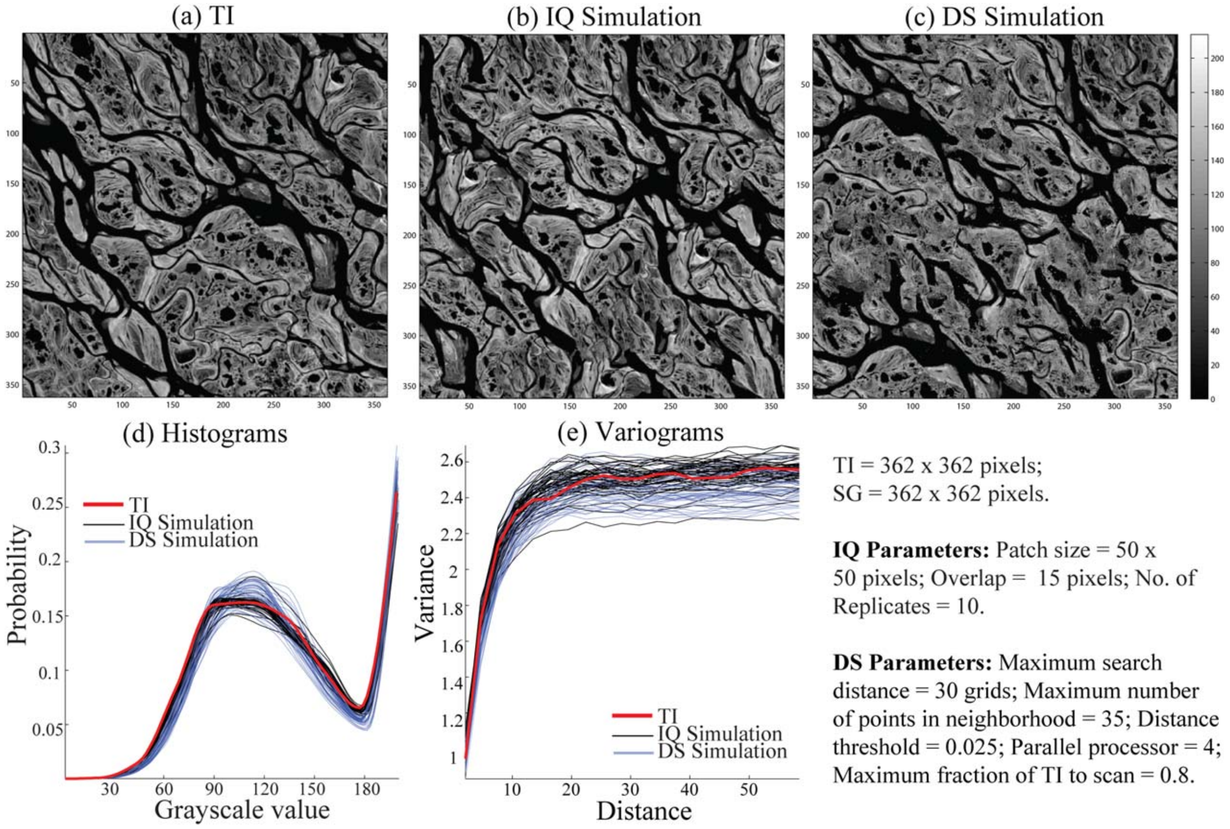


TI = 362 x 362 pixels; SG = 362 x 362 pixels; Patch size = 50 x 50 pixels; Overlap = 15 pixels

(a) Satellite image of Lena Delta (b) No of replicates = 1 (c) No of replicates = 3 (d) No of replicates = 5

(e) No of replicates = 10 (f) No of replicates = 20 (g) No of replicates = 30 (h) No of replicates = 50

Figure 7. Different realizations obtained by varying the number of replicates.



TI = 362 x 362 pixels;
SG = 362 x 362 pixels.

IQ Parameters: Patch size = 50 x 50 pixels; Overlap = 15 pixels; No. of Replicates = 10.

DS Parameters: Maximum search distance = 30 grids; Maximum number of points in neighborhood = 35; Distance threshold = 0.025; Parallel processor = 4; Maximum fraction of TI to scan = 0.8.

Figure 8. Comparison between IQ and DS in terms of histograms and variograms.

the figure. Figure 8a shows the TI for gray scale values. The histograms and variograms for the TI and the generated 50 realizations are shown in Figures 8d and 8e, indicating that both IQ and DS are equally able to reproduce the basic statistics of the TI. These results confirm that in terms of spatial statistics both methods produce similar results. In the next section, we will show that IQ provides such results with significantly reduced computational effort compared to DS.

4.3. Algorithm Performance

Algorithm performance is assessed using the same TI shown in Figure 8a. The average time per realization for different *p*, shown in Figure 9a, decreases rapidly with increasing patch size. The average time per realization for various *o* sizes rises exponentially with increasing overlapping region, as shown in Figure 9b, because the convolutions involve larger regions and also because the minimum error cut is calculated on a larger area (although this factor is not critical because Dynamic Programming scales linearly in term of computing time). The number of replicates does not affect CPU time since errors are precalculated for all possible patches during the convolution step.

A comparison between both unconditional and conditional IQ with DS is presented in Figure 10. Figures 10a and 10b show the CPU time corresponding to the simulation grid for 2-D and 3-D cases, respectively, expressed size as total number of pixels (all calculations are carried out on a notebook computer with a 2.20 GHz processor). Note that a different time axis is used for each simulation method and the same DS parameters are used provided in Figure 8. With both IQ and DS the relationship scales almost linearly, however the slope is very small for IQ. The computational performance of conditional IQ for 2-D cases is also presented in Figure 10c. Here the simulation time of DS is not affected by adding conditional data, but the horizontal line representing DS is higher than the curve for IQ. IQ/CIQ is preferable for cases with large training images or simulations grids and the difference is especially marked for continuous variables. This comparison shows a better performance of IQ over DS, for similar patterns reproduction. Moreover, our implementation of DS is parallel and uses four processors, whereas the IQ implementation is not parallel.

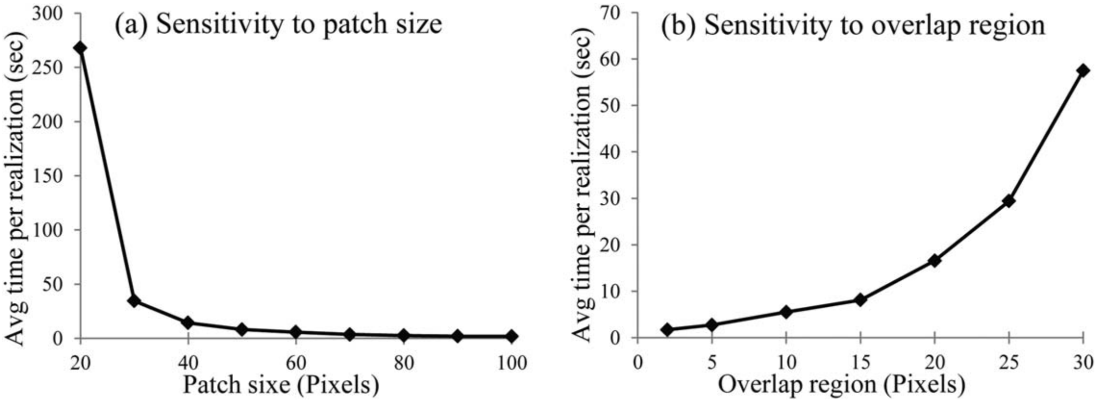


Figure 9. Effect of (left) patch size and (right) overlap region size on CPU cost. Times given are averaged on 10 realizations.

