



A patchwork approach to stochastic simulation: A route towards the analysis of morphology in multiphase systems

Ayoub El Ouassini ^a, Antoine Saucier ^{b,*}, Denis Marcotte ^{c,1}, Basil D. Favis ^{d,2}

^a École Polytechnique de Montréal, C.P. 6079, Station centre-ville, Montréal, Que., Canada H3C-3A7

^b École Polytechnique de Montréal, département de mathématiques et de génie industriel, C.P. 6079, Station centre-ville, Montréal, Que., Canada H3C-3A7

^c École Polytechnique de Montréal, département de génie civil, géologique et minier, C.P. 6079, Station centre-ville, Montréal, Que., Canada H3C-3A7

^d École Polytechnique de Montréal, département de génie chimique, C.P. 6079, Station centre-ville, Montréal, Que., Canada H3C-3A7

Accepted 16 June 2006

Communicated by Professor M.S. Elnaschie

Abstract

We propose a new sequential stochastic simulation approach for black and white images in which we focus on the accurate reproduction of the small scale geometry. Our approach aims at reproducing correctly the connectivity properties and the geometry of clusters which are small with respect to a given length scale called block size. Our method is based on the analysis of statistical relationships between adjacent square pieces of image called blocks. We estimate the transition probabilities between adjacent blocks of pixels in a training image. The simulations are constructed by juxtaposing one by one square blocks of pixels, hence the term patchwork simulations. We compare the performance of patchwork simulations with Strebelle's multipoint simulation algorithm on several types of images of increasing complexity. For images composed of clusters which are small with respect to the block size (e.g. squares, discs and sticks), our patchwork approach produces better results than Strebelle's method. The most noticeable improvement is that the cluster geometry is usually reproduced accurately. The accuracy of the patchwork approach is limited primarily by the block size. Clusters which are significantly larger than the block size are usually not reproduced accurately. As an example, we applied this approach to the analysis of a co-continuous polymer blend morphology as derived from an electron microscope micrograph.

© 2006 Elsevier Ltd. All rights reserved.

* Corresponding author. Tel.: +1 514 340 4711x4516; fax: +1 514 340 4463.

E-mail addresses: ayoub.el-ouassini@polymtl.ca (A. El Ouassini), antoine.saucier@polymtl.ca (A. Saucier), denis.marcotte@polymtl.ca (D. Marcotte), basil.favis@polymtl.ca (B.D. Favis).

¹ Tel.: +1 514 340 4711x4620; fax: +1 514 340 4191.

² Tel.: +1 514 340 4711x4527; fax: +1 514 340 4159.

1. Introduction

Stochastic simulations are used to model a wide variety of random systems and processes such as oil reservoirs [16], groundwater flow [5], porous media [1,2] and composite materials [30]. In this paper, we propose a new stochastic simulation approach in which we focus on the accurate reproduction of the small scale geometry. More precisely, our approach aims at modelling correctly the connectivity properties and the geometry of clusters which are small with respect to a given fixed length scale.

Our focus is motivated by the fact that morphology and connectivity often play a central role in the determination of physical properties which are of interest in various contexts. Here are a few examples: the absolute permeability of a porous medium is directly related to the degree of pore connectivity; the electrical conductivity of a composite material depends on the degree of connectivity of the conductive regions within the material; the large scale fluid flow properties of an oil reservoir can be determined by the degree of connectivity of large scale channels; finally, the physical properties (e.g. permeability to gas or liquids, electrical conductivity) of a co-continuous polymer blend [17] are essentially determined by the morphology of the two intertwined phases composing the blend.

In the context of porous media, some of the approaches which have been used to describe the morphology are network models [6,12], grain models [10], percolation models [9,22], diagenetic models [4,20] and fractal models [15,21,26,25]. These models rely either on specific assumptions about the pore geometry or on the existence of a scale invariance property. They do not provide a data-adaptive framework in which the simulation can be performed by extracting information from arbitrary training images.

Sequential simulation is one of the most widely used techniques in stochastic simulations. Sequential simulation techniques have long been based on two-point statistics. Efficient algorithms have been developed such as the sequential Gaussian simulation (SGSIM), or the sequential indicator simulations (SISIM) [11]. However, these methods cannot reproduce accurately complex and connected structures such as sinuous channels. Indeed SGSIM and SISIM reproduce only the histogram and the variogram, which is insufficient to characterize complex geometries.

The accurate reproduction of complex random geometries may require more than two-point statistics. In the context of a multipoint statistical characterization based on multiscale moments [27], Saucier and Muller showed that an accurate reproduction of multiscale moments does not necessarily lead to an accurate reproduction of cluster shapes and connectivity. The difficulty is that subtle morphological features are not necessarily captured by a given statistical characterization, e.g. multiscale moments.

Complex images can be simulated with the object-based approach, which is based on the parametrization of shapes corresponding to desired geometries, e.g. sinuous channels. These parameters are randomized to generate random geometries within a specified volume [10,14,18]. The main limitation of this approach is the difficulty to parametrize complex geometries. Moreover, the conditioning of the random parameters to local data can be difficult to achieve.

Guardiano and Srivastava [13] proposed a new algorithm based on multiple-point statistics. The method proposed is a sequential indicator simulation exploiting conditional probabilities extracted directly from a training image. The advantage of this approach compared with the object-based approach is the possibility to introduce conditional data in a simulation. Moreover, the approach is not iterative unlike other multiple-point approaches such as the Markov Chain Monte Carlo simulation of Caers and Journel [7,8]. However, the original code was computationally demanding and was not used in practice.

Strebelle [28,29] recently proposed a practical algorithm based on Guardiano and Srivastava's idea. This algorithm allowed the practical use of multiple-point statistics in simulations. Strebelle uses only the data observed in the training image to simulate possible realizations. In the course of a simulation, if one encounters a configuration of points which has not been observed in the training image, then the number of points is reduced iteratively until an observed configuration of points is found. Another approach proposed by Arpat [3] deals with unobserved configurations of points by modeling and interpolating between existing configurations in the training image.

In this paper, we present a new sequential simulation method which is based on the analysis of the statistical relationships between adjacent square pieces of image called blocks. More precisely, we estimate the conditional probability to observe a block knowing a set of adjacent blocks. These conditional probabilities are regarded as transition probabilities between adjacent sets of blocks. The statistical information contained in the training image is stored in the transition probabilities.

Our simulations are performed in two steps. Firstly, we scan the reference image to estimate the transition probabilities. Secondly, we use these probabilities to simulate the image block by block, linking each new square block to the previously simulated blocks. The linking procedure is designed to reproduce small scale structures and local connectivity properties as accurately as possible. This block based process resembles the fabrication of a quilt, hence the term patchwork simulation. This approach can reproduce correctly many morphological features because the blocks used in the patchwork process, which are drawn directly from the training image, are essentially correct.

This paper is structured as follows. In Section 2, we describe the patchwork simulation method. In Section 3, we apply our patchwork method to several types of training images and we compare the results with Strebelle's multipoint simulation algorithm [28,29]. We consider synthetic images composed of isolated clusters (squares, disks, sticks and channels) that mimic the morphology of two-phase systems for which one phase is predominant. Finally, we apply this approach to the specific case of a co-continuous polymer blend morphology.

2. Description of the patchwork simulation method

2.1. Preliminary definitions

Our method uses black and white images, i.e. each pixel of the reference image is coded on a single bit (1 or 0). The proportion of black pixels in an image will be called the image porosity. A square region in the image is called a block. Each block of size n pixels is composed of n^2 pixels, where $n > 0$ is an integer. A state S_i is a unique configuration of pixels observed in a block or in a set of several adjacent blocks in the image. A state S_i is represented by a sequence of bits such as

$$S_i = \underbrace{1010 \dots 100}_{b \times n^2} 10101,$$

where n is the block size and b is the number of adjacent blocks. In Fig. 1, the three adjacent blocks represented by circles correspond to a start state S_i coded on $3n^2$ bits, whereas the single block represented by a cross corresponds to an arrival state S_j coded on n^2 bits. We will say that the two states S_i and S_j are complementary because their association is a quadruplet of adjacent blocks which is observed in the reference image. Our method is based on the estimation of transition probabilities $P_{i,j}$ between such complementary states.

The transition probabilities $P_{i,j} \geq 0$ satisfy $\sum_j P_{i,j} = 1$ for each i . These probabilities are estimated by scanning a single or several reference images. For each image, the scanning can be done in a single direction or in the four directions of the image, i.e. directions obtained by rotating the reference image by 0° , 90° , 180° and 270° . If the reference image is statistically isotropic, then we scan it in the four directions to improve the precision of our estimates of the transition probabilities. If the reference image is statistically anisotropic, then the scanning is performed in a single direction.

In addition to the transition probabilities between adjacent blocks, we also estimate the probability of occurrence P_i of each observed start state. This information is used to correct unobserved states as we will see in Section 2.3.5.

2.2. Transition probability matrices

We use three types of transition probability matrices which are associated to different configurations of start and arrival states, as illustrated in Fig. 2. In the following, a specific configuration of start and arrival states will be called a *transition type*. The first transition probability matrix associated with the transition type represented in Fig. 2a is used to simulate the first line of blocks in the image. It contains transition probabilities between start and arrival states both

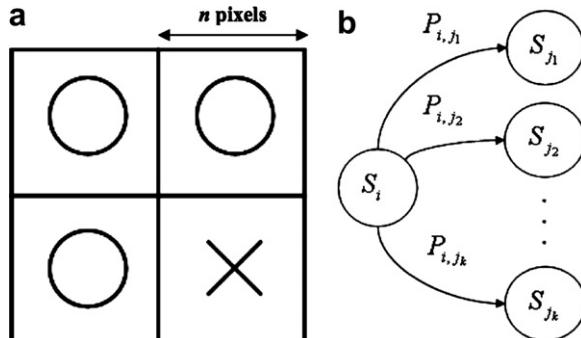


Fig. 1. Transition probabilities between adjacent blocks of size n pixels: (a) set of four adjacent blocks; (b) transition probabilities between a start state S_i and several possible arrival states $S_{j1}, S_{j2}, \dots, S_{jk}$. The start state S_i represented by the three circles in (a) is coded on $3n^2$ bits. The arrival state represented by a cross in (a) is coded on n^2 bits. The transition probability between the complementary states S_i and S_{jn} is denoted by $P_{i,jn}$, where $1 \leq n \leq k$.

