Journal of Microscopy

Journal of Microscopy, Vol. 0, Issue 0 2016, pp. 1–16 doi: 10.1111/jmi.12441
Received 27 January 2016; accepted 8 June 2016

Characterization and reconstruction of 3D stochastic microstructures via supervised learning

R. BOSTANABAD*, W. CHEN* & D.W. APLEY†
*Department of Mechanical Engineering, Northwestern University, Evanston, Illinois, U.S.A
†Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois, U.S.A

Key words. Characterization and reconstruction, statistical equivalency, 3D, stochastic microstructure, supervised learning.

Summary

The need for computational characterization and reconstruction of volumetric maps of stochastic microstructures for understanding the role of material structure in the processing–structure–property chain has been highlighted in the literature. Recently, a promising characterization and reconstruction approach has been developed where the essential idea is to convert the digitized microstructure image into an appropriate training dataset to learn the stochastic nature of the morphology by fitting a supervised learning model to the dataset. This compact model can subsequently be used to efficiently reconstruct as many statistically equivalent microstructure samples as desired. The goal of this paper is to build upon the developed approach in three major directions by: (1) extending the approach to characterize 3D stochastic microstructures and efficiently reconstruct 3D samples, (2) improving the performance of the approach by incorporating user-defined predictors into the supervised learning model, and (3) addressing potential computational issues by introducing a reduced model which can perform as effectively as the full model. We test the extended approach on three examples and show that the spatial dependencies, as evaluated via various measures, are well preserved in the reconstructed samples.

Introduction

Recent advances in imaging techniques (Edelman & Warach, 1993; Salvo *et al.*, 2003; Kastner *et al.*, 2012) have enabled the collection of high resolution three-dimensional (3D) maps of microstructures at various scales. The need for computational characterization and reconstruction (C&R) of the collected data for investigating the role of material structure in the processing–structure–property chain is well highlighted in the literature (Olson, 1997, 2000; Torquato, 2002a,b. 2010; Fullwood *et al.*, 2010; Niezgoda *et al.*, 2010, 2011; Ward, 2012; Breneman *et al.*, 2013; Xu *et al.*, 2013; Baghgar *et al.*, 2014; Qomi *et al.*, 2014, 2015; Bauchy *et al.*, 2014; Ballani & Stoyan, 2015; Zhang *et al.*, 2015). Given the digitized image of a microstructure sample, C&R provide the means for building a virtual ensemble of representative volume elements or statistical volume elements that can be used for material property prediction (Greene *et al.*, 2011). The accuracy of the estimated properties hinges on the *statistical equivalency* between the ensemble members and the original microstructure.[1]

To build a high fidelity ensemble whose elements are indeed statistically equivalent to the original microstructure, one needs to make sure (1) the inherent stochasticity of the original sample is identified and accurately characterized (or, in machine learning parlance, learned), and (2) the learned stochasticity is well preserved when generating (reconstructing) virtual samples. Motivated by these challenges, a handful of methods have previously been developed for microstructure C&R. Here, we briefly discuss methods that can be applied to 3D microstructure C&R and refer the reader to the references for more details.

Optimization-based approaches (and its variants) are one of the most popular and well-established methods in this area. Works that fall into this category iteratively adjust (i.e. optimize) the reconstructed image so as to minimize an appropriately defined energy (cost) function that measures the statistical differences between the original image and the reconstructed one. A subset of these approaches, often known as stochastic reconstruction (Yeong & Torquato, 1998a,b), characterize the material structure via various correlation functions (such as two-point and two-point cluster, see Debye & Bueche, 1949; Debye *et al.*, 1957; Beran, 1965; Corson, 1974a,b; Torquato & Stell, 1982; Berryman, 1985; Berryman & Blair, 1986; Rintoul & Torquato, 1997; Yeong & Torquato, 1998a,b; Torquato, 2002a,b, 2006, 2010; Jiao *et al.*, 2007,

Correspondence to: Daniel W. Apley, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL 60208, U.S.A. Tel: 847-491-2397; fax: 847-491-8005; e-mail: Apley@northwestern.edu

[1] Throughout the paper we use the terms 'image' and 'microstructure' interchangeably, because the samples are represented as images.

© 2016 The Authors
Journal of Microscopy © 2016 Royal Microscopical Society

2008, 2009; Li *et al.*, 2012; Liu *et al.*, 2013) and the difference between the correlation functions of the original and the reconstructed image is the basis for the cost function. Because most microstructures cannot be characterized solely by one specific correlation function, usually multiple correlation functions are incorporated into the cost function. Stochastic reconstruction methods are applicable to 3D microstructures and, with reasonable assumptions (i.e. isotropy and homogeneity), can also extrapolate two-dimensional (2D) images to reconstruct 3D microstructures. Recently, Zachary & Torquato (2011) and Guo *et al.* (2014) integrated the stochastic reconstruction method with dilation/erosion techniques in image analysis to better match the clusteredness by only integrating the two-point correlation function into the aforementioned cost function. Although several improvements in the pixel-/voxel-swapping heuristics have been developed (see Tang *et al.*, 2009; March *et al.*, 2012; Piasecki & Olchawa, 2012; Chen *et al.*, 2014; Pant *et al.*, 2014), the optimization is still computationally prohibitive for reconstructing large/anisotropic images. Methods based on phase recovery (see, e.g. Fullwood *et al.*, 2008a,b) can also be categorized as optimization-based, as they iteratively optimize a sample to match its full vector two-point correlation function (i.e. with phase information) to a target full vector two-point correlation function. Because the phase information is preserved, characterization of anisotropic and polycrystalline materials is to some extent possible. Phase recovery methods reconstruct microstructures with periodic boundaries because they use a fast Fourier transform, and the reconstruction is deterministic (i.e. only one sample can be reconstructed with given target statistics and if the target full vector two-point correlation function completely characterizes the original structure, the original structure will be recovered upon reconstruction).

In another set of optimization approaches, various physical descriptors (e.g. average particle size or grain boundaries) are used to characterize the original microstructure, and the optimization process aims to match the characteristics of the descriptors in the reconstructed image to the corresponding ones in the original image (Karasek & Sumita, 1996; Tewari & Gokhale, 2004; Rollett *et al.*, 2007; Wilson *et al.*, 2010; Holotescu & Stoian, 2011; Sintay & Rollett, 2012; Xu *et al.*, 2014). The choice of physical descriptors depends on the material system (e.g. polycrystalline, fibre composite, ...) and properties of interest. The use of physical descriptors enables sensible characterization of topological features, and hence this approach is desirable for design (see, e.g. Breneman *et al.*, 2013). However, image analysis is required to extract the characteristics of the descriptors from the original image, and one needs to have *a priori* knowledge of the appropriate descriptors. For example, because transport processes in particulate heterogeneous systems and intergranular corrosion in polycrystalline materials are sensitive to, respectively, nearest neighbour distances between particles (Torquato, 2002b) and grain boundaries (Rollett *et al.*, 2007), one needs to match the distributional characteristics (such as the mean and variance) of these descriptors in the original and reconstructed samples.