A parallel implementation of IQ (typically parallelizing the convolution step) would yield even better performance than shown in Figure 10.

4.4. The Occurrence of Verbatim Copy

When using training images as a basis for stochastic simulation, it is not possible to retain all spatial properties of the training image because the training image is limited in size and often contains nonrepetitive events. Strictly honoring the statistics of the training image would entail that all its properties, at all scales, need to be reproduced in the realizations. This would leave one possible outcome: the simulation has to be identical as the training image, which is evidently not desirable. When such exaggerated importance is given to the information coming from the training image, verbatim occurs. This mostly occurs with unconditional simulations, because the conditioning data generally cannot be reproduced by using an exact copy of the TI. We note here that verbatim copy is not specific to IQ: it has been observed with other patch-based simulation methods such as FILTERSIM, SIMPAT, Direct Sampling, or CCSIM. For example, comparing Figure 8b and Figure 8c shows that it occurs equally with IQ and DS. Although it is a sign of faithful pattern reproduction, verbatim copy is not desired because it produces an artificially reduced variability between the realizations, which is not compatible with the goal of geostatistics, namely uncertainty modeling.

The only way to completely avoid verbatim copy is to use a sufficiently large and repetitive TI, which may not be possible in all practical applications. For example, the TI of Figure 5a, relatively small, with continuous variable and long connected channels, is the typical case for verbatim copy to occur. Since large TIs are not always available, avoiding verbatim copy in patch-based methods is done by parameterizing the simulation algorithms to create new patterns. The patch-based nature of the method implies that entire blocks are pasted from the TI. Although smaller patches do reduce verbatim copy, it is almost impossible to avoid this

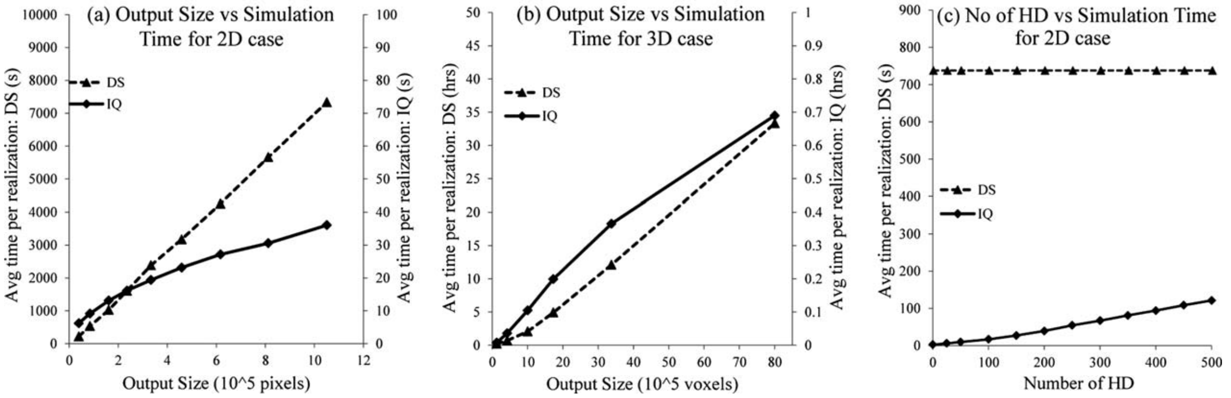


Figure 10. CPU-time comparison for (a and b) different output sizes and (c) different number of hard data. Note that a different time axis is used in plots (a) and (b). Times given are averaged over 10 realizations.

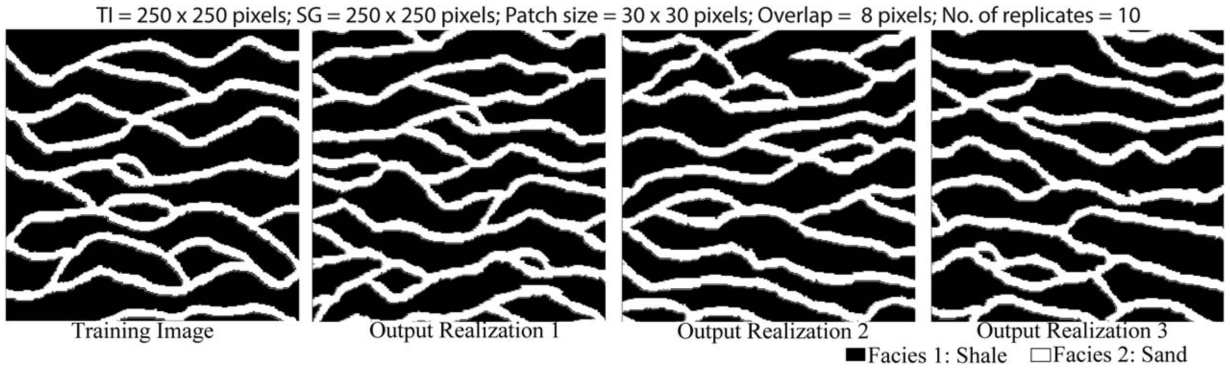

TI = 250 x 250 pixels; SG = 250 x 250 pixels; Patch size = 30 x 30 pixels; Overlap = 8 pixels; No. of replicates = 10
Training Image
Output Realization 1
Output Realization 2
Output Realization 3
■Facies 1: Shale
□Facies 2: Sand

Figure 11. Performance of IQ algorithm for a TI with binary channels.

to some extent with larger patches (see Figures 5e, 6e, and 7d, which we judged optimal but still present identical portions of the TI). To avoid verbatim copy in IQ, we provide the following recommendations:

1. Retain a high number of replicates (>5) ensuring a nonunique patch selection.

2. Define a large overlap area, resulting in significantly altered patterns along the minimum error boundary (creation of new patterns).

5. Application Examples

To demonstrate the extent of applicability of IQ algorithm, we present in this section examples using both categorical and continuous training images in conditional and unconditional settings. These examples are presented and discussed in the following sections.