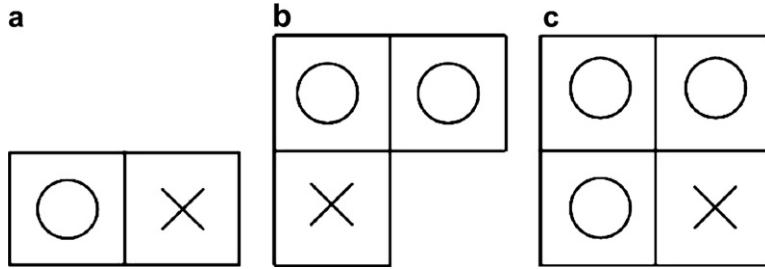


Fig. 2. The three configurations of start and arrival states used in the patchwork algorithm. Each square represents an image block of length n pixels. Start and arrival states are represented by circles and crosses respectively: (a) configuration used to simulate the blocks of the first line of the image; (b) configuration used to simulate the blocks of the first column of the image; (c) configuration used to simulate the rest of the blocks of the image.

coded on n^2 bits. The second transition matrix associated with the transition type represented in Fig. 2b is used to simulate the first column of blocks in the image. This matrix contains transition probabilities between a start state coded on $2n^2$ bits and an arrival state coded on n^2 bits. Finally, the third and main transition probability matrix associated with the third transition type (Fig. 2c) contains transition probabilities between a start state coded on $3n^2$ bits and an arrival state coded on n^2 bits. This matrix is used to simulate all the remaining blocks of the image.

2.3. Sequential simulation algorithm

The three transition types represented in Fig. 2 are used sequentially to construct an image, as illustrated in Fig. 3. The construction is performed block by block. The construction order is left to the right, top to bottom. In this section, we describe the initialization of a simulation and we explain how the three transition types are used in the construction.

2.3.1. Initialization

The first step in a simulation consists in selecting randomly a block from the training image. This block, which is positioned in the upper left corner of the image in Fig. 3a, is the start state of the simulation.

2.3.2. Simulating the first line of blocks with the first transition type

The next step consists in simulating a linear chain of blocks by using sequentially the first transition type (Fig. 2a). The corresponding transition probability matrix is used to simulate the next block (cross in Fig. 3b) located at the right side of the start state (black disc in Fig. 3b). Next, we shift to the right and the previous arrival state turns into a start state (black disc in Fig. 3c). We keep shifting to the right until all the blocks of the first line are simulated.

Note that the simulation of the first line of the image is a Markov chain. We emphasize that all pairs of adjacent blocks in the first line are observable configurations of pixels in the image, whereas a group of three or more adjacent blocks is not necessarily an observable configuration.

2.3.3. Simulating the blocks in the first column with the second transition type

Once all the blocks of the first line are simulated, we shift to the second line to simulate the first block of the second line (the cross in Fig. 3e). The second transition type (Fig. 2b) is used to simulate the first block of the second line knowing the two first blocks of the first line (the two black discs in Fig. 3e). More generally, we use the second transition type for each block located in the first column (Figs. 3e, i and m), except for the initialization block. Note that all the blocks of the second line must be simulated via the third transition type before the first block of the third line (Fig. 3i) can be simulated.

2.3.4. Simulating the inner blocks of the image with the third transition type

For all the remaining blocks, i.e. blocks which are neither located in the first line nor in the first column, we use the third transition type (Fig. 2c). This transition is used to simulate a single block knowing three adjacent blocks as illustrated in Figs. 3f, g, h, j, k, l, n, o and p.

2.3.5. The case of unobserved states

Starting from the second line and third column in the simulation (Fig. 3g), we may observe the occurrence of states that do not exist in the training image (the three black discs in Fig. 3g). In Fig. 3c, the use of the first transition probability

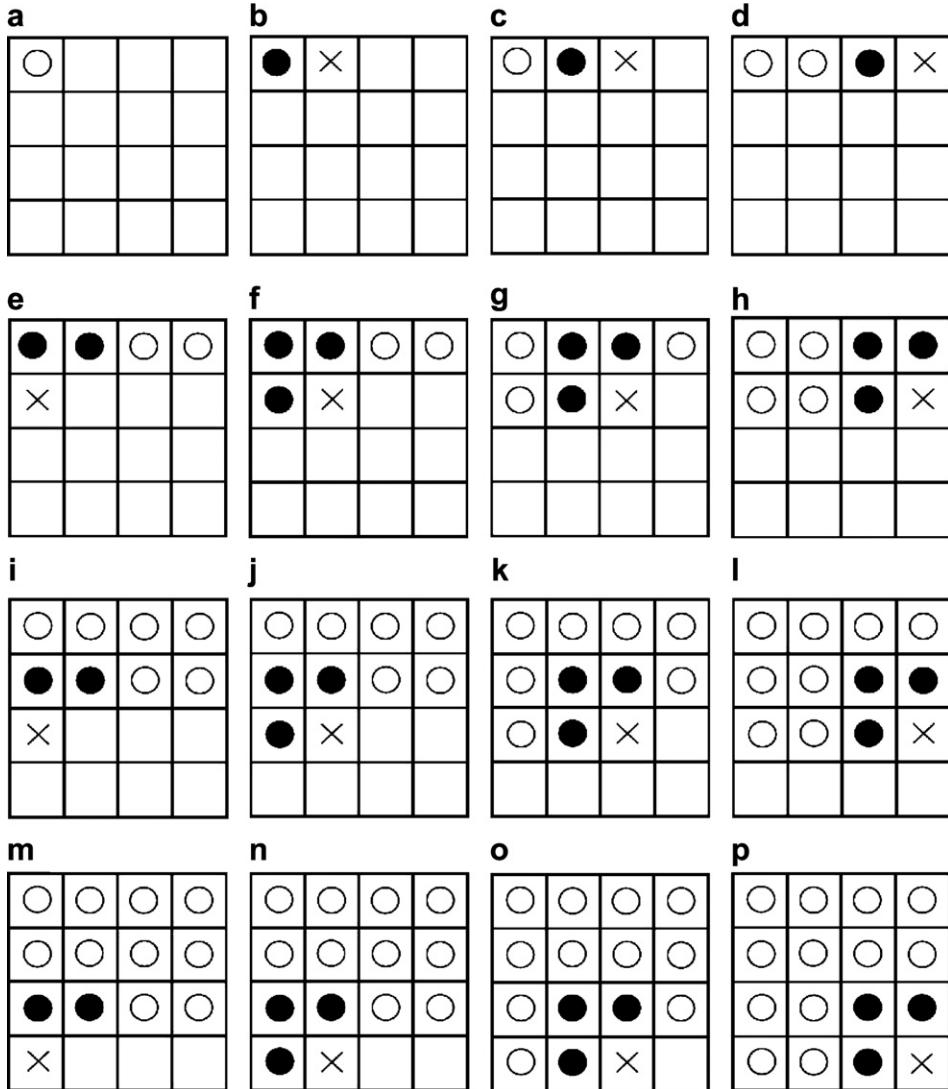


Fig. 3. Illustration of the use in a simulation of the three transition types. White discs represent blocks which have already been simulated. Black discs represent blocks currently used to simulate a complementary block represented by a cross. The initialization step (a) is followed in (b) by the use of the first transition matrix (Fig. 2a) to simulate the blocks of the first line in (b)–(d). The second transition matrix (Fig. 2b) is used to simulate the blocks of the first column in (e), (i) and (m). For the rest of the image, the third matrix (Fig. 2c) is used in (f), (g), (h), (j), (k), (l), (n), (o) and (p).

matrix ensures that the second pair of horizontal blocks in the top line is an observable configuration. In Fig. 3f, the use of the third transition probability matrix ensures that the second pair of vertical blocks is an observable configuration. However, the association of the two pairs of blocks, i.e. the three black discs in Fig. 3g, is not necessarily an observable configuration. This triplet of blocks can be either observed, unobserved or impossible to observe for this particular training image. This shows that our patchwork approach can generate new states, unlike Markov chains.

If an unobserved state S_U is encountered, then our strategy is to substitute it for an observed state S_O which is similar to the unobserved state. We quantify the degree of similarity between S_U and S_O in two complementary ways.

Firstly, the state S_O is chosen so that the porosities of S_O and S_U are comparable, i.e.

$$|\phi(S_O) - \phi(S_U)| \leq \epsilon \phi(S_U), \quad (1)$$

where $\phi(S)$ denotes the porosity of a state S and ϵ is a tolerance ratio ($\epsilon = 0.1$ usually gives good results). In the following, the application of the constraint (1) will be called porosity filtering. This filtering helps controlling the global

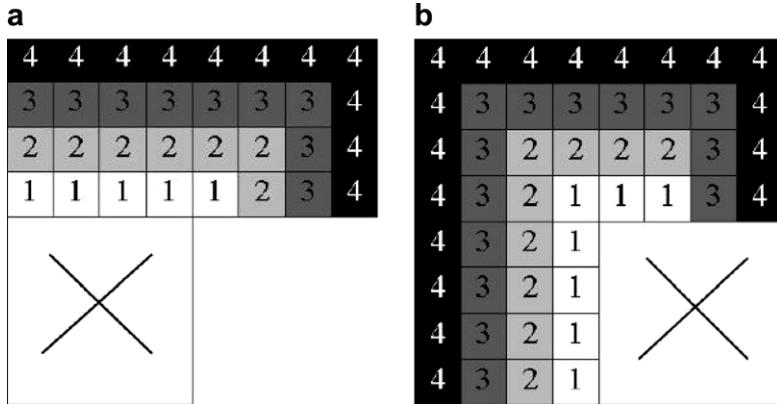


Fig. 4. Spatial weight distributions used for the metric. The weights range from 1 to 4: (a) weight distribution for the transition type displayed in Fig. 2b; (b) weight distribution for the transition type displayed in Fig. 2c. The weights decrease when the distance from the border with previously simulated blocks increases. This border is represented by a black strip.

porosity of the simulated image. In some cases, we will see that it can be advantageous to perform simulations without applying the porosity filtering.