Poisson (Grigoriu, 2003; Rahman, 2008) or Gaussian (Quiblier, 1984; Levitz, 1998; Talukdar *et al.*, 2002; Tang *et al.*, 2008; Jiang *et al.*, 2013) random fields have also been previously used to model stochastic (esp. porous) materials by level-cutting a random field. These methods rely on correlation functions because the model is fitted by matching its correlation functions to those of the original image. The methodology works in 2D, 3D and 2D to 3D (i.e. characterization in 2D and reconstruction in 3D) but is usually restricted to bi-phase isotropic structures based on only two-point correlation function and hence lacks high accuracy. In Jiang *et al.* (2013), Tang *et al.* (2008) and Talukdar *et al.* (2002), the authors optimized a random field-based reconstructed image (e.g. via simulated annealing) to reduce the differences between correlation functions. Although this integration improves the accuracy (in matching correlation functions), it is still subject to the above limitations (e.g. restriction to bi-phase isotropic structures). Some of these methods (see Rahman, 2008) are applicable to statistically inhomogeneous structures and can be extended to consider higher-order point correlation functions but are limited to specific material systems (e.g. penetrable ellipsoidal particles) and involve calibration and Mote Carlo sampling to, respectively, match the moments of the random field to the target statistics, and generate realizations from the filtered Poisson field.

Texture[2] synthesis methods (Wei & Levoy, 2000; Efros & Leung, 1999) have also been applied to material C&R. In this approach, the microstructure is assumed to behave as a stationary Markov random field (MRF; see "Review of the 2D supervised learning approach for stochastic microstructure C&R" section), the characterization is implicit and each image is reconstructed pixel-by-pixel (voxel in 3D) in a specific order (e.g. raster scan). Each pixel's value in the reconstructed image is found by searching for the pixel (or set of pixels) in the original image whose neighbouring pixels best match the neighbours of the pixel to be generated. The methods differ in their choice of neighbourhood geometry, definition of similarity and search method (see, e.g. Sundararaghavan, 2014; Liu & Shapiro, 2015). Texture synthesis-based methods are applicable to various 2D/3D material systems but involve no fitted model to characterize the microstructure, and reconstruction is done by exhaustively searching for similar neighbourhoods in the original image. The exhaustive search makes these methods computationally prohibitive for microstructures with long-range correlations or high resolution samples where a relatively large neighbourhood is required in the search process. We note that some texture synthesis methods (Efros & Freeman, 2001) use patches instead of pixels and hence are

[2] In materials science, the preferred crystallographic orientation in a polycrystalline structure is sometimes referred to as a texture. In the cited works, however, texture is defined as a realization of a stationary random field.

© 2016 The Authors

computationally more efficient (see Tahmasebi *et al.* (2012) and Tahmasebi & Sahimi (2013) for similar works in geostatistics).

Works similar to texture synthesis also exist in geostatistics literature (Guardiano & Srivastava, 1993; Caers & Journel, 1998; Elfeki & Dekking, 2001; Strebelle, 2002; Wu *et al.*, 2004; Okabe & Blunt, 2005; Hajizadeh *et al.*, 2011) where C&R of porosity in geological structures (such as sandstone) is of particular interest. The methodology used in these works is based on multiple-point statistics, characterizing the structure by calculating and storing the conditional probabilities of finding a specific phase at a pixel, given the phases of a particular configuration of neighbouring pixels. The multiple-point statistics approach implicitly characterizes the microstructure by exhaustive enumeration of all possible phase combinations for all possible neighbourhood configurations that have occurred in the training image. Reconstruction is accomplished pixel-by-pixel, also similar to texture synthesis, by searching for the training neighbourhood that best matches that of the pixel being reconstructed and then subsequently sampling from the conditional probability for that neighbourhood. Different methods vary in their choice of neighbourhood geometry, search method and reconstruction order (e.g. random or raster scan).

In texture synthesis and multiple-point statistics methods, the results depend on the neighbourhood size used in the search process (which is oftentimes chosen manually via *ad hoc* methods). Liu & Shapiro (2015) show that the chosen neighbourhood size affects the quality (e.g. the preserved randomness) of the results, and they recommend choosing it to be the range after which the two-point correlation function dies out. In addition, Harrison (2001) shows that the reconstruction order (e.g. a bottom-up raster-scan vs. a top-down raster-scan) could affect the results in texture synthesis (and hence in multiple-point statistics). Addressing these issues in 3D C&R is particularly challenging, not only because 3D microstructures are much more complex than 2D ones, but also because the associated computational costs are significantly higher.

Bostanabad *et al.* (2016) have recently developed a versatile and promising approach for microstructure C&R based on supervised learning that is applicable to a wide range of material systems [e.g. isotropic, anisotropic and porous with high/low volume fraction (VF)] and is much faster than other methods in the literature. The fundamental idea is to first convert the original microstructure image into an appropriate training dataset and then to fit a supervised leaning model to this dataset, for predicting the phase of each pixel as a function of its surrounding pixel phases. This fitted model is subsequently employed for fast and efficient reconstruction of statistically equivalent microstructures with arbitrary size. In this paper, we improve and generalize the approach in three important ways: (1) C&R of 3D microstructure images: as most important material properties (e.g. permeability) require 3D simulations, we extend the approach to characterize 3D images and demonstrate its computational superiority in 3D reconstruction over other existing methods. (2) Incorporation of user-defined predictors into the supervised learning model: in Bostanabad *et al.* (2016), the model predictors consisted of individual pixels and user-defined predictors (i.e. knowledge- or physics-based characteristics involving combinations of pixels) were *not* included into the model. In this paper, we illustrate how such predictors can be defined and incorporated to better learn a particular characteristic (e.g. local VF or connectivity) in the original microstructure image and preserve it in the reconstructed ones. The user-defined predictors can be defined in 2D or 3D and on any microstructure sample; an attractive characteristic of this approach is that if the user-defined predictor does *not* improve the predictive power of the supervised learner, it will be automatically pruned from the model. (3) Addressing potential computational issues that may arise: if the original image is complex and high resolution (large number of pixels/voxels), the aforementioned training dataset will require large memory. This issue is of particular concern in 3D, and in the 'Reduced model' section we demonstrate how the size of the dataset can be reduced with little adverse effect on the results. In the 'Results and discussions' section, we apply the developed approach to three examples: an isotropic clustered nanocomposite, a porous medium and an anisotropic structure. In the 'Conclusion and future work' section, we conclude the paper and elaborate on the challenges and possible future directions.