5.1. Unconditional Binary 2-D Case

We have used the same 2-D binary TI from Figure 2 with optimum parameters of patch size 30×30 pixels, an overlap region of 8 pixels and $\varepsilon = 10$ replicates to find out the best possible candidates. These parameters give the best balance of CPU speed and realistic simulation with regard to the sensitivity analysis carried out in section 4.1. Simulations obtained with IQ are shown in Figure 11, with an average simulation time of 2.91 s based on 100 realizations. In the TI, the proportion of sand channels is 0.28. We retained a similar proportion such as 0.278, 0.284, and 0.282 for the first, second, and third realization, respectively (shown in Figure 11). We have also computed the indicator variograms of the TI and compared them with those of the generated realizations (Figure 12). In addition, connectivity functions [*Pardo-Igúzquiza and Dowd*, 2003; *Renard and Allard*, 2013] were computed, consisting of the probability of two locations to belong to the same connected component as a function of the distance along both *X* and *Y* directions. The results, displayed in Figure 12, show that the IQ algorithm is able to reproduce fairly well both the variograms and connectivity of the TI.

5.2. Conditional Binary 2-D Example with Patch Selection

We have used the same problem setting as the binary example of Figure 11, with 50 conditioning data points consisting of values taken from an unconditional realization. Figure 13 presents two realizations that honor conditioning points (red circles) as well as the ensemble average based on 100 conditional simulations. We have used the same parameters as in the unconditional IQ example and a weighting factor $w = 0.9$, which resulted in exact conditioning (hard data conditioning in geostatistical jargon). Continuity of the TI structures is well reproduced. Locally, near conditioning data, spatial uncertainty is reduced as shown by the ensemble average of the 100 binary realizations. The uncertainty at the data locations is exactly zero, showing that accurate conditioning is possible by using a sufficiently high *w* value. The average simulation time, based on 100 realizations is 5.84 s, almost double compared to the unconditional case, indicating that conditioning does incur significant overhead and the effect was illustrated in section 4.3. Also, the indicator variograms and the connectivity functions were computed along both *X* and *Y* directions similar to the

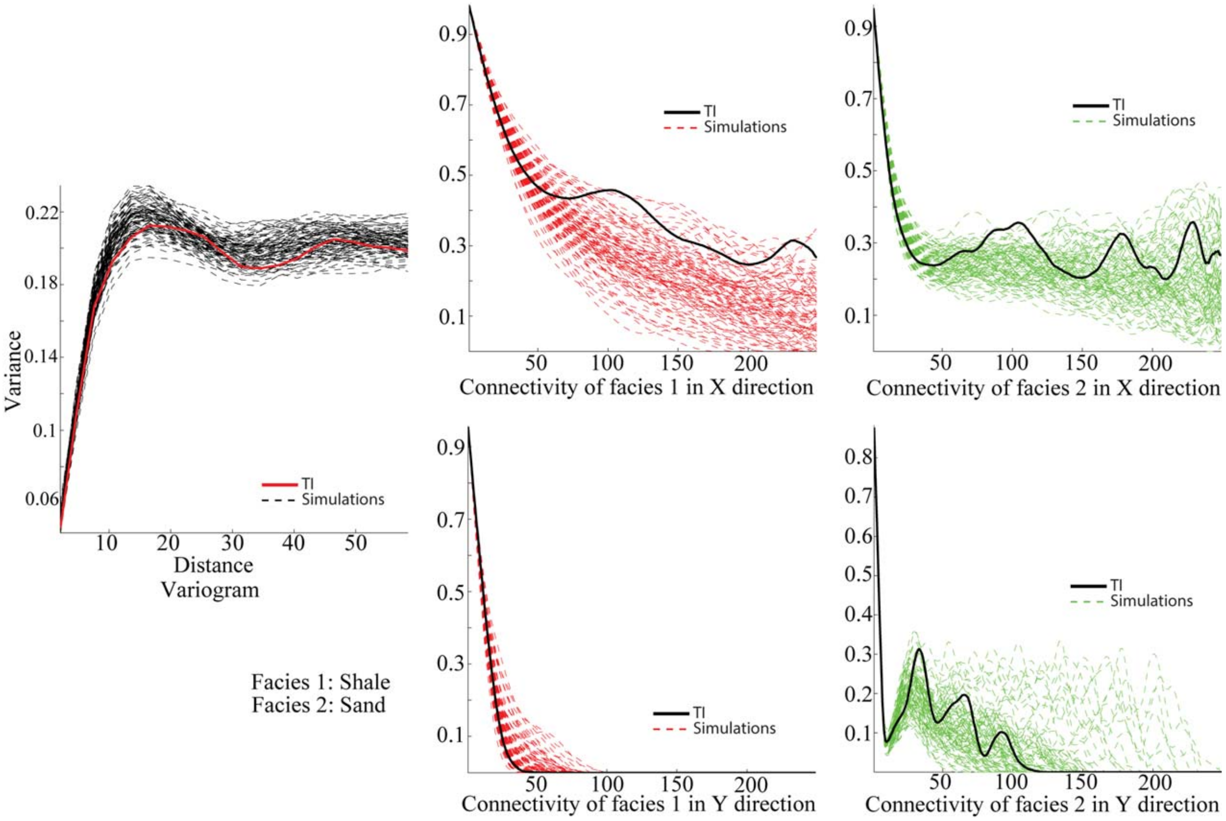


Figure 12. Comparisons of the variograms and the connectivity functions of different facies in both *X* and *Y* directions of several realizations with that of the TI.

unconditional case. The results presented in Figure 14 show that incorporating conditioning data does not come at the expense of patterns reproduction, and that the conditioning is accurate.

5.3. Conditional Continuous 2-D Example with Patch Selection

The next test uses a TI extracted from a flume experiment [*Paola et al.*, 2009]. These are laboratory scale experiments simulating sediment deposition in a realistic physical setting. After the experiment is completed, the resulting volume of sediments is cut into consecutive sections of which photographs are taken. One such photograph is shown in Figure 15a. This TI is used with the 50 conditioning data shown in Figure 15b taken from an unconditional realization. Since the variable is continuous, the goal is that the realizations exhibit a user specified standard deviation at the measurement locations, reflective of the measurement error. This is accomplished by adjusting *w* by trial-and-error. Note that it would be possible to condition exactly by using template splitting. Figure 15 presents the results obtained by adjusting *w* in two cases assuming an average measurement error of 0 and an error standard deviation of 3.0 in the first case (left column) and 0.5 in the second case (right column). After adjustment by trial-and-error, the *w* values obtained are $w=0.1$ to obtain an error standard deviation of 2.9854 (case 1) and $w=0.9$ to obtain an error standard deviation of 0.49529 (case 2), which corresponds reasonably well to the target measurement error. In both cases, the mean error to the conditioning data is very small (-0.032093 for case 1 and -0.01908 for case 2), indicating no systematic bias in conditioning. The cross-plots (Figure 15f) further illustrate that our method allows considering a tolerance to the conditioning data, which has so far not been considered in the context of MPS methods.