Secondly, we consider the set of all the observed states which satisfy the porosity constraint (1) and we choose in this set the state S_O which is *closest* to S_U , i.e. the state S_O which minimizes the distance $d(S_O, S_U)$ between S_O and S_U . We define $d(S_O, S_U)$ by the metric

$$d(S_O, S_U) = \sum_{k=1}^M w_k |S_O(k) - S_U(k)|, \quad (2)$$

where $S_O(k)$ and $S_U(k)$ denote the k th bit of the states S_O and S_U respectively, M is the number of pixels on which the two states are coded, and w_k is the weight assigned to k th bit. If no porosity filtering is used, then the substitution state is simply the state S_O which minimizes $d(S_O, S_U)$. If several observed states are at the same distance from the unobserved state, then we select randomly one of them according to their occurrence probability.

We design the spatial weight distribution to reduce the connectivity discontinuities between the substitute state and the blocks simulated previously. To reach this goal, we give more weight to the bits which are located close to previously simulated blocks. More precisely, the weights w_k decrease linearly with the distance between the k th bit location and the border with previously simulated blocks, as shown in Fig. 4. The weights w_k satisfy $1 \leq w_k \leq P_{\max}$, where $P_{\max} \geq 1$ is the maximal weight. In Fig. 4, we illustrate the spatial weight distribution associated to the second and third transition types (Figs. 2b and c). A maximum weight of four is assigned to the bits located at the outer frontier whereas a minimum weight of one is assigned to the bits located at the inner frontier.

The use of the metric (2) with non-uniform weights for the correction of unobserved states has a significant impact on the simulations quality. The major effect of this metric is to reduce the breaking of the clusters which overlap between adjacent blocks. In that sense, using this metric helps respect the local connectivity properties. Fig. 5 shows the improvement obtained in simulations obtained with a training image composed of discs. The algorithm used to construct the image displayed in Fig. 5a is described in Section 3.2. We present in Fig. 5b a simulation in which we substitute each unobserved state by a randomly chosen observed state. This simulation is clearly of less quality than the simulations of Figs. 5c and d where our metric is used with a 10% porosity filtering, i.e. with $\epsilon = 0.1$. The simulation in Fig. 5c uses a uniform weight distribution, i.e. all the weights are equal to one, whereas the simulation in Fig. 5d uses a non-uniform weight distribution. We observe that the number of deformed discs is much smaller with the non-uniform weights metric than with the uniform weights metric (i.e. 27 versus 81, respectively, including isolated pixels). All the simulations presented in the following use the non-uniform weights metric with $P_{\max} = 4$.

3. Simulation results

In this section, we present simulation results obtained with our patchwork approach and with Strebelle's algorithm. The two approaches are tested and compared on several training images of increasing complexity.

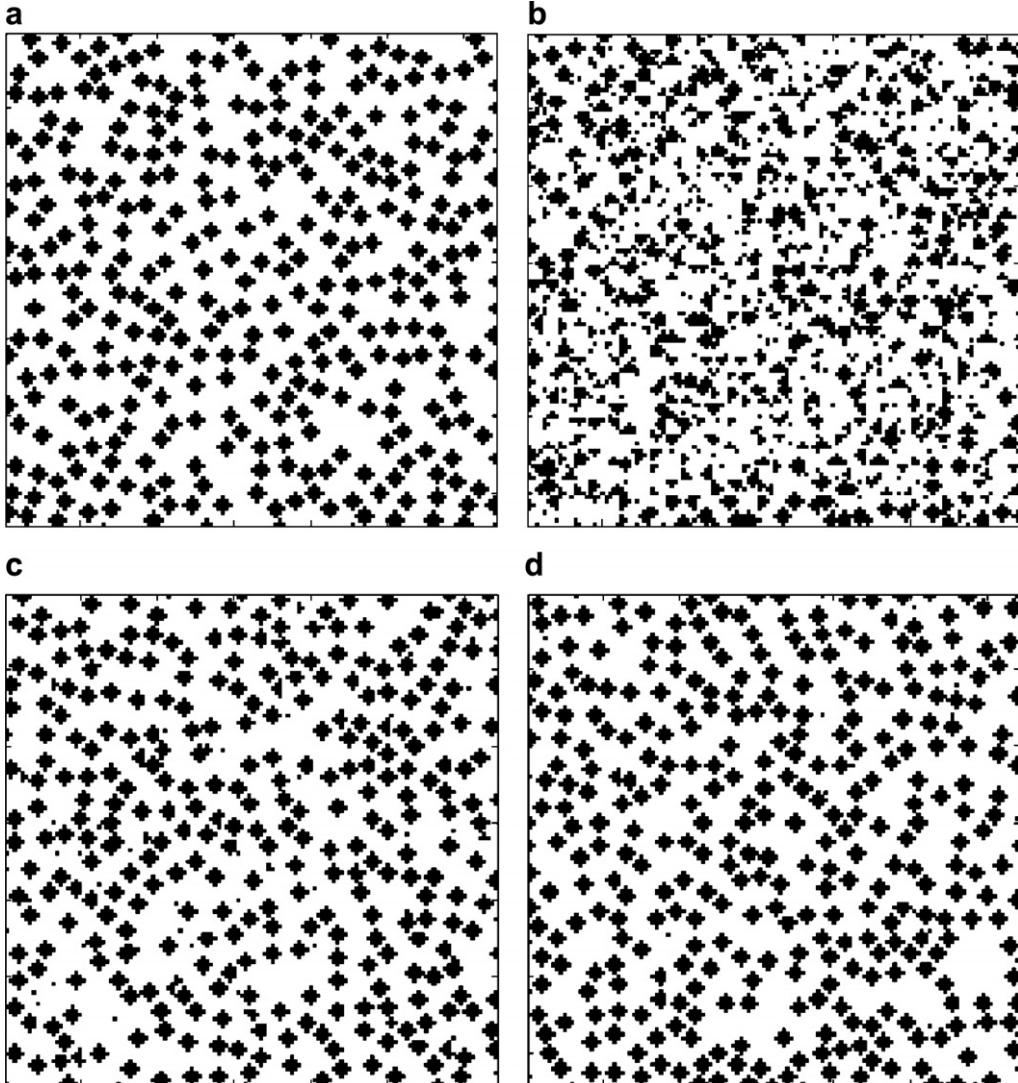


Fig. 5. Comparison between three simulations using different approaches for the correction of unobserved states, using the training image displayed in (a) which contains discs of diameter 4 pixels. In these simulations, we substitute each unobserved state by: (b) an observed state chosen randomly; (c) an observed state which is close to the unobserved state, using the metric with a uniform weight distribution; (d) an observed state which is close to the unobserved state, using the metric with a non-uniform weight distribution such that $P_{\max} = 4$. A 10% porosity filtering is used in (c) and (d).

3.1. Simulations using checkerboards as training images

For simple training images such as checkerboards, which are composed of blocks laid out regularly on a grid, we expect a priori our block based method to perform well because the number of possible states is relatively small. In Fig. 6, we show simulations obtained with a checkerboard used as a training image. The size of the images displayed is 32×32 pixels whereas the size of the training images used is 256×256 pixels. Checkerboards are composed of black and white square meshes. In our simulations, the mesh size varies from 1 pixel to 6 pixels. We used the simulation model based on a 4×4 pixels grid without porosity filtering. The results displayed in the second column of Fig. 6 show that the 4×4 model can reproduce perfectly checkerboards with a mesh size smaller than 5 pixels. We emphasize that our model has the ability to reproduce checkerboards with an offset effect. For checkerboards with a mesh size larger than 4 pixels (Figs. 6e and f), the 4×4 model does not reproduce checkerboards perfectly. Indeed, we observe in the second column of Figs. 6e and f the occurrence of several rectangles. Nevertheless, the black and white rectangles preserve the checkerboard visual appearance.