Review of the 2D supervised learning approach for stochastic microstructure C&R

Denote the collection of pixels in the original microstructure image of size n_1 (rows) $\times$ n_2 (columns) by $\boldsymbol{X}$. The elements of $\boldsymbol{X}$ are indexed in a raster-scan order; starting from the pixel at the top left corner and, progressing line-by-line horizontally, ending at the bottom right pixel. For a bi-phase material, the elements of $\boldsymbol{X}$ are binary, $X_{ij} \in \{0, 1\}$ for $i = 1, 2, \ldots, n_1$, and $j = 1, 2, \ldots, n_2$. As described in Bostanabad *et al.* (2016), $\boldsymbol{X}$ can be thought of as a random sample from its underlying full joint distribution, denoted by $f(\boldsymbol{X})$. From this perspective, reconstruction of a statistically equivalent image ($\boldsymbol{Y}$) of any size requires the estimation of $f(\boldsymbol{X})$, which can be decomposed as:

$$\begin{aligned} f(\boldsymbol{X}) &= f(X_{11})\, f(X_{12}|X_{11})\, f(X_{13}|X_{11}, X_{12}) \cdots \\ &\quad f(X_{n_1 n_2}|X_{11}, X_{12}, \ldots, X_{n_1(n_2-1)}) \\ &= f\left(X_{11}|\boldsymbol{X}^{(<11)}\right) f\left(X_{12}|\boldsymbol{X}^{(<12)}\right) f\left(X_{13}|\boldsymbol{X}^{(<13)}\right) \cdots \\ &\quad f\left(X_{n_1 n_2}|\boldsymbol{X}^{(<n_1 n_2)}\right), \end{aligned} \tag{1}$$

where $\boldsymbol{X}^{(<ij)}$ denotes the set of all the pixels in $\boldsymbol{X}$ ordered before X_{ij}. Equation (1) illustrates that, given a model that

© 2016 The Authors

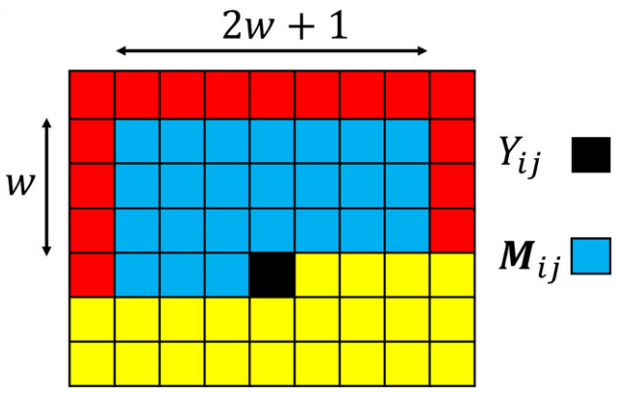

Fig. 1. (Colour online) A subset of the pixels indexed (in a left-to-right then top-to-bottom raster-scan order) before Y_{ij} is called the causal neighbourhood ($\boldsymbol{M}_{ij}$) of that pixel (here, the size of $\boldsymbol{M}_{ij}$ is set to $w = 3$). The total number of pixels in $\boldsymbol{M}_{ij}$ is $\frac{(2w+1)^2-1}{2}$. The response pixel and those within $\boldsymbol{M}_{ij}$ are colour-coded as, respectively, black and blue. The red and blue pixels are indexed (in a raster-scan order) before Y_{ij}, whereas the yellow pixels are indexed after Y_{ij} (the colours do not represent phase values).

approximates the conditional distributions (and hence $f(\boldsymbol{X})$) on the right-hand side, we can reconstruct $\boldsymbol{Y}$ pixel-by-pixel by sampling from that model (in the 'Reconstruction in 3D' section we elaborate on how the various conditional probabilities on the right-hand side in Eq. (1) can be estimated with only a single model). To facilitate the estimation of $f(\boldsymbol{X})$ (or to make the approximation of $f(\boldsymbol{X})$ tractable), we assume $\boldsymbol{X}$ can be modelled as a form of stationary MRF with the following properties:

- Locality: $f(X_{ij}|\boldsymbol{X}^{(<ij)}) = f(X_{ij}|\boldsymbol{M}_{ij})$ for a sufficiently large (causal) neighbourhood $\boldsymbol{M}_{ij}$.
- Stationarity: $f(X_{ij}|\boldsymbol{M}_{ij})$ does not depend on pixel location (i, j).

In the above, $\boldsymbol{M}_{ij}$ is a sufficiently large group of pixels surrounding one side of X_{ij} (see Fig. 1) and the conditional probability $f(X_{ij}|\boldsymbol{M}_{ij})$ has a Bernoulli distribution with an event probability (the event being defined as $X_{ij} = 1$) that depends on the pixel values in $\boldsymbol{M}_{ij}$. $\boldsymbol{M}_{ij}$ is called the *causal* (Burt & Adelson, 1983) neighbourhood of X_{ij}, because it only includes a subset of the pixels in $\boldsymbol{X}^{(<ij)}$, i.e., only pixels that are before X_{ij} in a raster-scan order (see the 'Reconstruction in 3D' section for more discussions).

Intuitively, the above assumptions mean that given $\boldsymbol{M}_{ij}$, there is no additional information in $\boldsymbol{X}^{(<ij)}$ that could further improve the predictability of X_{ij}. With these assumptions, $\boldsymbol{X}$ can be converted into a training dataset in which the phase of each of its pixels is represented as a function of the phase values of some of its surrounding pixels (see Fig. 2). Once the training dataset is built, any off-the-shelf supervised learner can be used to fit a model to it that represents the conditional distribution $f(X_{ij}|\boldsymbol{M}_{ij})$. Making use of the decomposition (1) with the MRF property, this model can subsequently be used to reconstruct an image of arbitrary size pixel-by-pixel in a raster-scan order.