The adjustment is based on the conditioning error considered over an ensemble of 100 realizations. This adjustment by trial-and-error is done in a matter of minutes because the image quilting method is computationally efficient, as shown in section 4.3. The average simulation time based on 100 realizations is 15 s, close to the unconditional case, and comparable to the binary case. This is an important fact because it has

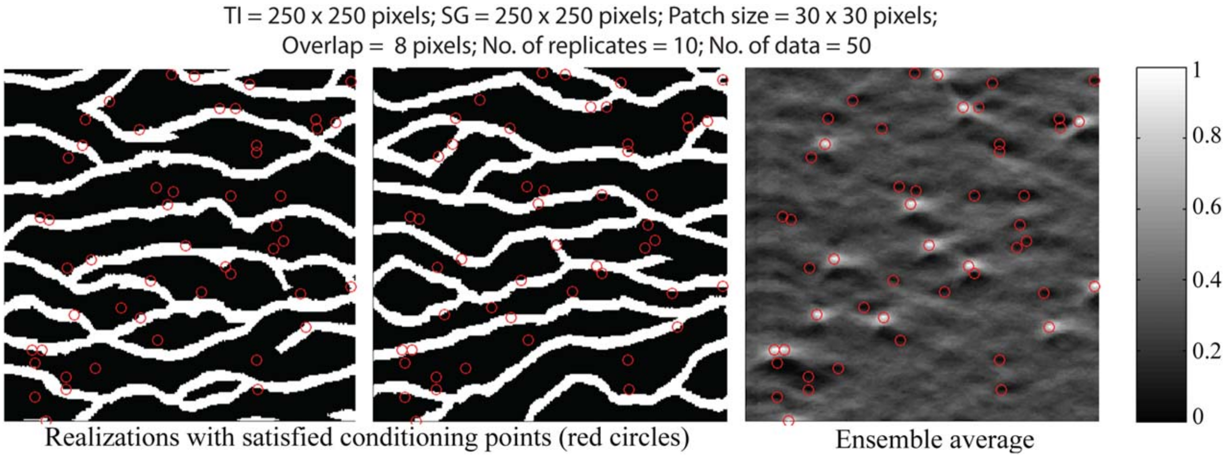


Figure 13. Conditional simulation for a binary case with 50 data points.

been observed with other approaches that continuous simulation can be very computationally inefficient compared to the categorical case [*Meerschman et al*., 2013]. The ensemble averages and the standard deviation maps (Figures 15d and 15e), based on 100 realizations, show a slight variance reduction in the vicinity of the data.

5.4. Conditional Continuous 2-D Example with Template Splitting

The template splitting scheme has been tested with the same TI of Figure 15a. For the analysis, 100 realizations were generated using the same parameters and conditioning data as in section 5.3. For calculating

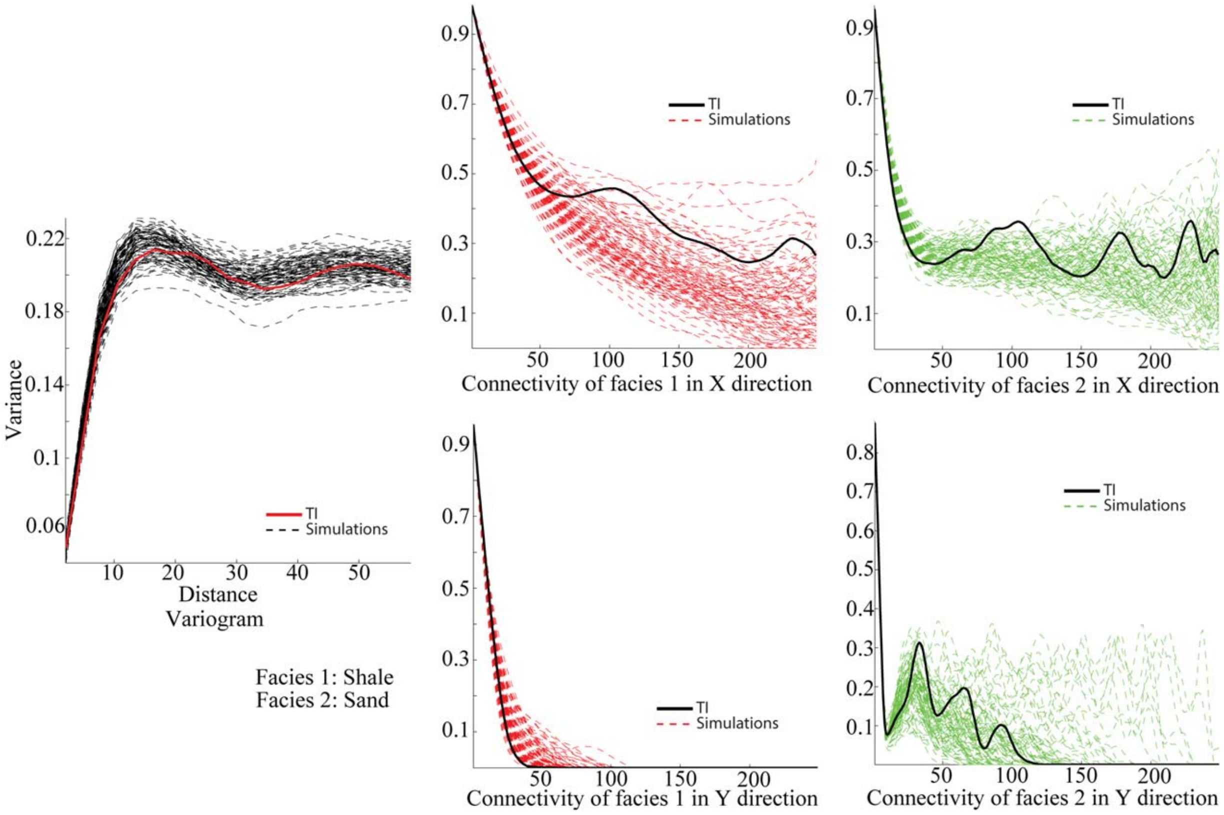
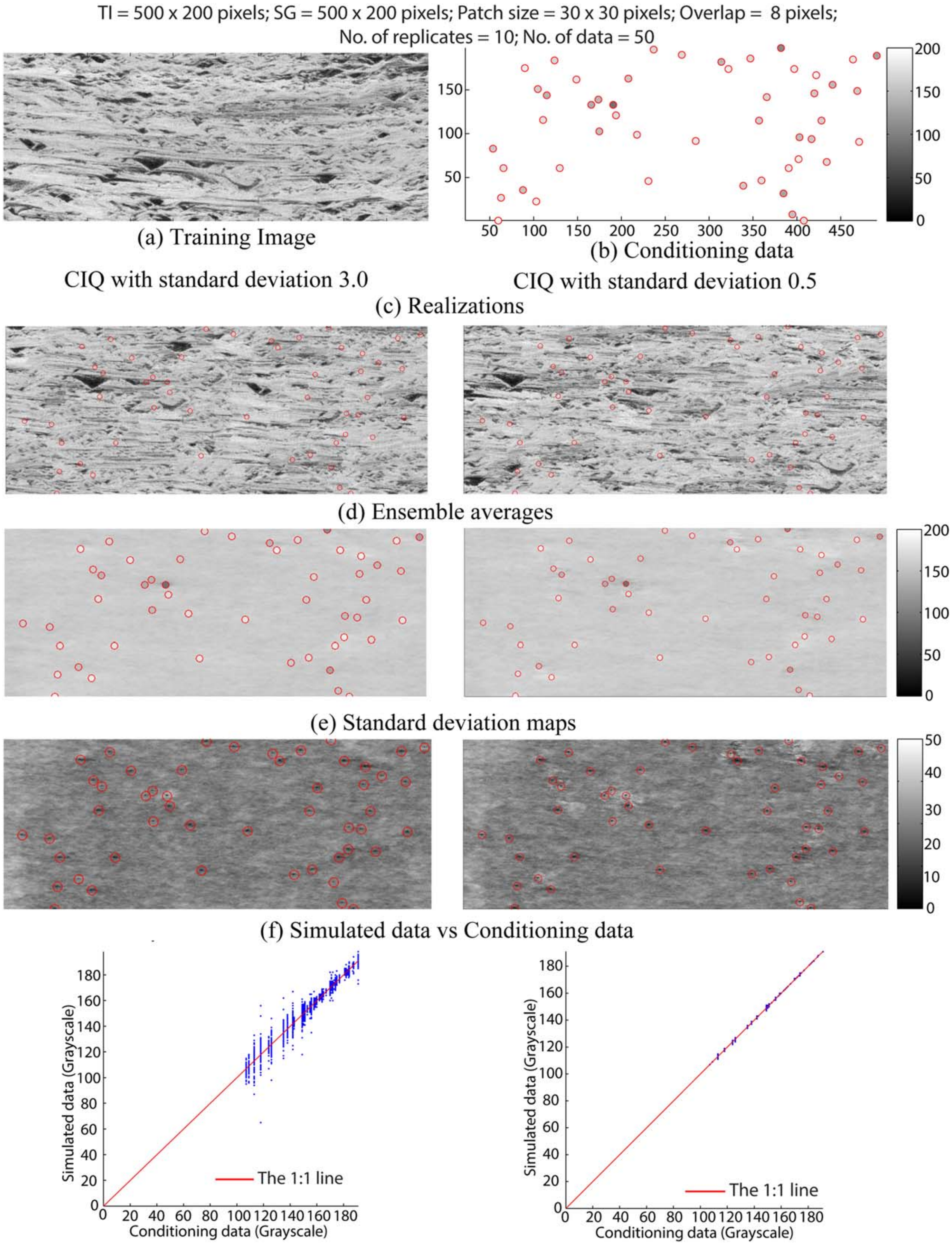