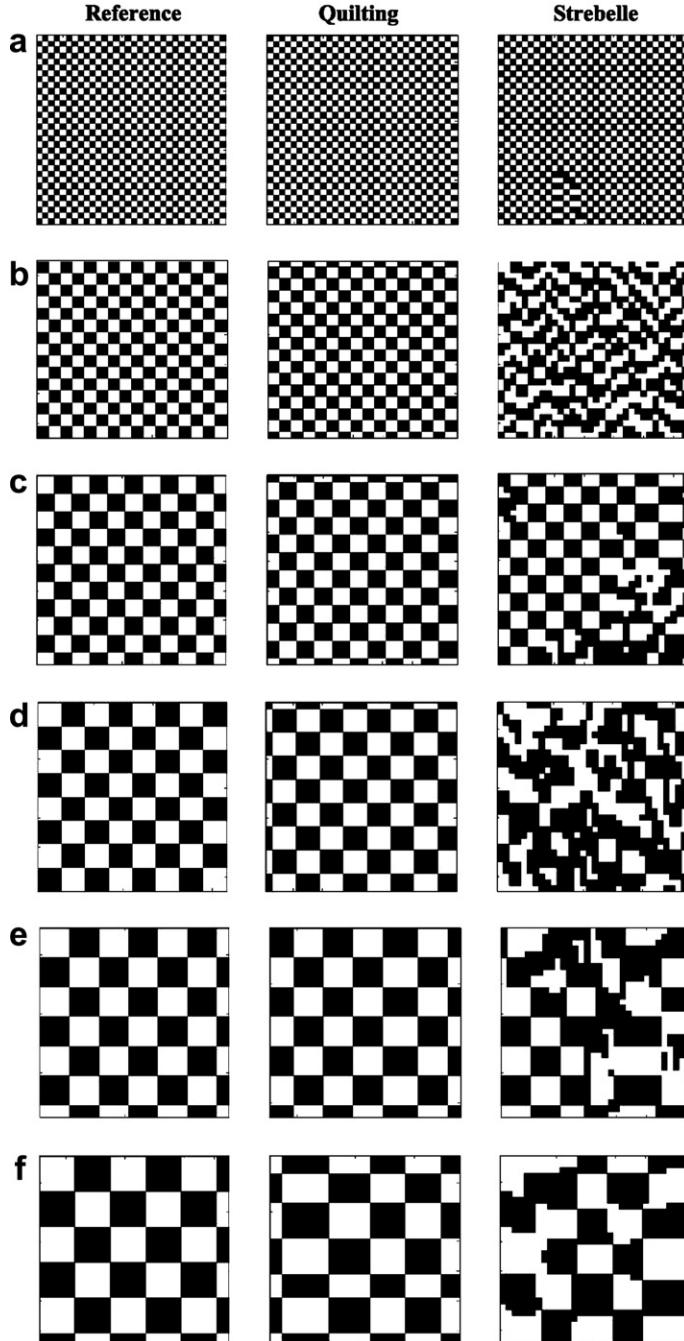


Fig. 6. Simulations using checkerboards with varying mesh size as training images. For each of the six cases (a)–(f), the image in the first column is the reference image, the image in the second column is the patchwork simulation using a model based on a 4×4 pixels grid and the third column contains the simulation using Strebelle's algorithm with a maximum of 60 points for conditioning and six multigrids. The dimensions of the black square (in unit of pixels) are (a) 1×1 ; (b) 2×2 ; (c) 3×3 ; (d) 4×4 ; (e) 5×5 ; (f) 6×6 .

We tested Strebelle's algorithm on the same checkerboard images. In this paper, all the simulations based on Strebelle's algorithm were produced with the software SGEMS [19]. The simulations presented in the third column of Fig. 6 were obtained by using a maximum of 60 conditioning points and six multigrids (these parameters were found to give the best results). The multi-grid concept consists in simulating first the coarsest grid (large scale), then a refined grid

conditional to the values already simulated on the coarse grid, and so on until the final desired grid is obtained. All the simulations involving Strebelle's algorithm use a porosity equal to the training image porosity. We can see that Strebelle's approach does not reproduce perfectly any of the checkerboards. Even checkerboards with a mesh size of 1 pixel (third column in Fig. 6a) are not reproduced perfectly. Hence our approach produces much better results than Strebelle's approach for this particular type of training images.

3.2. Simulations using training images composed of discs

3.2.1. Discs of constant diameters

We tested the two methods on images composed of discs having a fixed diameter. A disc is a simple object in the sense that it can be fully described by only three parameters, i.e. the radius and the two coordinates of the center. It

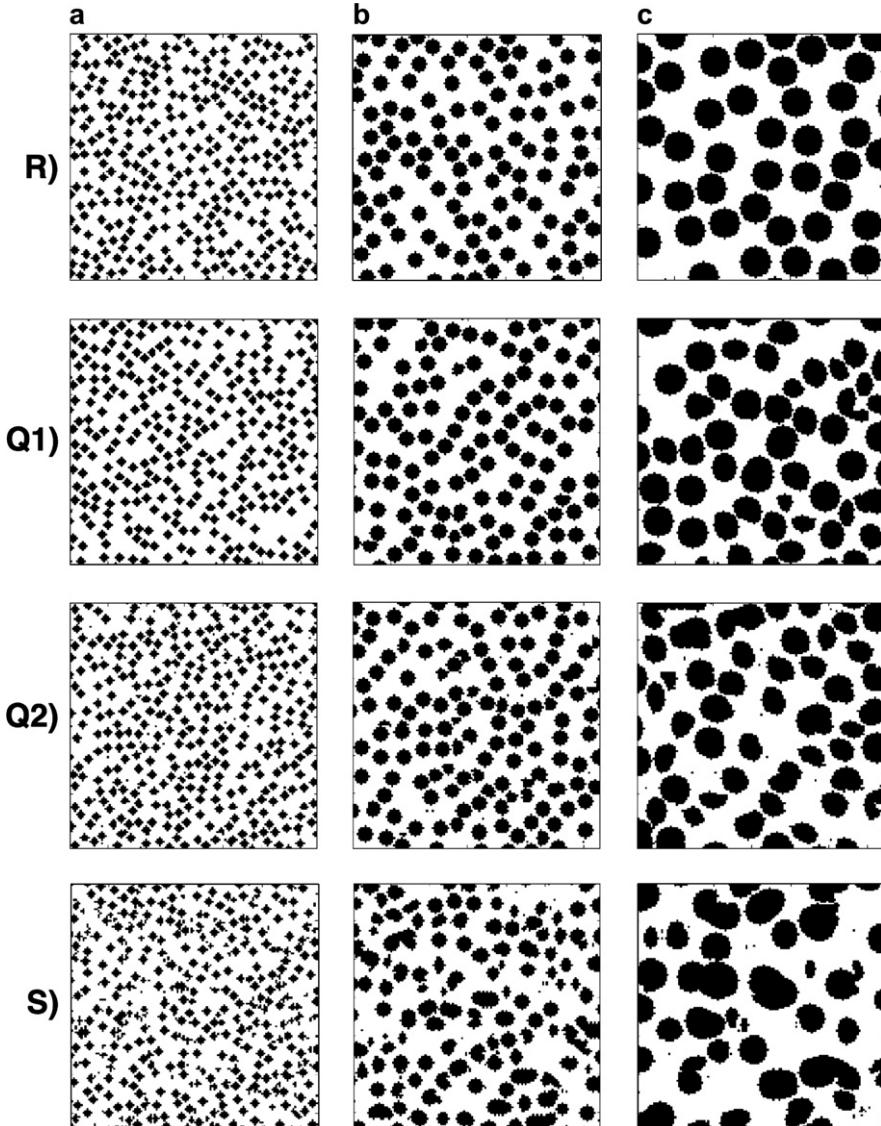


Fig. 7. Simulations based on training images composed of discs with a fixed diameter D : (a) $D = 4$ pixels; (b) $D = 8$ pixels; (c) $D = 16$ pixels. The patchwork simulations are done with a 4×4 pixels model. The first line (R) contains the reference images. Three types of simulations are presented: (Q1) patchwork simulations without porosity filtering; (Q2) patchwork simulations with a 10% porosity filtering; (S) simulations obtained with Strebelle's algorithm using a maximum of 60 points for conditioning and four multigrids. The size of the images displayed is 128×128 pixels, i.e. a quarter of the training image size.

follows from this simplicity that the visual comparison between reference and simulated images is easy, especially if the discs do not intersect each other. The discs locations in the image are chosen according to a uniform probability distribution, but the discs are constrained to be disjoint. The algorithm used to construct these images is the following. Firstly, we draw a random disc location from a uniform distribution. If the resulting disc does not intersect any of the previously simulated discs, then we keep it. Otherwise we draw a new disc location. This process is stopped when the total number of discs in the image reaches a predetermined threshold.

Fig. 7 displays simulations based on training images composed of discs with diameters ranging from 4 to 16 pixels. The size of the training images is 256×256 pixels, but the size of the images displayed in **Fig. 7** is 128×128 pixels. The patchwork simulations are performed with the 4×4 model.

In the second line of **Fig. 7**(Q1), we present simulations obtained with the patchwork approach without porosity filtering. Apart from a slight porosity loss, the simulations with $D = 4$ pixels and $D = 8$ pixels are excellent. There are few errors in the simulated discs and the similarity between the reference and the simulation is visually striking. Similar results were obtained for $1 \leq D \leq 8$. However, for $D = 16$ pixels, we observe a significant degradation of the simulations quality. Nevertheless, the general appearance of the simulation remains acceptable.

In the third line of **Fig. 7**(Q2), we present the patchwork simulations obtained with a 10% porosity filtering. The results look similar, apart from the occurrence of a few isolated pixels which do not exist in the reference image. This is explained by the smaller choice of substitution states available when correcting unobserved states. For $D = 4$ pixels, we also observe that the simulation porosity is closer to the reference image porosity. This is a positive effect of the porosity filtering.

In the fourth line of **Fig. 7**(S), we present the simulations obtained with Strebelle's algorithm on the same training images. We see that our patchwork approach produces better results than Strebelle's approach, especially for images containing discs with a small diameter. Indeed, we observe that the proportion of deformed discs in these simulations (**Figs. 7a-S** and **b-S**) is clearly higher than for patchwork simulations.

Fig. 8 displays four simulations obtained with our patchwork approach using a 3×3 pixels model without porosity filtering. The training images contain discs with diameters ranging from 3 to 16 pixels. The simulations obtained with $D = 3$ and $D = 6$ pixels, which are displayed in **Figs. 8a** and **b** respectively, are good. The absence of porosity filtering results in a slight porosity loss. Note that for $D = 3$ pixels the small number of pixels implies that the discs are actually squares. For $D > 6$ pixels (**Figs. 8c** and **d**), we observe that the discs are definitely not reproduced accurately because several large deformed discs are present. These results indicate that the patchwork model produces good quality simulations if the disc diameter is smaller than twice the block size. Above this limit, the simulation quality degrades significantly.

3.2.2. Discs of variable diameters

We tested the two models on images containing discs with different diameters within the same image, as illustrated in **Fig. 9**. These images were constructed as follows. First, we choose randomly a disc location from a distribution which is uniform in the image square. Next, we generate a disc diameter D that takes D_{\max} equiprobable values, where D_{\max} is the maximum diameter and D is an integer such that $1 \leq D \leq D_{\max}$. If the disc generated does not intersect any of the previously simulated discs, then we keep it. Otherwise, a new disc location is chosen and another disc diameter is generated. These steps are repeated until the total number of discs reaches a predetermined threshold. The size of the training images used is 256×256 pixels and the images displayed in **Fig. 9** have a size of 128×128 pixels.