Proposed extensions of the approach

The basic procedures of the extended approach are summarized in Figure 3 and will be described throughout this section. We first elaborate on the reconstruction details (the 'Reconstruction in 3D' section) and then explain how user-defined predictors can be incorporated into the model (the 'User-defined predictors' section). We conclude this section by introducing the reduced model (the 'Reduced model' section)

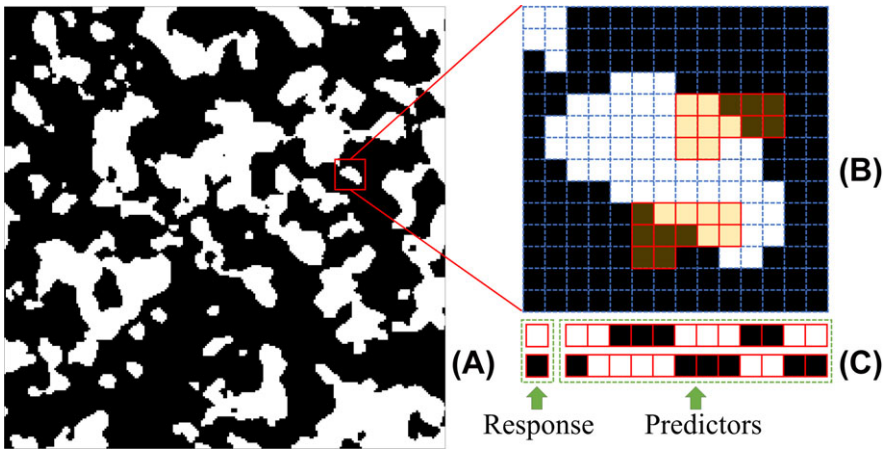

Fig. 2. (Colour online) (A) The original binary image **X**. The white and black colours represent phase values. (B) The magnified view of a small portion of **X** in which the pixels are separated with dashed blue lines. The highlighted regions (pixels within red boxes) represent two instances of a causal neighbourhood with size $w = 2$. (C) Rearrangement of the pixels in the highlighted regions in (B) into row vectors for building a training dataset. This procedure is repeated for all the pixels in **X** to convert it into a training dataset.

© 2016 The Authors

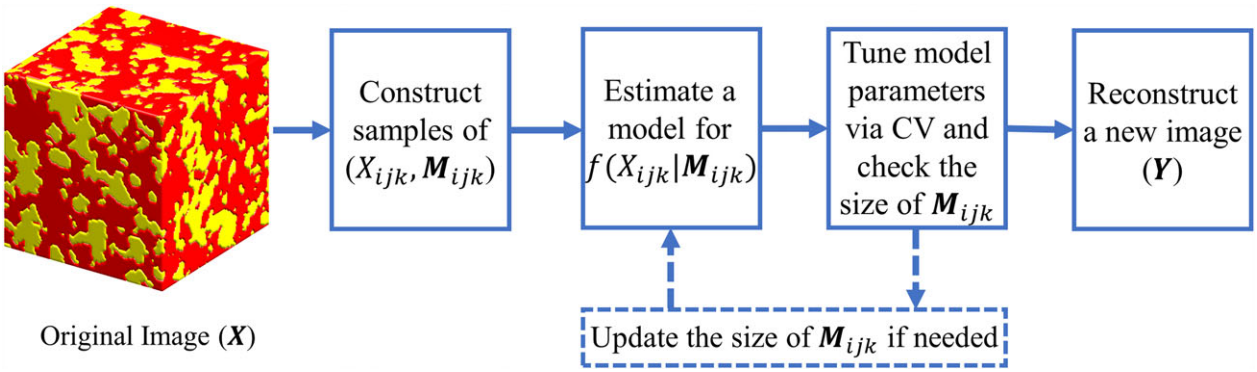

Fig. 3. Flowchart of the supervised learning approach for microstructure C&R.

and addressing potential boundary issues (the 'Boundary effects and initialization' section).

Reconstruction in 3D

Bostanabad *et al.* (2016) developed two methods for reconstruction in 2D; a noncausal approach based on Gibbs sampling (Casella & George, 1992) and a causal one. As the former approach is iterative and hence computationally prohibitive in 3D, we only extend and use the causal approach in this paper.

We reconstruct an image in 3D slice-by-slice; with a raster-scan order within each slice. This way the shape of $\boldsymbol{M}$ in 3D needs to be similar to half a cube (see Fig. 4) to only include the already reconstructed surrounding voxels. To justify this, we write the joint distribution $f(\boldsymbol{X})$ in the conditional form as:

$$\begin{aligned} f(\boldsymbol{X}) &= f(X_{111})\, f(X_{121}|X_{111})\, f(X_{131}|X_{111}, X_{121}) \cdots \\ &\quad f(X_{n_1 n_2 n_3}|X_{111}, X_{121}, \ldots, X_{n_1(n_2-1)n_3}) \\ &= f\left(X_{111}|\boldsymbol{X}^{(<111)}\right) f\left(X_{121}|\boldsymbol{X}^{(<121)}\right) \\ &\quad f\left(X_{131}|\boldsymbol{X}^{(<131)}\right) \cdots f\left(X_{n_1 n_2 n_3}|\boldsymbol{X}^{(<n_1 n_2 n_3)}\right), \end{aligned} \quad (2)$$

where $\boldsymbol{X}^{(<ijk)}$ denotes the set of all the voxels in $\boldsymbol{X}$ ordered *before* X_{ijk}. Equation (2) holds for both the original image and the one being reconstructed. It illustrates that if we use $\boldsymbol{X}$ to learn a model that can predict the phase value at any location (i, j, k) given the phase values of voxels located *before* it, we can use that model to predict the phase value at any location (i, j, k) in $\boldsymbol{Y}$ where all the voxel phases before (i, j, k) have already been predicted.

With the stationary MRF assumptions, this model can be simplified to $f(X_{ijk}|\boldsymbol{X}^{(<ijk)}) = f(X_{ijk}|\boldsymbol{M}_{ijk})$ with the same conditional probability function $f(X|\boldsymbol{M})$ (independent of voxel index (ijk)) for every voxel in the image. To learn *one single* model in characterization that we can use for predicting the Bernoulli phase probability value at *any* voxel location during reconstruction (and then generating a voxel phase value from the predicted Bernoulli probability), we must fix the size of $\boldsymbol{M}$. Otherwise, the number of neighbouring voxels in the conditional distributions in Eq. (2) would change and one would need to learn multiple models (each corresponding to an $\boldsymbol{M}$ with a different size) and use them in reconstruction accordingly (for instance, at X_{121} and X_{131} the voxels which are already reconstructed are, respectively, $\boldsymbol{M}_{121} = \{X_{111}\}$ and $\boldsymbol{M}_{131} = \{X_{111}, X_{121}\}$, and the corresponding size of $\boldsymbol{M}_{ijk}$ is 1 and 2). For a fixed neighbourhood size, three questions must be answered: (1) what is an appropriate size (w) for $\boldsymbol{M}$, (2) how to sample from the boundary voxels in $\boldsymbol{X}$ and (3) how to reconstruct the boundary voxels in $\boldsymbol{Y}$.