Figure 14. Performance of CIQ for the binary TI with 50 conditioning data in terms of variograms and connectivity functions.



(a) Training Image

(b) Conditioning data

(c) Realizations

(d) Ensemble averages

(e) Standard deviation maps

(f) Simulated data vs Conditioning data

Figure 15. Performance of CIQ algorithm for a continuous TI with the same 50 conditioning data and different measurement errors. The TI is a cross-section photograph of a flume experiment. Seen on this photograph are scour features (black color). (left column) Standard deviation of measurement errors = 3.0. (right column) Standard deviation of measurement errors = 0.5. The same color bar is used for conditioning data, TI, and realizations.

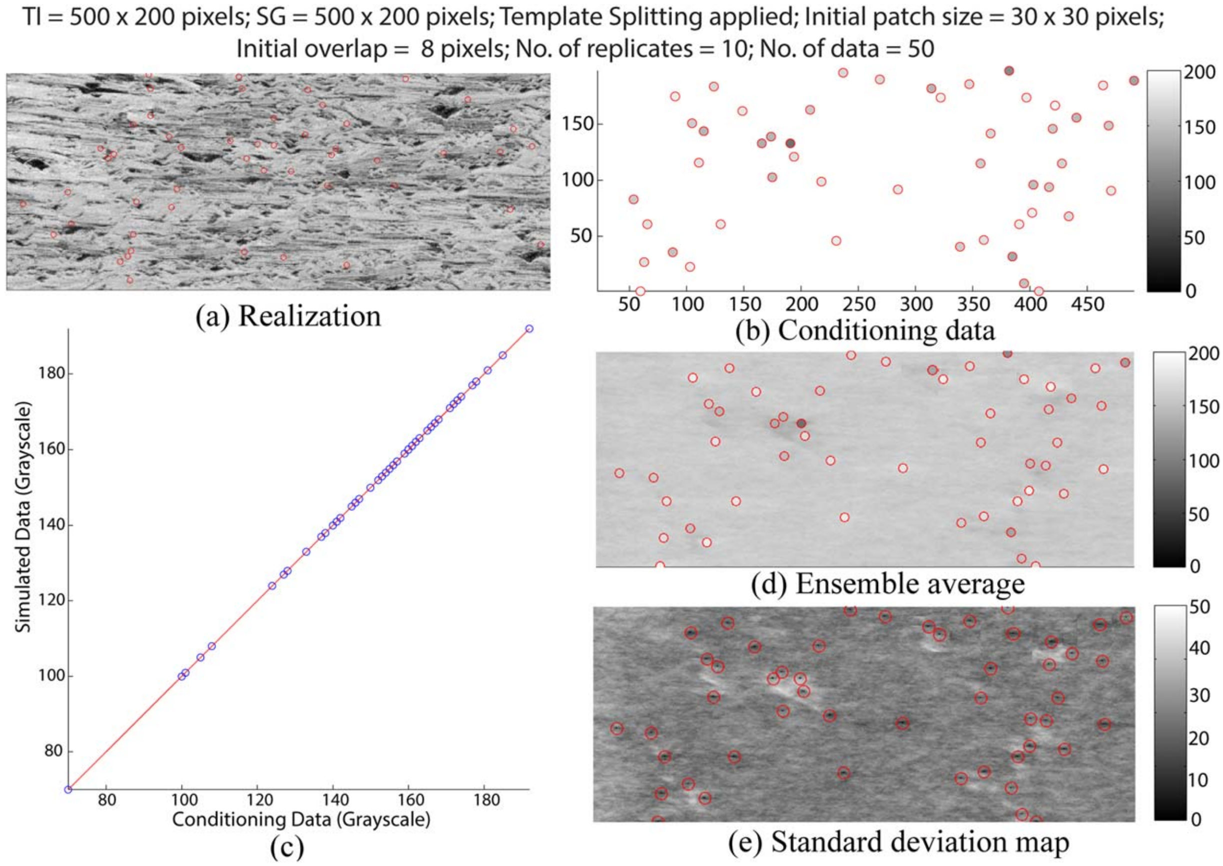


Figure 16. Performance of CIQ with template splitting for a continuous TI with 50 conditioning data. The same color bar is used for conditioning data and realization.

the error terms, $E_c = 0$ for all patches that match the data, and $E_c =$ infinity (or a very high value) for all patches that do not match. From the plot of simulated data against conditioning data (Figure 16c), it is evident that all the points are conditioned exactly. A realization in Figure 16a shows that conditioning using the template splitting technique does not produce artifacts in the neighborhood of the conditioning data. The average simulation time per realization (based on 100 realizations) is 40 s, far less than the CIQ case with smaller patch size. The ensemble average and the standard deviation map (Figures 16d and 16e), based on 100 realizations, show a variance reduction in the vicinity of few challenging data points.