The second column of **Fig. 9** presents the simulations obtained with the patchwork approach based on a 4×4 pixels model with a 10% porosity filtering. For $D_{\max} \leq 8$ pixels, the simulations are good. However, the simulation quality degrades for $D_{\max} > 8$ pixels. These results show that the simulations quality is primarily limited by the maximum disc diameter. We observe that the proportion of large discs is reduced with respect to the reference images, and a porosity loss occurs.

The results obtained with Strebelle's algorithm on the same training images are presented in the third column of **Fig. 9**.

Comparing the second and third columns of **Fig. 9**, it is clear that the patchwork approach gives better results. The simulations using Strebelle's algorithm exhibit a larger number of deformed discs and the degree of deformation of these discs is also higher. Moreover, we observe an excess of oversized deformed discs. In **Fig. 9c**, our patchwork approach preserves better the discs shape but Strebelle's algorithm reproduces better the number of large discs and the global porosity of the image.

3.3. Simulations using training images composed of sticks

We first tested the two methods on training images composed of sticks of constant length and varying orientation. These images are slightly more complex than images composed of discs because each stick is characterized by four parameters (length, orientation and two position coordinates). The simple geometry of sticks facilitates the visual comparison between the simulations and the training images, especially if the sticks do not intersect each other.

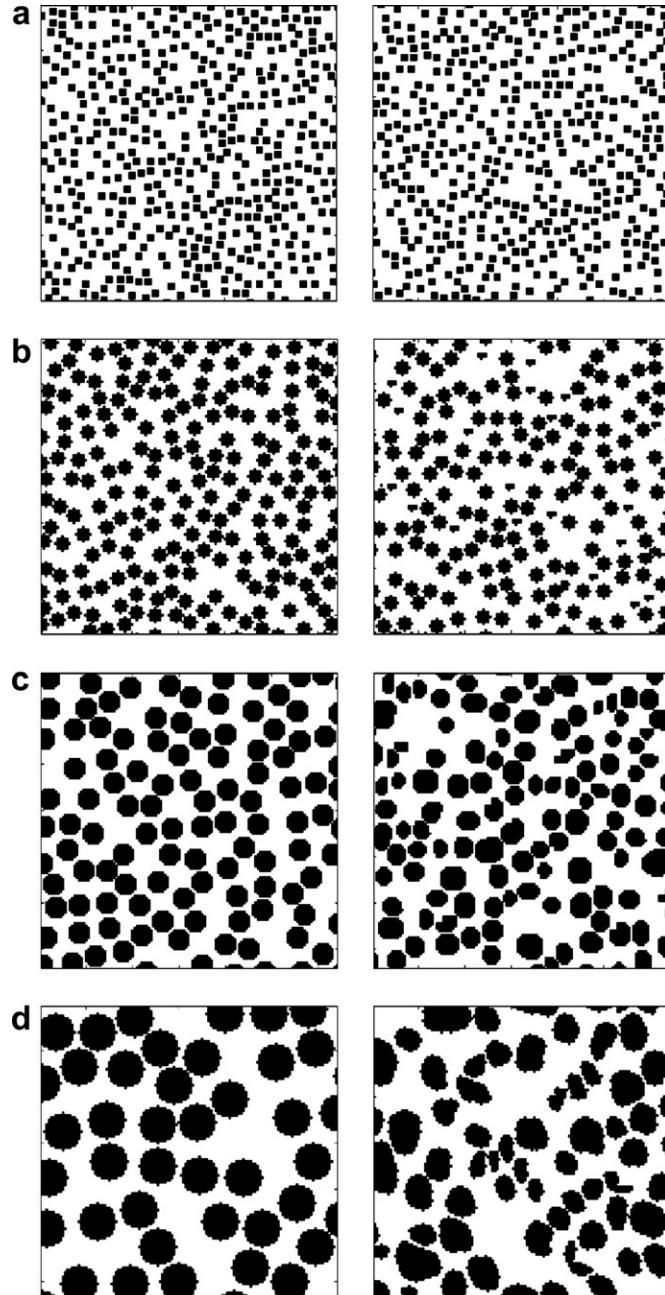


Fig. 8. patchwork simulations based on training images composed of discs with a fixed diameter D : (a) $D = 3$ pixels; (b) $D = 6$ pixels; (c) $D = 9$ pixels; (d) $D = 16$ pixels. The left column contains the reference images and the right column the corresponding simulations using a 3×3 pixels model without porosity filtering. The size of the images displayed is 128×128 pixels, whereas the training images have a size of 256×256 pixels.

The sticks locations in the training image are chosen according to a uniform probability distribution, but the sticks are constrained to be disjoint. The stick orientation is chosen randomly with equal probability among the four inclination angles 0° , 45° , 90° and 135° . Fig. 10 displays simulations based on training images of 256×256 pixels containing sticks of length ranging from 4 to 16 pixels. The thickness of each stick is one pixel. The second line of Fig. 10 contains the patchwork simulations based on a 4×4 model with a 10% porosity filtering. The third line of Fig. 10 contains the

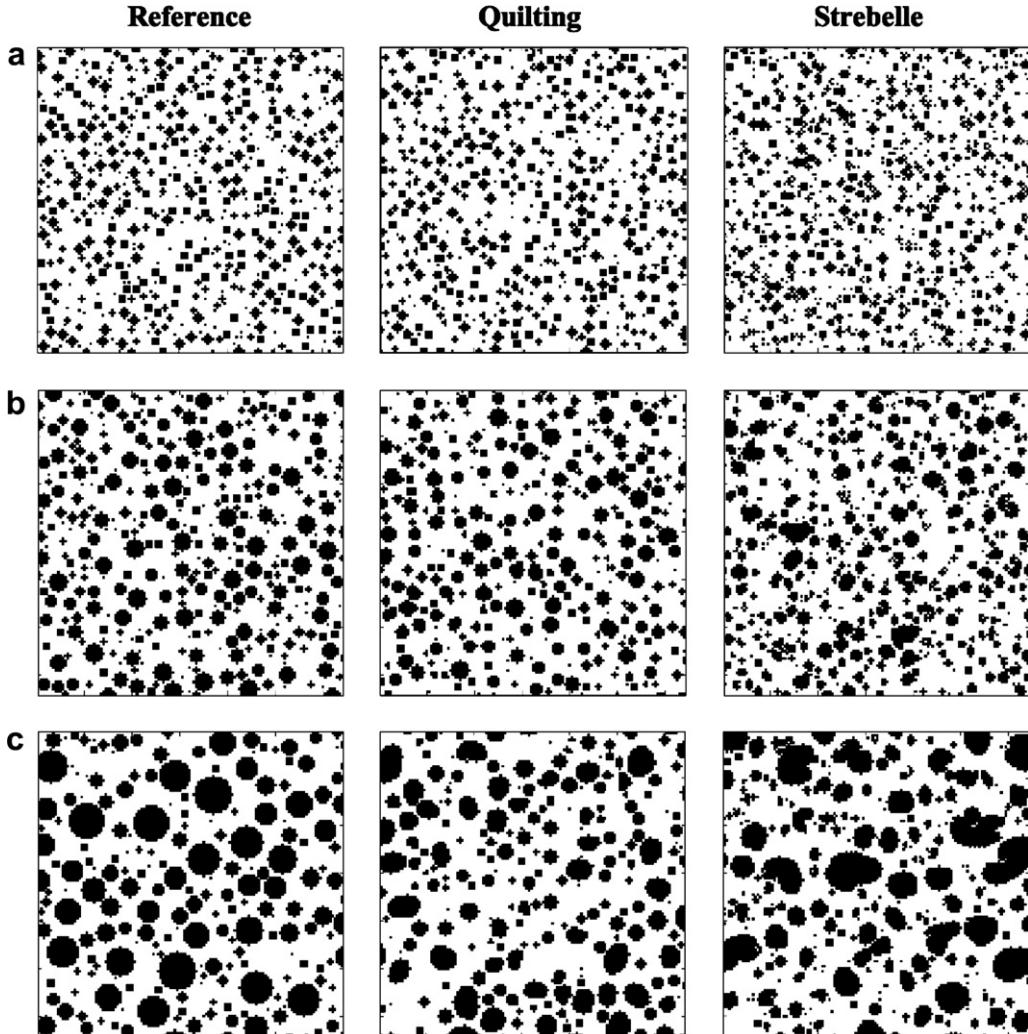


Fig. 9. Simulations based on training images composed of discs with a varying diameter size D : (a) $1 \leq D \leq 4$ pixels; (b) $1 \leq D \leq 8$ pixels; (c) $1 \leq D \leq 16$ pixels. The first column contains the reference images. The second column contains the patchwork simulations using the 4×4 pixels model with a 10% porosity filtering. The third column contains the simulations using Strebelle's algorithm using a maximum of 60 points for conditioning and six multigrids. The size of the images displayed is 128×128 pixels, whereas the training images have a size of 256×256 pixels.

patchwork simulations based on a 4×4 model without porosity filtering. The fourth line of Fig. 10 displays the simulations obtained with Strebelle's algorithm.

The patchwork simulations obtained without porosity filtering (Fig. 10Q2) suffer from a porosity loss. The simulations quality decreases when the stick length is larger than the block size (4 pixels). The stick length distribution is good for $L = 4$, i.e. most sticks have the correct length $L = 4$ pixels, but degrades for $L = 8$ and 16 pixels. Indeed, simulated sticks are often longer or shorter than the sticks in the reference image. The patchwork simulations obtained with a 10% porosity filtering (Fig. 10Q1) have a porosity which is comparable to the reference image porosity. In Figs. 10c-Q1 and c-Q2, the abundance of undersized sticks orientated in the north-east direction is striking.

The simulations obtained with Strebelle's algorithm (Fig. 10S) have an appropriate porosity because the algorithm allows a partial control on porosity. However, the simulations suffer from an excess of undersized sticks in all three cases, and an almost complete absence of long sticks. Moreover, we observe that many of the simulated sticks fail to be disjoint.