As discussed in Bostanabad *et al.* (2016), the optimum neighbourhood size is typically on the order of the largest topological feature in $\boldsymbol{X}$ and can be determined in a data-driven manner via cross-validation (CV) by starting with a relatively large neighbourhood and shrinking the size down until the CV error is minimized. An attractive feature of this approach is that, provided that $\boldsymbol{M}$ is not too small, the results are not overly sensitive to the size of $\boldsymbol{M}$: If the chosen size is larger than the optimum one, the supervised learner will automatically exclude the unimportant voxels (predictors) from $\boldsymbol{M}$. If, however, the chosen size is somewhat smaller than the optimum value, the supervised learner can partially compensate for this by building a more complex model. Because we use classification trees as the supervised learner, complexity refers to the number of tree leaves and not the number of predictors (which equals the number of voxels in $\boldsymbol{M}$) of the tree.

Regarding the second question, we simply exclude the boundary voxels of $\boldsymbol{X}$ when building the training dataset (see the 'User-defined predictors' section) because the size of $\boldsymbol{X}$ is significantly larger than $\boldsymbol{M}$ and hence little (if any) information will be lost. In reconstruction, however, the predictions close to the boundaries (where $\boldsymbol{M}$ does not fall entirely in $\boldsymbol{Y}$) will be inaccurate. To mitigate this issue, we reconstruct a larger image than required and discard the boundaries (see the 'Boundary effects and initialization' section).

After the model (with an $\boldsymbol{M}$ of size w, see Fig. 4) is learned, reconstructing an image of arbitrary size $s_1 \times s_2 \times s_3$ is achieved via the steps in the following pseudo-code (due to causal

© 2016 The Authors

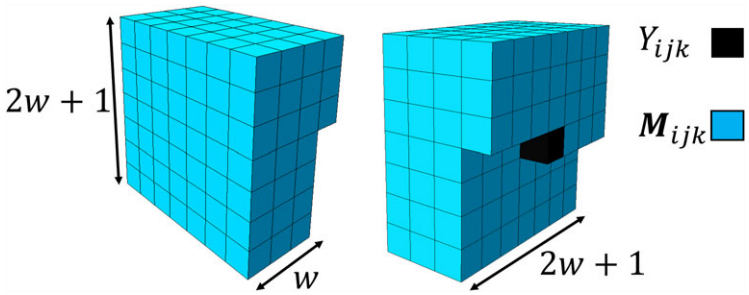

Fig. 4. (Colour online) A full 3D causal neighbourhood of voxel Y_{ijk} with size $w = 3$ from two different perspectives (the voxels outside of $\boldsymbol{M}_{ijk}$ are not shown). The total number of voxels in $\boldsymbol{M}_{ijk}$ is $\frac{(2w+1)^3-1}{2}$. The response voxel and those within $\boldsymbol{M}_{ijk}$ are colour-coded as, respectively, black and blue (the colours do not represent phase values).

reconstruction, only the boundaries of the initial $\boldsymbol{Y}$ are needed in Step 1):

(1) Start with an initial image $\boldsymbol{Y}$ of size $(m_1 + 2w) \times (m_2 + 2w) \times (m_3 + w)$, where $m_1 > s_1, m_2 > s_2$ and $m_3 > s_3$ (see the 'Boundary effects and initialization' section).
(2) For $k = w + 1, w + 2, \ldots, w + m_3$ (slice)
For $i = w + 1, w + 2, \ldots, w + m_1$ (row)
For $j = w + 1, w + 2, \ldots, w + m_2$ (column)
   (a) Use the fitted supervised learning model to predict the Bernoulli parameter $p_{ijk} = f(Y_{ijk}|\mathbf{M}_{ijk})$ and generate $Y_{ijk} \sim$ Beroulli(p_{ijk}).
   (b) Use the newly generated Y_{ijk} to update the corresponding voxel in $\boldsymbol{Y}$.
(3) Pick the central part of $\boldsymbol{Y}$ with size $s_1 \times s_2 \times s_3$ as the new image.

User-defined predictors

The performance of the developed approach can be enhanced via user-defined predictors. By 'user-defined predictor', we mean some combination of the voxels within each causal neighbourhood into a single variable that will typically represent some physically meaningful descriptor of the neighbourhood and that will be included as an additional predictor variable in the supervised learning model. The purpose is to improve the prediction of the individual response voxel associated with each neighbourhood. User-defined predictors can be used to effectively capture a particular morphological characteristic and are general in that they can be applied to 2D/3D microstructures of any kind (e.g. isotropic and anisotropic). An appealing feature of the approach is that if the user-defined predictor does not increase the predictive power of the model, the supervised leaner will automatically exclude it from the final fitted model at the end of the learning stage. We have tried a myriad of user-defined predictors, and in this paper we demonstrate the use of the ones that consistently improved the performance of the fitted model. We emphasize that (1) all the user-defined predictors are incorporated directly in the supervised learning stage in a data-driven manner (i.e. no tuning is required), and (2) how we use these predictors is completely different than how physical descriptors are used in optimization-based approaches (reviewed in the 'Introduction' section). We elaborate more on the second point at the end of this section.

Below, we first elaborate on how the model is learned and then explain how the user-defined predictors can be incorporated into it.

Supervised learner. Having chosen the size of $\boldsymbol{M}$, the next step is to build the training dataset by (1) placing $\boldsymbol{M}$ on all the voxels (excluding the boundary ones) of $\boldsymbol{X}$, (2) recording the phase values observed at each voxel in the neighbourhood and (3) rearranging the order of the stored values into a row vector (see Fig. 2 for a 2D illustration). This way, a 2D array of data, denoted by D, with $R = (n_1 - 2w) \times (n_2 - 2w) \times (n_3 - w)$ rows and $C = 1 + \frac{(2w+1)^3-1}{2}$ columns will be built from the collection of voxels in the original $n_1 \times n_2 \times n_3$ image $\boldsymbol{X}$. In each row of D, the first column stores the 'response' variable X_{ijk} and the remaining columns store the corresponding 'predictor' variables in $\boldsymbol{M}_{ijk}$.

Once D is built, any off-the-shelf supervised learning algorithm can be used to automatically fit a model to it. Our studies indicate that a nonparametric classification tree (Breiman *et al.*, 1984; Hastie *et al.*, 2009) is particularly well suited to model $f(X_{ijk}|\boldsymbol{M}_{ijk})$. Trees are especially suited for handling categorical variables, and our response and predictor variables are as such. Trees are also highly interpretable and, as shown in the 'Results and discussions' section, very computationally efficient to either fit or make predictions with.