5.5. 3-D Test Cases

Unconditional 3-D simulations results for three different TIs are shown in Figure 17. TI (a) consists of 3-D meandering channels. TI (b) was generated with the FFT-MA method [*Le Ravalec-Dupin et al.*, 2000], using a Gaussian variogram model that results in very smooth structures. TI (c) is a portion of 3-D flume model of *Paola et al.* [2001] (Only a 2-D section of this image was used in section 5.3.) The realizations have the same dimensions of TIs. In all cases we used a patch size of $p = 20 \times 20 \times 20$ voxels with an overlap region of $o = 5$ pixels. In all cases, the realizations reasonably are satisfactory:

1. Reproduction of the connected 3-D channels in TI (a);

2. Reproduction of the smooth spatial continuity of TI (b);

3. Reproduction of the complex sedimentary features, in particular the scour structures of TI (c).

The realizations are visually similar to the training images. In order to have a quantitative assessment of the performance of 3-D IQ, we also compare in Figure 17 the omnidirectional variograms of the training image and 10 realizations for all three cases considered (respectively, the indicator variograms for the categorical case). They indicate that IQ is able to reproduce the essential features of the 3-D TIs considered.

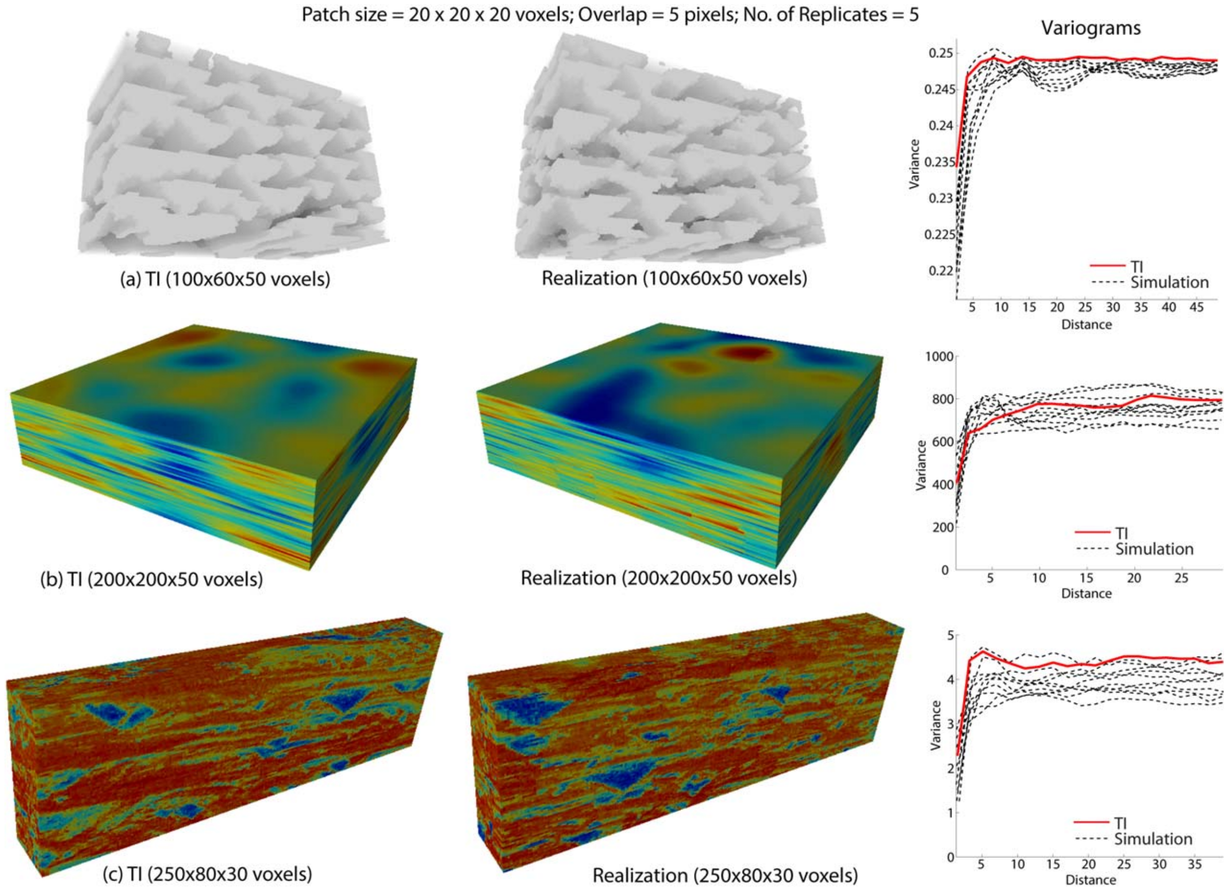


Figure 17. Performance of IQ algorithm for 3-D cases.

6. Conclusion

Recently, multiple-point geostatistics has gained a lot of attention in the fields of hydrogeology, remote sensing, mining, and petroleum engineering. However, in some of these applications, their computational cost presents a limiting factor limiting their application. Despite this, using training images is an alternative way of conveying complex spatial continuity that is difficult to formulate mathematically. For example, one of the prime validation criteria used to assess geostatistical simulations is still visual examination, because perception by the human eye is an excellent proxy for spatial statistical analysis.

Following this logic, we explored texture generation methods developed in computer graphics, where the human eye is the sole validation criterion. In this paper, we present image quilting, a highly efficient method inspired from computer graphics algorithms. We adapted the method for conditional simulation (conditional image quilting, CIQ), and introduced a conditional strategy that includes, for the first time, the possibility to condition to data in the framework of MPS.

We found that CIQ is as good as state-of-the art multiple-point algorithms, offering similar capabilities in terms of statistics reproduction and dealing with different types of variables (categorical/continuous). A major advantage is the improved computational performance of IQ compared to established methods, with a reduction in computational cost of a factor at least 50 for large 3-D cases.

At the same time, an important feature of CIQ is that it is amenable to parallel processing [*Bergen and Adelson*, 1988; *Peredo and Ortiz*, 2011; *Tahmasebi et al.*, 2012b], especially for the convolution step that can be carried on cluster computers of GPUs. Finally, the efficiency could be further improved by applying other acceleration strategies used in computer graphics, such as a multiscale search, or using the graph cuts approach [*Kwatra et al.*, 2003], offering increased flexibility to integrate conditioning data. These issues will be the topic of future research.