We tested the models on images containing sticks with varying lengths and orientations. The stick length L is chosen randomly and takes one of L_{\max} equiprobable values, where L_{\max} is the maximum stick length and L is an integer such that $1 \leq L \leq L_{\max}$. For patchwork simulations, we observe in Fig. 11 that the simulations quality degrades if L_{\max} is

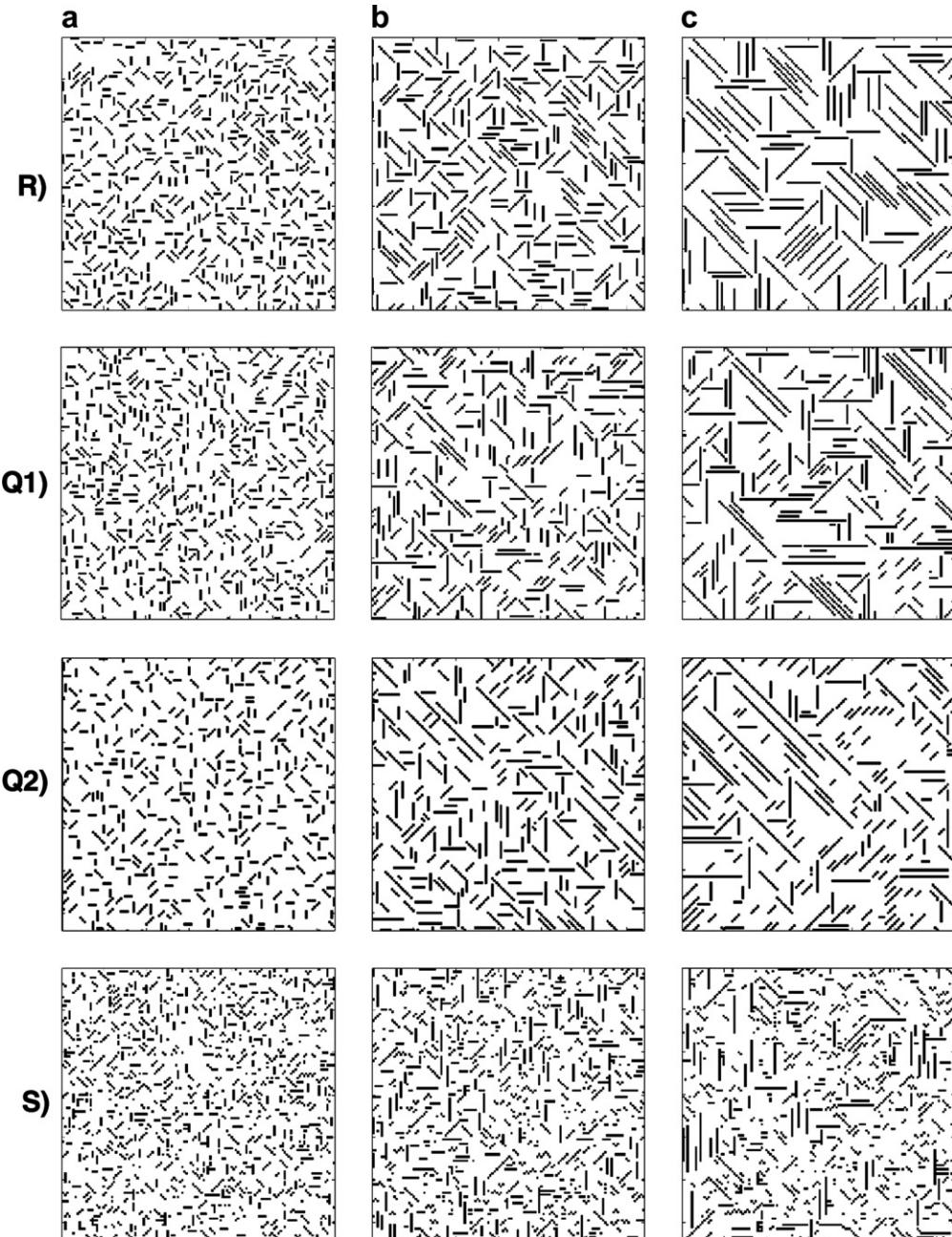


Fig. 10. Simulations based on training images composed of sticks of a fixed length L : (a) $L = 4$ pixels; (b) $L = 8$ pixels; (c) $L = 16$ pixels. The patchwork simulations are based on the 4×4 pixels. The first line contains the training images (R). The second line (Q1) contains the patchwork simulations with a 10% porosity filtering. The third line (Q2) contains the patchwork simulations without porosity filtering. The fourth line (S) contains the simulations produced with Strebelle's algorithm using a maximum of 60 points for conditioning and four multigrids. The size of the images displayed is 128×128 , whereas the size of the training images is 256×256 pixels.

larger than the block size (4 pixels). We also observe a porosity loss for the patchwork simulations obtained without porosity filtering. The simulations obtained with Strebelle's algorithm again suffer from an excess of undersized sticks and intersecting sticks, as well as a deficit of long sticks.

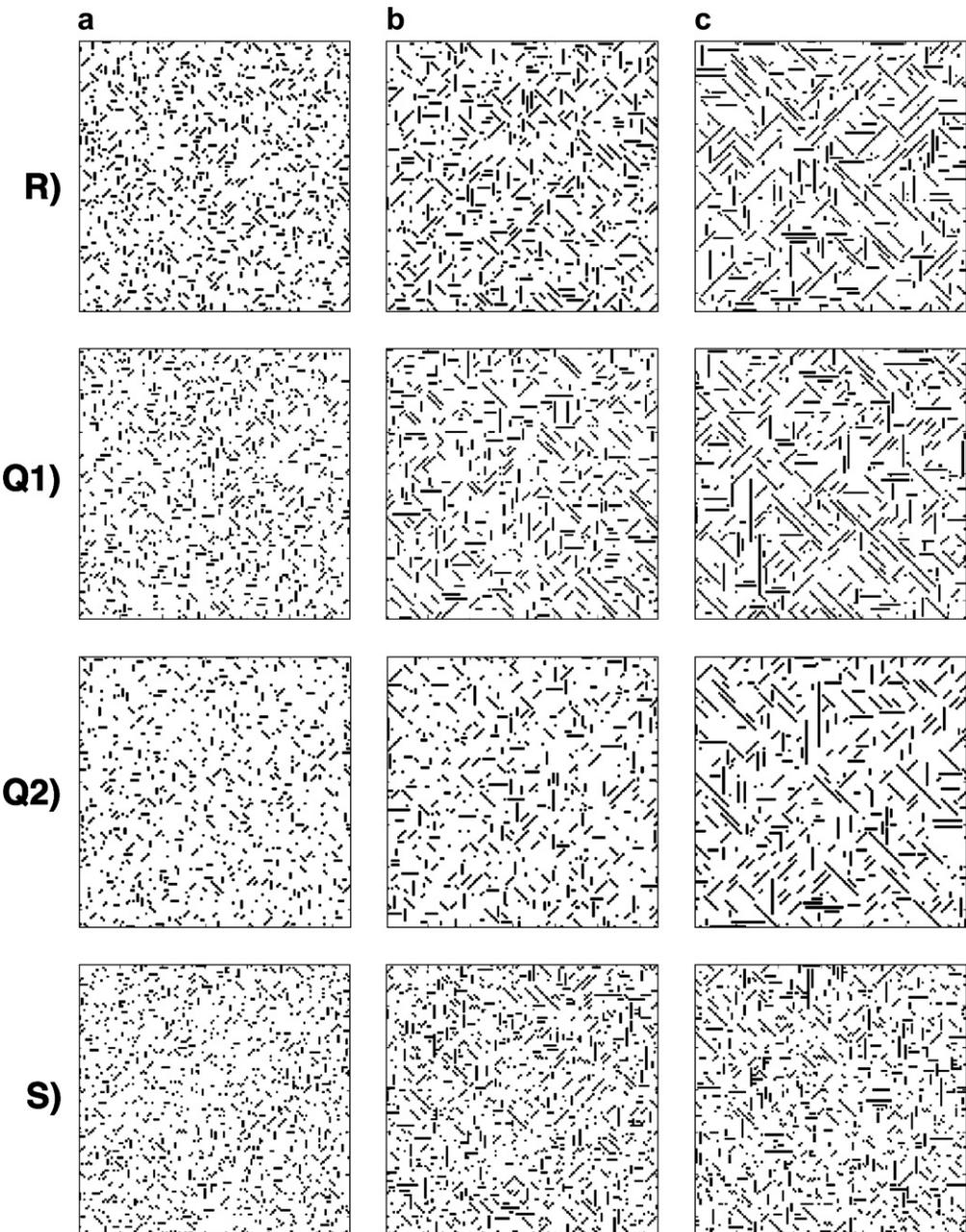


Fig. 11. Simulations based on training images composed of sticks with varying stick length L : (a) $1 \leq L \leq 4$ pixels; (b) $1 \leq L \leq 8$ pixels; (c) $1 \leq L \leq 16$ pixels. The patchwork simulations are based on the 4×4 pixels. The first line contains the training images (R). The second line (Q1) contains the patchwork simulations with a 10% porosity filtering. The third line (Q2) contains the patchwork simulations without porosity filtering. The fourth line (S) contains the simulations produced with Strebelle's algorithm using a maximum of 60 points for conditioning and four multigrids. The size of the images displayed is 128×128 , whereas the size of the training images is 256×256 pixels.