Construction of classification trees from training data dates back to the 1960s (Ripley, 1996). In a classification tree (see Fig. 5), the root node is located at the top level, leaf nodes are located at the bottom level and interior nodes are in between the root and the leaves. Given a fitted classification tree (fitting is automatically done in commercial software such as

© 2016 The Authors

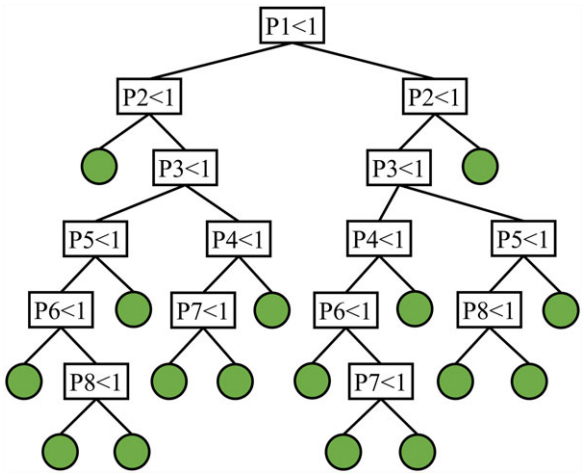

Fig. 5. (Colour online) Pedagogical example of a classification tree model with eight predictors. The root node is located at the top and the leaves (green circles) at the bottom. The predictors are denoted by P1 thorough P8 and each nonleaf node represents a splitting rule on the corresponding predictor: when traversing the tree down, if a predictor's value is smaller than the splitting value (here 1) its left branch is chosen. Otherwise, the right branch is chosen.

Python, Pedregosa *et al.*, 2011), the response probability p_{ijk} (Step 2.a in the pseudo-code) of a future observation is predicted by traversing the tree down following the sequences of splits from the root node to the correct leaf node and assigning the stored probability in that leaf node to the observation. The class probability in a leaf is learned in the fitting process via dividing the number of training observations with response values in that class by the total number of training observations which fell into that leaf. In our application, each training observation corresponds to one nonboundary voxel (say voxel (i, j, k)) in $\boldsymbol{X}$, and X_{ijk} and the corresponding $\boldsymbol{M}_{ijk}$ are the response variable and the set of predictor variables, respectively, for that observation.[3]

Classification tree algorithms, like any other supervised learning algorithm, have some tuning parameters that can be adjusted to increase the predictive power. An advantage of our approach is that we can simply use CV to select all the tuning parameters of the supervised learning algorithm in order to best approximate $f(X_{ijk}|\boldsymbol{M}_{ijk})$ and do not rely on other methods (e.g. correlation functions) to determine any of the parameters (i.e. the neighbourhood size, w).

Incorporation of user-defined predictors into the model. Potential model inaccuracies, coupled with the boundary effects, may sometimes result in reconstructed images having different VFs than the original image. Bostanabad *et al.* (2016) addressed this issue by introducing an *ad hoc* offset parameter that empirically adjusts the conditional probabilities of the fitted tree model in an attempt to better match the VFs of the original and reconstructed images. As the adjustment is done iteratively, this method might be unappealing in 3D since the reconstruction cost is in the order of a few seconds (as opposed to 2D where the cost is less than a second). Moreover, it alters the supervised learning model in that it no longer represents exactly what was learned *purely* from the data. In this paper, to closely match the VF of the reconstructed samples with that of $\boldsymbol{X}$ in 3D, we define a new predictor denoted by P_{VF} which captures the local VF in $\boldsymbol{X}$. As opposed to other predictors in the model (which are binary and correspond to the individual voxels in $\boldsymbol{M}_{ijk}$), $P_{VF_{ijk}}$ is defined as the average of all the phase values in $\boldsymbol{M}_{ijk}$ at X_{ijk} and $0 \leq P_{VF_{ijk}} \leq 1$. When fitting the supervised learning model, the additional predictor P_{VF} is included as an additional column in the training dataset D. During reconstruction, for predicting the phase value at Y_{ijk}, the set of predictor voxels in the neighbourhood $\boldsymbol{M}_{ijk}$ of Y_{ijk} is also augmented with $P_{VF_{ijk}}$ before passing it to the fitted classification tree. We note that the same supervised learning software can be applied whether or not P_{VF} is included as an additional predictor, because tree fitting algorithms can handle continuous variables. Although the use of a continuous predictor in the training dataset might slightly increase the fitting cost, it eliminates the iterative approach for matching the VFs in the reconstruction step.

Our studies show that sometimes the phase connectivity in the reconstructed images is slightly less than that in the original image if only VF is used as an additional predictor. To address this issue, we consider three categorical user-defined predictors (denoted by P_{C_i}, $i = 1, 2, 3$), which capture the phase connectivity of the response voxel in three orthogonal directions. In particular, because for a two-phase microstructure (i.e. phase values are either 0 or 1) the probability that $X_{ijk} = 1$ is estimated, we define the P_{C_i}'s as (w is the size of $\boldsymbol{M}$):

$$\begin{cases} P_{C\,1} = 1 \text{ if } X_{(i-1)jk} = X_{(i-2)jk} = \cdots = X_{(i-w)jk} = 1 \\ P_{C\,1} = 0 \qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise} \end{cases}, \tag{3.1}$$

$$\begin{cases} P_{C\,2} = 1 \text{ if } X_{i(j-1)k} = X_{i(j-2)k} = \cdots = X_{i(j-w)k} = 1 \\ P_{C\,2} = 0 \qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise} \end{cases}, \tag{3.2}$$

$$\begin{cases} P_{C\,3} = 1 \text{ if } X_{ij(k-1)} = X_{ij(k-2)} = \cdots = X_{ij(k-w)} = 1 \\ P_{C\,3} = 0 \qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise} \end{cases}. \tag{3.3}$$

Put another way, the P_{C_i}'s help the model to learn how likely the $X_{ijk} = 1$ event is if *all* the voxels to one side of X_{ijk} along the three principal direction lines are 1. Similar to P_{VF}, to incorporate these predictors into the model, the training dataset must be augmented with three more columns, and in reconstruction (for predicting the phase value at Y_{ijk}) the set of predictor voxels in $\boldsymbol{M}_{ijk}$ is augmented with the additional

[3] In commercial software, the set of predictors should be passed to the supervised learner as a row/column vector and hence the predictors in $\boldsymbol{M}_{ijk}$ need to be rearranged accordingly. See Figure 2(C).

© 2016 The Authors

user-defined predictors before passing the set to the fitted classification tree. As illustrated in the 'Results and discussions' section, the P_{C_i}'s notably improve the performance of the reduced model (introduced in the 'Reduced model' section) especially if the original microstructure is anisotropic.