We believe that the major contribution of this paper is to offer the possibility to perform stochastic modeling of complex structures on a simple workstation, even for very large cases involving many realizations. This is valid for a range of modeling applications including representation of geological structures, Earth surface features, and spatiotemporal applications. Removing the need for large parallel cluster or even graphics card (GPUs) which are difficult to program, can make MPS much easier to use and to apply to new fields of earth sciences.

Acknowledgments
This work was supported by a scholarship from the University of New South Wales and the research projects undertaken as part of the Australian Research Council and National Water Commission funding for the National Centre for Groundwater Research and Training (NCGRT). We are thankful for very insightful comments by the Associate Editor and two anonymous reviewers, which were very helpful in improving the paper.

References

Alcolea, A., and P. Renard (2010), The blocking moving window sampler. Conditioning multiple-point simulations to hydrogeological data, *Water Resour. Res.*, *46*, W08511, doi:10.1029/2009WR007943.

Arpat, B., and J. Caers (2007), Conditional simulations with patterns, *Math. Geol.*, *39*(2), 177–203.

Bergen, J., and E. Adelson (1988), Early vision and texture perception, *Nature*, *333*(6171), 363–364.

Boucher, A., C. Kyriakidis, and C. Cronkite-Ratcliff (2008), Geostatistical solutions for super-resolution land cover mapping, *IEEE Trans. Geosci. Remote Sens.*, *46*(1), 272–283.

Caers, J., and T. Zhang (2004), Multiple-point geostatistics: A quantitative vehicle for integrating geologic analogs into multiple reservoir models, in *Integration of Outcrop and Modern Analog Data in Reservoir Models, AAPG Memoir 80*, edited by G. M. Grammer, P. M. Harris, and G. P. Eberli, pp. 383–394, Am. Assoc. of Pet. Geol., Tulsa, CA

Caers, J., S. Strebelle, and K. Payrazyan (2003), Stochastic integration of seismic data and geologic scenarios: A West Africa submarine channel saga, *Leading Edge*, *22*(3), 192–196.

Comunian, A., P. Renard, J. Straubhaar, and P. Bayer (2011), Three-dimensional high resolution fluvio-glacial aquifer analog—Part 2: Geostatistical modeling, *J. Hydrol.*, *405*(1–2), 10–23.

de Marsily, G., F. Delay, J. Gonçalvès, P. Renard, V. Teles, and S. Violette (2005), Dealing with spatial heterogeneity, *Hydrogeol. J.*, *13*(1), 161–183.

dell'Arciprete, D., R. Bersezio, F. Felletti, M. Giudici, A. Comunian, and P. Renard (2012), Comparison of three geostatistical methods for hydrofacies simulation: A test on alluvial sediments, *Hydrogeol. J.*, *20*, 299–311.

Dijkstra, E. (1959), A note on two problems in connexion with graphs, *Numer. Math.*, *1*, 269–271, doi:10.1007/BF01386390.

Efros, A., and T. Leung (1999), Texture synthesis by non-parametric sampling, in *Proceedings of Seventh IEEE International Conference on Computer Vision*, pp. 1033–1038, Kerkyra, Corfu, Greece.

Efros, A. A., and W. T. Freeman (2001), Image quilting for texture synthesis and transfer, paper presented at the ACM SIGGRAPH Conference on Computer Graphics, Los Angeles.

El Ouassini, A., A. Saucier, D. Marcotte, and B. D. Favis (2008), A patchwork approach to stochastic simulation: A route towards the analysis of morphology in multiphase systems, *Chaos Solit. Fract.*, *36*(2), 418–436.

Falivene, O., P. Arbués, A. Gardiner, G. Pickup, J. Munõz, and L. Cabrera (2006), Best practice stochastic facies modeling from a channel-fill turbidite sandstone analog (the Quarry outcrop, Eocene Ainsa basin, northeast Spain), *AAPG Bull.*, *90*(7), 1003–1029.

Faucher, C., A. Saucier, and D. Marcotte (2012), A new patchwork simulation method with control of the local-mean histogram, *Stochastic Environ. Res. Risk Assess.*, *27*(1), 253–273.

Garcıa-Morales, M. B. (2003), *Non Stationnarite Dans les Modeles de Type Booleen: Application a la Simulation d'unites Sedimentaires*, Ecole des Mines de Paris, Paris.

Gómez-Hernández, J. J., and X. H. Wen (1998), To be or not to be multi-Gaussian? A reflection on stochastic hydrogeology, *Adv. Water Resour.*, *21*(1), 47–61.

Goovaerts, P. (1997), *Geostatistics for Natural Resources Evaluation*, 496 pp., Oxford Univ. Press, Oxford, U. K.

Guardiano, F., and M. Srivastava (1993), Multivariate geostatistics: Beyond bivariate moments, in *Geostatistics-Troia*, edited by A. Soares, pp. 133–144, Kluwer Acad., Dordrecht, Netherlands.

Hansen, T. M., K. S. Cordua, and K. Mosegaard (2012), Inverse problems with non-trivial priors: Efficient solution through sequential Gibbs sampling, *Comput. Geosci.*, *16*(3), 593–611.

He, X., T. O. Sonnenborg, F. Jørgensen, and J. K. H. (2013), The effect of training image and secondary data integration with multiple-point geostatistics in groundwater modeling, *Hydrol. Earth Syst. Sci. Discuss.*, *10*, 11,829–11,860, doi:10.5194/hessd-10-11829-2013.

Honarkhah, M. (2011), Stochastic simulation of patterns using distance-based pattern modeling, PhD thesis, Stanford Univ., Stanford, Calif.

Honarkhah, M., and J. Caers (2010), Stochastic simulation of patterns using distance-based pattern modeling, *Math. Geosci.*, *42*(5), 487–517.

Hu, L., and T. Chugunova (2008), Multiple-point geostatistics for modeling subsurface heterogeneity: A comprehensive review, *Water Resour. Res.*, *44*, W11413, doi:10.1029/2008WR006993.

Huysmans, M., and A. Dassargues (2009), Application of multiple-point geostatistics on modelling groundwater flow and transport in a cross-bedded aquifer (Belgium), *Hydrogeol. J.*, *17*(8), 1901–1911.

Huysmans, M., and A. Dassargues (2011), Direct multiple-point geostatistical simulation of edge properties for modeling thin irregularly shaped surfaces, *Math. Geosci.*, *43*(5), 521–536, doi:10.1007/s11004-011-9336-7.

Jha, S. K., G. Mariethoz, J. P. Evans, and M. F. McCabe (2013a), Demonstration of a geostatistical approach to physically consistent downscaling of climate modeling simulations, *Water Resour. Res.*, *49*, 245–259, doi:10.1029/2012WR12602.

Jha, S. K., G. Mariethoz, and B. F. J. Kelly (2013b), Bathymetry fusion using multiple-point geostatistics: Novelty and challenges in representing non-stationary bedforms, *Environ. Modell. Software*, *50*, 66–76.

Journel, A., and T. Zhang (2006), The necessity of a multiple-point prior model, *Math. Geol.*, *38*(5), 591–610.