3.4. Simulations using a training image containing channels

In this section, we compare our approach and Strebelle's algorithm on an image which contains objects having a size comparable to the training image size. The objects chosen mimic the shape of a channel, as shown in Fig. 12R. Fig. 12 displays simulations obtained with the two approaches. The patchwork simulation uses a 4×4 pixels model. Unlike all

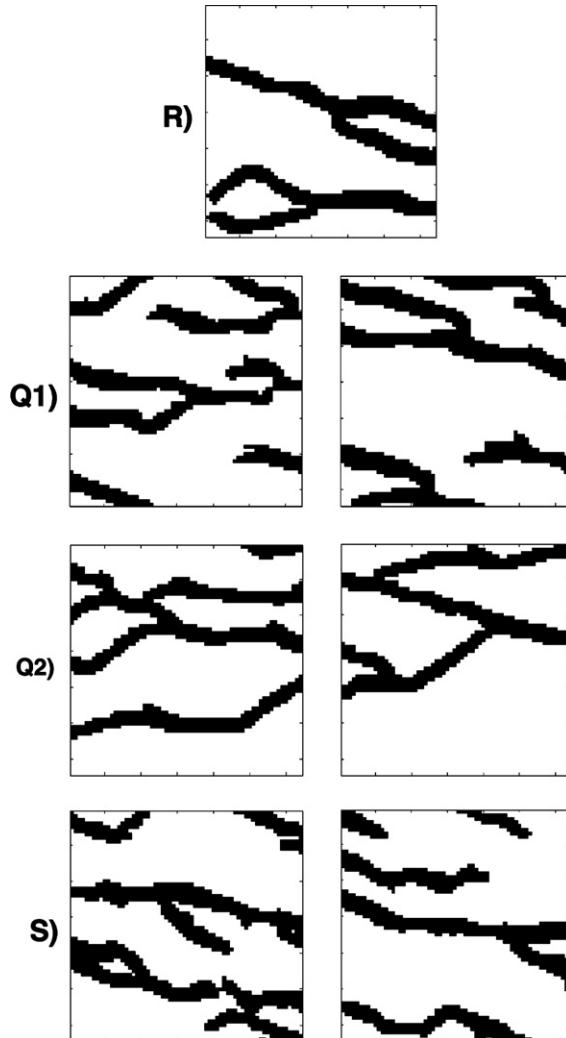


Fig. 12. Simulations based on the reference image (R) which is composed of two channels: (Q1) patchwork simulations with a 10% porosity filtering; (Q2) patchwork simulations without porosity filtering; (S) simulations based on Strebelle's algorithm. The patchwork simulations are based on a 4×4 pixels model. Strebelle's algorithm was used with a maximum of 60 conditioning points and five multigrids. The size of the images displayed is 64×64 pixels.

the previous patchwork simulations, we scanned the training image in a single direction because the image is anisotropic. The simulations obtained with Strebelle's algorithm use 60 conditioning points and five multigrids.

For both methods, the simulated channels look similar to the training image channels. Best results are obtained in Fig. 12Q2 where the channels are continuous across the whole image, as in the reference image. In Figs. 12Q1 and S, channels appear (or disappear) abruptly within the simulated window.

3.5. Simulations based on an electron microscope polymer blend image

A polymer blend is a material obtained by mixing two or more single phase polymers [23,24]. The vast majority of polymer blends are not miscible in the thermodynamic sense and hence form multiphase systems resembling oil-in-water type emulsions. For immiscible polymers, there are two main morphologies for the blend [17]: dispersed or co-continuous. In the co-continuous morphology, each polymer is fully interconnected through a complex microstructure. The understanding of this morphology is important for the development of new materials based on polymer blends and the potential applications include: carriers for conductive polymers, membranes and scaffolds for tissue engineering and drug delivery.

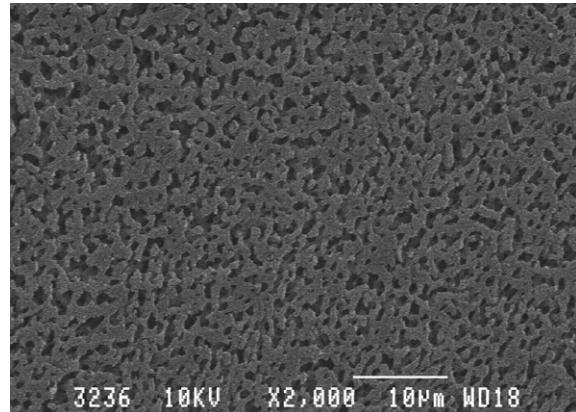


Fig. 13. Electron microscope image of a co-continuous polymer blend. The image size is 694×503 pixels.

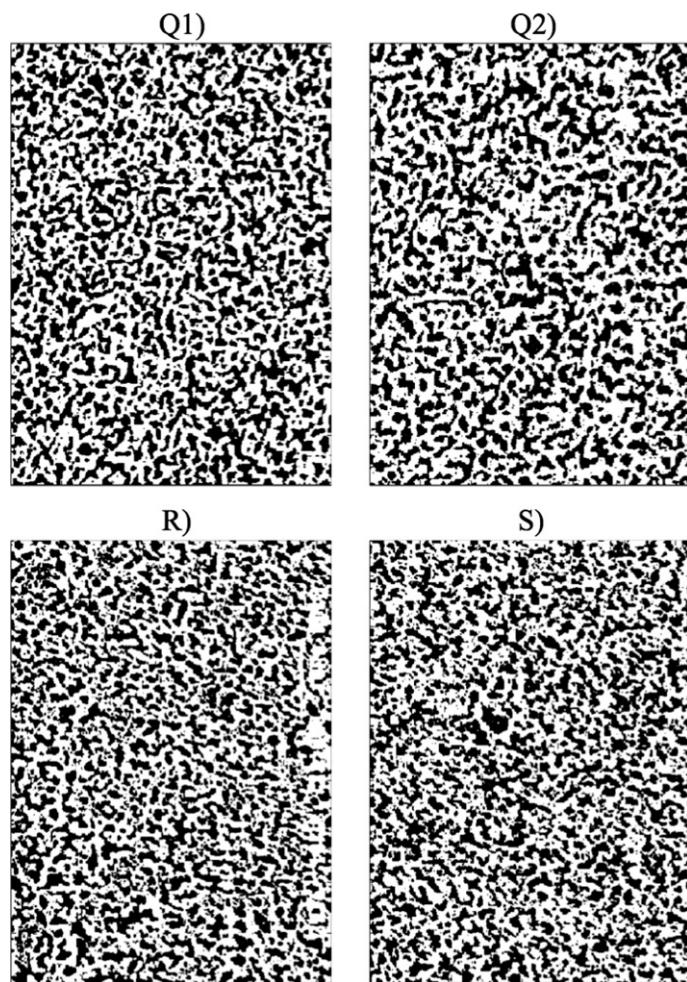


Fig. 14. Simulations using a polymer blend microscopic image as a training image: (R) reference image; (Q1) patchwork simulation using a 4×4 model and a 10% porosity filtering; (Q2) patchwork simulation using a 7×7 model and a 10% porosity filtering; (S) simulation obtained with Strebelle's algorithm using 60 conditioning points and four multigrids. The size of these images is 314×434 pixels.

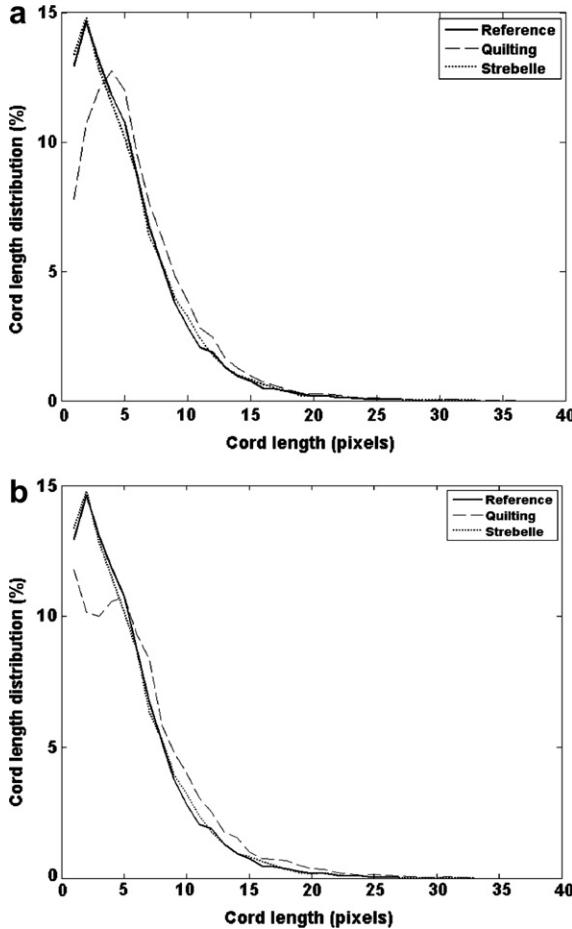


Fig. 15. Cord length distributions of the four images in Fig. 14: (a) patchwork simulation using the 4×4 model; (b) patchwork simulation using the 7×7 model.

The plastic industry shows an increasing interest for polymer blends because of their economical importance. They are low cost materials with superior properties compared with monophase polymers.

In this section, we focus on the stochastic simulation of two-dimensional cuts of a co-continuous polymer blend. More precisely, we test our model on a electron microscope image of a co-continuous polymer blend. The image (Fig. 13) was produced by microtomy and scanning electron microscopy (SEM).

Fig. 13 displays an electron microscope image of a blend composed of polystyrene and polycaprolactone (PS/PCL) in volumic proportions of 50%–50%. The specimens were microtomed to create a perfect plane face using a Reichert Jung 2050 Supercut Microtome with a glass knife equipped with the Liquid Nitrogen Freezing Device LN20. In order to improve the contrast in the image, the PLC was extracted with acetic acid. After selective solvent extraction and coating with a gold–palladium alloy, the microtomed specimens were observed under a Jeol JSM 840 scanning electron microscope at a voltage of 10 kV.

The original image displayed in Fig. 13 was filtered to produce a training image for our model. A threshold was applied to the image after noise removal and contrast enhancement. The size of the training image displayed in Fig. 14R is 314 × 434 pixels. The patchwork simulations using a 4×4 and a 7×7 model with a 10% porosity filtering are displayed in Figs. 14Q1 and Q2 respectively. The similitude between the training image and the simulations is visually striking. However, the white channels appear to be broader in the patchwork simulations than in the reference image. In addition, we observe an excess of large black clusters, especially with the 7×7 model. The simulation obtained with Strebelle's algorithm using 60 conditional points and four multigrids is displayed in Fig. 14S. This simulation looks more similar to the training image than the patchwork simulation. In particular, we observe that the number of small black grains found in the reference image is better reproduced.