One way to view our user-defined predictors is as physical descriptors that characterize each small neighbourhood (rather than the entire microstructure image). In fact, physically meaningful descriptors from the existing literature have inspired our choice of analogous user-defined predictors that we have tried in our method. This is how we came up with P_{VF} (inspired by VF) and P_{Ci} (inspired by lineal-path function). In prior work that has used physical descriptors, a single value (or distribution) is calculated for an entire microstructure image (e.g. a microstructure image with a VF of 20% or with a mean nearest-neighbour distance of 100 nm, etc.). This single value for the descriptor(s) serves as a summary characterization of the microstructure image and, in optimization-based approaches, is used as an optimization criterion for matching the reconstructed and original samples. In contrast, in our use of user-defined predictors, we do not have a single value assigned to them for the entire image. Rather, a separate value for the predictor is calculated for the small causal neighbourhood surrounding each voxel of the microstructure image (see Fig. 4 for an illustration of the neighbourhood). For example, in the case of P_{VF}, the neighbourhood surrounding each voxel has its own VF value calculated. This value is then used as an additional 'predictor variable' when predicting the phase value for that voxel.

Finally, we emphasize that to implement the approach with any user-defined predictors incorporated into the model, all that needs to be done is the appropriate augmentation of the training dataset. The supervised learner will automatically determine whether the defined predictor increases the predictive power of the model and if so, how it should be incorporated into the model.

Reduced model

Computational issues are always a concern in 3D C&R. In our case, a potential computational issue is the size of the training dataset (D) that needs to be stored until the model is learned. This issue will be particularly challenging if a large neighbourhood is required to learn the morphology of a relatively large microstructure (e.g. if $w = 10$, the number of predictors, including P_{VF} and P_{Ci}'s, would be 4634). We address this issue by changing the shape of the neighbourhood from full 3D to partial 3D (see Fig. 6 and compare it to Fig. 4). This neighbourhood consists of three orthogonal planes of voxels as well as four individual voxels. The individual voxels are located at the intersection corners of the planes. The reason that such reduction in the number of neighbouring voxels (which are not immediately next to the response voxel) bears negligible adverse effect on the performance is that once the number of predictors is reduced, the classification tree model partially compensates for it by finding appropriate rules,[4] which results into partitioning the predictor–variable space into smaller regions (i.e. the tree will have more leaves). In this paper, we use both the full 3D and the partial 3D neighbourhoods to characterize and reconstruct three 3D microstructures and illustrate that the results are comparable.

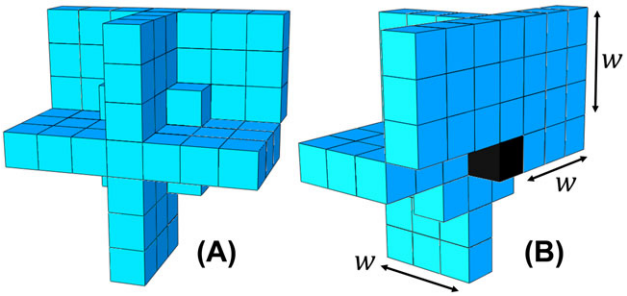

Fig. 6. (Colour online) A partial 3D neighbourhood of size $w = 3$ from two different perspectives. This neighbourhood has less voxels in it than a full 3D neighbourhood (Fig. 4). The total number of voxels in this neighbourhood is $3w(2w + 1) + 4$.

Boundary effects and initialization

As a single model is used to estimate the various conditional probabilities on the right-hand side of Eq. (2), inaccuracies will be introduced into the reconstructed image. That is, since at the boundary voxels of $\boldsymbol{Y}$ not all the predictors are available in $\boldsymbol{M}$, the predictions at those voxels will be less reliable. To address this issue, we note that this source of prediction error dies out as one moves away from the boundary voxels, and hence our solution is to reconstruct a larger image than required and choose the central part. Figure 7 illustrates this for a 3D structure (half of the structure to the right is cropped to reveal its interior). The blue region represents the reconstructed image (from the pseudo-code in the 'Reconstruction in 3D' section) and the green voxels are added to its exterior so as its boundary voxels would not have missing data in their neighbourhood. The thickness of the green region equals the neighbourhood size (w). Since the green region is not updated, the boundary voxels of the blue region may be adversely affected. This adverse effect will die out as we move to the interior of the blue region. Hence, we choose the central part of the blue region of size $s_1 \times s_2 \times s_3$ (a cross-section of which is shown with a black box in Fig. 7) as the final reconstructed image.

The differences $m_1 - s_1$, $m_2 - s_2$ and $m_3 - s_3$ depend on the initial image (the green region in Fig. 7). We investigated the performance of our approach with different initial images: (1) pure white/black, (2) pure random and (3) generated by splicing copies of the original 3D image side-by-side.

[4] The sequences of the splits (on the predictors) from the root node to a leaf, determine a rule in a classification tree.

© 2016 The Authors

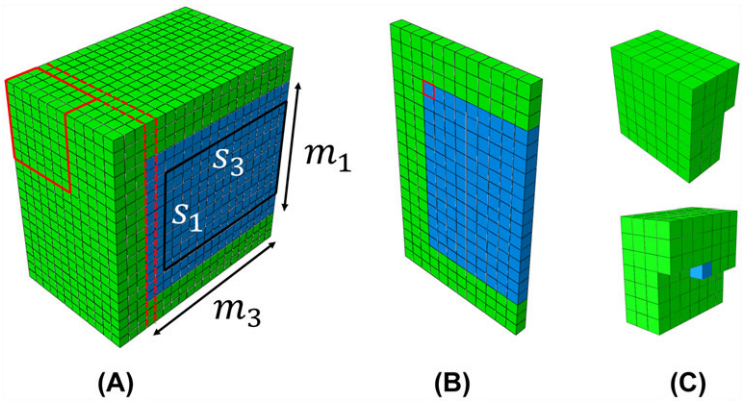

Fig. 7. (Colour online) To reduce the inaccuracies due to boundary effects, a larger image is reconstructed and the central part is chosen as the final image (the colours do not represent phase values). (A) The initial image of size $(m_1 + 2w) \times (m_2 + 2w) \times (m_3 + w)$. If the desired size of $\boldsymbol{Y}$ is $s_1 \times s_2 \times s_3$, a larger image of size $m_1 \times m_2 \times m_3$ is reconstructed. In this figure, $w = 3$ and only half of the structure is shown to reveal the interior. (B) The first 2D slice that will be reconstructed (its position in the 3D image is illustrated with dashed red lines). Y_{444} (the first voxel to be reconstructed) is coloured in red. (C) The 3D causal neighbourhood of Y_{444} from two different perspectives (the boundaries of the neighbourhood are coloured with solid red lines in A).

Our studies indicate that the latter produces better results and also demonstrates that the algorithm can learn the randomness of the microstructure and automatically inject it into the reconstructed image. In this case, the differences can be set to $(m_1 - s_1,\ m_2 - s_2,\ m_3 - s_3) \sim (2w, 2w, w)$.