Khodabakhshi, M., and B. Jafarpour (2013), A Bayesian mixture-modeling approach for flow-conditioned multiple-point statistical facies simulation from uncertain training images, *Water Resour. Res.*, *49*, 328–342, doi:10.1029/2011WR010787.

Kwatra, N., A. Schödl, I. Essa, G. Turk, and A. Bobick (2003), Graphcut textures: Image and video synthesis using graph cuts, *ACM Trans. Graph.*, *22*(3), 277–286.

Le Coz, M., P. Genthon, and P. M. Adler (2011), Multiple-point statistics for modeling facies heterogeneities in a porous medium: The Komadugu-Yobe Alluvium, Lake Chad Basin, *Math. Geosci.*, *43*(7), 861–878.

Le Ravalec-Dupin, M., B. Noetinger, and L. Y. Hu (2000), The FFT moving average (FFT-MA) generator: An efficient numerical method for generating and conditioning Gaussian simulations, *Math. Geol.*, *32*(6), 701–723.

Liu, Y. (2006), Using the Snesim program for multiple-point statistical simulation, *Comput. Geosci.*, *23*(2006), 1544–1563.

Mariethoz, G., and S. Lefebvre (2014), Bridges between multiple-point geostatistics and texture synthesis, *Math. Geosci.*, *66*, 66–80.

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

Mariethoz, G., M. McCabe, and P. Renard (2012), Multivariate spatio-temporal reconstruction of gaps for spatially continuous satellite based retrievals, *Water Resour. Res.*, *48*, W10507, doi:10.1029/2012WR012115.

Meerschman, E., G. Pirot, G. Mariethoz, J. Straubhaar, M. Van Merivenne, and P. Renard (2013), A practical guide to performing multiple-point geostatistical simulations with the direct sampling algorithm, *Comput. Geosci.*, *52*, 307–324, doi:10.1016/j.cageo.2012.09.019.

Michael, H., A. Boucher, T. Sun, J. Caers, and S. Gorelick (2010), Combining geologic-process models and geostatistics for conditional simulation of 3-D subsurface heterogeneity, *Water Resour. Res.*, *46*, W05527, doi:10.1029/2009WR008414.

Mustapha, H., and R. Dimitrakopoulos (2010), High-order stochastic simulation of complex Spatially distributed natural phenomena, *Math. Geosci.*, *42*(5), 457–485.

Okabe, H., and M. Blunt (2007), Pore space reconstruction of vuggy carbonates using microtomography and multiple-point statistics, *Water Resour. Res.*, *43*, W12S02, doi:10.1029/2006WR005680.

Paola, C., et al. (2001), Strong experimental stratigraphy, *GSA Today*, *11*(7), 4–9.

Paola, C., K. Straub, D. Mohrig, and L. Reinhardt (2009), The "unreasonable effectiveness" of stratigraphic and geomorphic experiments, *Earth Sci. Rev.*, *97*(1–4), 1–43.

Pardo-Igúzquiza, E., and P. Dowd (2003), CONNEC3D: A computer program for connectivity analysis of 3D random set models, *Comput. Geosci.*, *29*, 775–785.

Park, H., C. Scheidt, D. Fenwick, A. Boucher, and J. Caers (2013), History matching and uncertainty quantification of facies models with multiple geological interpretations, *Comput. Geosci.*, *17*(4), 609–621. DOI: 10.1007/s10596-013-9343-5.

Parra, A., and J. M. Ortiz (2011), Adapting a texture synthesis algorithm for conditional multiple point geostatistical simulation, *Stochastic Environ. Res. Risk Assess.*, *25*(8), 1101–1111.

PCAST (2010), *Designing a Digital Future: Federally Funded Research and Development in Networking and Information Technology*, 71 pp., Office of Science and Technology Policy, Washington, D. C.

Peredo, O., and J. M. Ortiz (2011), Parallel implementation of simulated annealing to reproduce multiple-point statistics, *Comput. Geosci.*, *37*(8), 1110–1121.

Pham, T. D. (2012), Supervised restoration of degraded medical images using multiple-point geostatistics, *Comput. Methods Programs Biomedicine*, *106*(3), 201–209.

Renard, P., and D. Allard (2013), Connectivity metrics for subsurface flow and transport, *Adv. Water Resour.*, *51*, 168–196.

Renard, P., J. Gómez-Hernández, and S. Ezzedine (2006), Characterization of porous and fractured media, in *Encyclopedia of Hydrological Sciences*, edited by M. G. Anderson and J. J. McDonnell, John Wiley, Chichester, U. K. pp. 1–18. DOI: 10.1002/0470848944.hsa154.

Renard, P., J. Straubhaar, J. Caers, and G. Mariethoz (2011), Conditioning Facies simulations with connectivity data, *Math. Geosci.*, *43*(8), 879–903, doi:10.1007/s11004-011-9363-4.

Ronayne, M. J., S. M. Gorelick, and J. Caers (2008), Identifying discrete geologic structures that produce anomalous hydraulic response: An inverse modeling approach, *Water Resour. Res.*, *44*, W08426, doi:10.1029/2007WR006635.

Ronayne, M. J., S. M. Gorelick, and C. Zheng (2010), Geological modeling of submeter scale heterogeneity and its influence on tracer transport in a fluvial aquifer, *Water Resour. Res.*, *46*, W10519, doi:10.1029/2010WR009348.

Sánchez-Vila, X., J. Carrera, and J. P. Girardi (1996), Scale effects in transmissivity, *J. Hydrol.*, *183*(1–2), 1–22.

Schlüter, S., and H. J. Vogel (2011), On the reconstruction of structural and functional properties in random heterogeneous media, *Adv. Water Resour.*, *34*(2), 314–325.

Shannon, C. (1948), A mathematical theory of communication, *Bell Syst. Tech. J.*, *27*, 379–423.

Strebelle, S. (2002), Conditional simulation of complex geological structures using multiple-point statistics, *Math. Geol.*, *34*(1), 1–22.

Tahmasebi, P., and M. Sahimi (2013), Cross-correlation function for accurate reconstruction of heterogeneous media, *Phys. Rev. Lett.*, *110*, 078002.

Tahmasebi, P., A. Hezarkhani, and M. Sahimi (2012a), Multiple-point geostatistical modeling based on the cross-correlation functions, *Comput. Geosci.*, *16*, 779–797, doi:10.1007/s10596-012-9287-1.

Tahmasebi, P., M. Sahimi, G. Mariethoz, and A. Hezarkhani (2012b), Accelerating geostatistical simulations using graphics processing units (GPU), *Comput. Geosci.*, *46*, 51–59.

Western, A., G. Blöschl, and R. Grayson (2001), Toward capturing hydrologically significant connectivity in spatial patterns, *Water Resour. Res.*, *37*, 83–97.

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

Zhou, H., J. Gomez-Hernandez, and L. Li (2012), A pattern search based inverse method, *Water Resour. Res.*, *48*, W03505, doi:10.1029/2011WR011195.