To compare quantitatively the simulations with the polymer blend training image, we measured the porosity and the cord length distribution (CLD) of the images. Cords are black segments of thickness one pixel which are found at the intersection of the image with a straight line. The cord lengths are measured in the horizontal and vertical directions only. The porosities of the patchwork simulations are 42.8% (4×4 model) and 41.7% (7×7 model), whereas the porosity of the simulation with Strebelle's method is 42.9%. These porosities are close to the training image porosity (42.3%). Fig. 15 displays the CLDs for the four images in Fig. 14. Comparing the CLDs displayed in Fig. 15, we observe that the CLD associated with Strebelle's algorithm is closest to the training image CLD. The CLD associated with the patchwork simulation suffers from a deficit of small cord lengths and a slight excess of long cord lengths.

4. Discussion

The simulations presented in this paper on several types of training images show the strengths and limitations of our patchwork approach. We showed that the patchwork method performs very well on checkerboards or on images composed of small discs. For such images, patchwork simulations give better results than Strebelle's algorithm. Indeed, we have shown that the checkerboards with a mesh size smaller or equal to the block size are perfectly reproduced. Moreover, images containing discs of a fixed or varying diameter are reproduced with a good quality if the maximum diameter is smaller than twice the block size. Our patchwork approach is primarily limited by the relationship between the size of the objects we want to reproduce and the block size. The quality of the reconstructions decreases when the difference between these two parameters increases.

For images composed of long sticks, Strebelle's algorithm failed completely whereas the patchwork approach failed to generate long sticks in the 45° direction. This directional bias is linked to the inverse "L" shape of the conditioning pattern (Fig. 2c). Let us suppose that the upper part of a long stick in the 45° direction is simulated in a block labeled by A. The lower part of this stick should intersect a block B located on the lower left side of block A. However, block B will be simulated using only the knowledge of blocks which are located on the left side of block A, thus without the knowledge that the beginning of a long stick exists in block A. Consequently, the sticks simulated in block B will only match by chance the sticks simulated in block A, explaining why so many small sticks appear in that direction. Images could be rotated to alleviate the directional bias induced by the L-shape model.

The simulations of channels were comparable for both methods. This shows that in some cases the patchwork approach can simulate correctly structures which are much longer than the block size.

For the microscopic image of a polymer blend, Strebelle's algorithm produced a slightly better simulation than the patchwork approach. In particular, the cord length distribution of the patchwork simulations exhibits a moderate excess of long cord lengths with respect to the reference image. In this image, the presence of clusters which are much larger than the block size explains the poorer performance of the patchwork simulations, which are not designed to accurately reproduce large-scale features.

Both approaches require a good estimation of the transition probabilities to produce good results. To reach this goal, the size of the training images has to be large compared with the block size. With small training images, the transition probabilities are poorly estimated and the simulation results have a lower quality.

The block based approach presented in this article uses black and white images only. This can be easily improved by changing the binary coding of the states to a base p coding where p is the number of possible states at each pixel. In the patchwork approach, conditioning to hard data is possible on a local basis only by choosing among the arrival states that fits the hard data within the currently simulated block.

Generalization of the patchwork approach to three-dimensional (3D) images is straightforward when 3D training blocks are available. However, the real interest would rather be to extrapolate to 3D in a reasonable way from a series of 2D training images. For the moment, no satisfying model or technique exists to accomplish this challenging task.

5. Conclusion

We presented in this paper a new sequential method of simulation based on the estimation of transition probabilities between adjacent blocks composing the training image. Our approach has the ability to reproduce correctly many morphological features because the blocks used in the patchwork process, which are drawn directly from the training image, are essentially correct.

We compared the performance of patchwork simulations with Strebelle's multipoint simulation algorithm on several types of images of increasing complexity. For images composed of clusters which are small with respect to the block size (e.g. squares, discs and sticks), our patchwork approach produces better results than Strebelle's method. The most noticeable improvement is that the cluster geometry is usually reproduced accurately. In general, the simulation quality

degrades when the maximum cluster size is larger than the block size. The patchwork approach was applied to the training image of a co-continuous polymer blend. The resulting simulations are good visually but the CLD associated with the patchwork simulation suffers from a deficit of small cord lengths and a slight excess of long cord lengths. In the polymer blend image, the presence of clusters which are much larger than the block size explains the poorer performance of patchwork simulations, which are not designed to accurately reproduce large-scale features.

Although the patchwork approach seems promising, some questions need further research, such as: (1) how to control the probability distribution of the porosity at the block scale; and (2) how to alleviate the interaction of directional features with the shape of the conditioning pattern. Once these questions resolved, the formidable task of generalizing from 2D images to 3D blocks will have to be examined carefully.

Acknowledgements

A. Saucier, D. Marcotte and B.D. Favis acknowledge financial support from individual NSERC grants. The authors thank Dr. P. Sarazin for providing the polymer blend micrograph used in the analysis.

References

- [1] Adler PM. Porous media. Boston: Butterworth-Heinemann; 1992.
- [2] Adler PM, Jacquin C, Quiblier CJ. Flow in simulated porous media. *Int J Multiphase Flow* 1990;16:691–712.
- [3] Arpat B. Report No. 16: A pattern recognition approach to multiple-point simulation. Stanford Center for reservoir forecasting. Stanford University; 2003.
- [4] Bakke S, Øren P. 3-D pore-scale modeling of sandstones and flow simulations in pore networks. *SPE J* 1997;2:136.
- [5] Bear J, Verruijt A. Modeling groundwater flow and pollution. Dordrecht: Kluwer Academic Publishers; 1987.
- [6] Bryant S, Mellor D, Cade C. Physically representative network models of transport in porous media. *AIChE J* 1993;39:387.
- [7] Caers J. Stochastic simulation using neural networks. Stanford Center for reservoir forecasting annual meeting 11, vol. 2. Stanford University; 1998. p. 1–66.
- [8] Caers J, Journel AG. Stochastic reservoir simulation using neural networks trained on outcrop data. *SPE paper no. 49026*; 1998.
- [9] Chatzis I, Dullien F. Modelling pore structure by 2-d and 3-d networks with applications to sandstones. *J Canad Petrol Technol* 1977;97.
- [10] Stoyan D, Kendall W, Mecke J. Stochastic geometry and its applications. 2nd ed. Berlin, Chichester: Wiley; 1987.
- [11] Deutsch C, Journel A. GSLIB: geostatistical software library. 2nd ed. New York: Oxford University Press; 1998.
- [12] Fatt I. The network models of porous media i. capillary pressure characteristics. *AIME Petrol Trans* 1956;207:144.
- [13] Guardiano F, Srivastava RM. Multivariate geostatistics: beyond bivariate moments. In: Soares A, editor. Geostatistics-Troia, vol. 1. Dordrecht, Netherland: Kluwer Academic; 1993. p. 133–44.
- [14] Haldorsen H, Damsleth E. Stochastic modeling. *J Petrol Technol* 1990;40:4–12.
- [15] Katz A, Thompson A. Quantitative prediction of permeability in porous rock. *Phys Rev B* 1986;34:8179.
- [16] Lake L. Enhanced oil recovery. Englewood Cliffs: Prentice-Hall; 1989.
- [17] Li J, Ma PL, Favis BD. The role of the blend interface type on morphology in cocontinuous polymer blends. *Macromolecules* 2002;35:2005–16.
- [18] Omre H. Stochastic models for reservoir characterization. In: Kleppe J, Skjæveland SM, editors. Recent advances in improved oil recovery methods for north sea oil sandstone reservoirs. Dordrecht, Netherland: Norwegian Petroleum Directorate; 1991.
- [19] Remy N. Stanford Geostatistical Modeling Software (S-GeMS), version 1.2, build September 22, 2004. Contributors: Joy Rajiv and Michael Sharps. Available from: <http://prdownloads.sourceforge.net/sgems/sgemsInstaller-1.2.exe?download>.
- [20] Roberts J, Shwartz L. Grain consolidation and electrical conductivity in porous media. *Phys Rev B* 1985;31:5990.
- [21] Roy S, Trafaldar S. Archie's law from a fractal model for porous rock. *Phys Rev B* 1997;55:8038.
- [22] Sahimi M. Flow phenomena in rocks: from continuum models to fractal, percolation, cellular automata and simulated annealing. *Rev Mod Phys* 1993;65:1393.
- [23] Sarazin P, Favis BD. Morphology control in co-continuous poly (L-lactide)/polystyrene blends: a route towards highly structured and interconnected porosity in poly (L-lactide) materials. *Biomacromolecules* 2003;4:1669–79.
- [24] Sarazin P, Favis BD. Influence of temperature-induced coalescence effects on co-continuous morphology in poly (ε-caprolactone)/polystyrene blends. *Polymer* 2005;46:5966–78.
- [25] Saucier A, Muller J. Remarks on some properties of geometrical multifractals. *Physica A* 1993;199:350–62.
- [26] Saucier A, Muller J. Textural analysis of disordered materials with multifractals. *Physica A* 1999;268:221–38.
- [27] Saucier A, Richer J, Muller J. Assessing the scope of the multifractal approach to textural characterization with statistical reconstruction of images. *Physica A* 2002;311(1–2):231–59.
- [28] Strebel S. Conditional simulation of complex geological structures using multiple point statistics. *Math Geol* 2002;34(1):1–22.
- [29] Strebel S. New multiple-point statistics simulation implementation to reduce memory and cpu-time demand. In: Presented at the 2003 conference of the international association for mathematical geology. Portsmouth, UK, September 7–12; 2003. Available from: <http://www.iagm.org/meetings/Proceedings-2003/papers/Strebel2.pdf>.
- [30] Torquato S. Modeling of physical properties of composite materials. *Int J Solids Struct* 2000;37:411–22.