Results and discussions

In this section, we apply the developed approach to three examples (EX1, EX2 and EX3) and evaluate the statistical equivalency of the secondary phase (inclusions or pores) between the original and reconstructed 3D images. The evaluations are based on four measures: the VF, the specific surface[5] (SS), the two-point correlation ($S_2(r)$) function and the lineal-path ($L(r)$) function. In case of $S_2(r)$ and $L(r)$, the errors are calculated via the following formula (which is equivalent to how the energy is defined in the optimization-based approaches):

$$\Delta f = \sum_r \left[f_{\text{original}}(r) - f_{\text{reconstructed}}(r) \right]^2,$$

where r is the length of the thrown line (see the Appendix), and f is either $S_2(r)$ or $L(r)$. We note that in all the examples (1) the size of the reconstructed samples is set to that of the original microstructure but this is not necessary, and (2) the statistical evaluations are not reported for the primary phase (the matrix) because the associated errors were relatively smaller than those reported herein for the secondary phase.

It should be emphasized that none of the aforementioned performance measures are used in the C&R process. In particular, all the model tuning parameters of the supervised learner as well as the neighbourhood size are solely determined via CV. Moreover, the CV accuracy measure is the standard supervised learning classification measure of predictive accuracy for individual voxels, and not some measure that involves the four performance measures discussed above.

[5] Area of the interphase per unit volume of the medium.

Clustered isotropic microstructure

Figure 8(A) illustrates the microstructure of interest in this example; a clustered isotropic nanocomposite with 9.27% silica (secondary phase) in a rubber matrix. The inclusions form small clusters and possess random geometry and irregular spatial distribution pattern.

Following the procedure outlined in the 'Proposed extensions of the approach' section, we fitted two causal models to the training image in Figure 8(A); one via a full 3D neighbourhood and one with a partial 3D neighbourhood. Each of these models was subsequently used to reconstruct a batch of 200 microstructure samples to assess the sample-to-sample variations. Figures 8(B) and (C) each illustrate a randomly chosen sample from each batch and indicate that both models capture the stochasticity of the original structure quite well. Details on the fitted parameters, computational costs and summary of statistical evaluations are given in Tables 1 and 2. In this example, the inclusions are quite small (no larger than 15 pixels on each side) and so we set the neighbourhood size (w) in the full and partial models to, respectively, 5 and 7. With these choices of w, almost all the predictors (including all the user-defined ones) are found to be important in both models, and the trees have many leaves (see the first two rows of the last column in Table 1). We repeated the above procedure with larger w's and achieved similar results in terms of statistical equivalency.

As the first two rows of Table 2 show, the total computational cost (the sum of Char. and Avg. rec. costs) is quite small, and this is of particular interest if a large number of samples

© 2016 The Authors

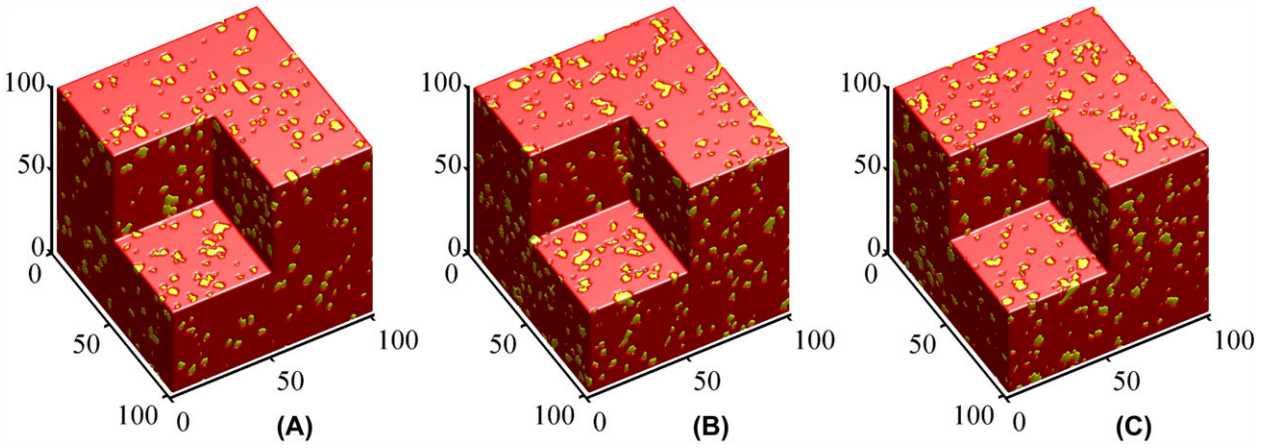

Fig. 8. (Colour online) EX1: (A) The original bi-phase training image with 9.27% silica in a rubber matrix, (B) a representative reconstructed image with a model trained using a full 3D neighbourhood and (C) a representative reconstructed image with a model trained using a partial 3D neighbourhood. The numbers are voxel indices and the two colours merely distinguish the two constituents. A small portion of all the images is cropped to reveal the interior of the structures.

Table 1. The fitted parameters and computational costs for all the examples. A full 3D neighbourhood is used in the full 3D models, whereas a partial 3D neighbourhood is used in the partial 3D models. The characterization (Char.) cost includes the total cost of fitting the tree. The average reconstruction (Avg. rec.) cost is calculated by averaging the reconstruction cost of all the samples in a batch (200 samples in each batch). The last two columns enumerate the number of leaves and predictors (out of all the initial predictors) in the fitted tree. See Figures 4 and 6 for the definition of window size.

Example	Window size	Char. cost (s)	Avg. rec. cost (s)	Leaf count	Retained predictors
1, Full 3D model	5	389.30	3.91	5002	658/669
1, Partial 3D model	7	114.47	2.87	4767	320/333
2, Full 3D model	6	374.72	5.88	7689	1098/1102
2, Partial 3D model	10	147.59	5.41	6223	636/638
3, Full 3D model	7	473.93	9.38	3791	1430/1691
3, Partial 3D model	12	162.52	7.96	3588	878/908

are to be reconstructed. We note that (1) although the neighbourhood size parameter w is larger in the partial model, the model with a full 3D neighbourhood has more voxels (predictors), and hence its fitting cost is higher (389.3 vs. 114.47 s), and (2) to arrive at similar results (in terms of performance measures), w must be larger in the partial 3D model.

In this example, we used each model to generate a batch of 200 samples, each sample comprising one reconstructed image. Each of the samples was compared to the original structure based on the aforementioned four measures and the results were averaged over the entire 200 reconstructed samples (the associated standard deviations were negligible and therefore are not reported). Table 2 summarizes the evaluations and indicates that (1) incorporation of the user-defined predictors into the model has made it possible to closely match the VF as well as the connectivity of the inclusions in the original and reconstructed structures (see TableA1 in the Appendix for evaluation results without these predictors); and (2) the algorithm can capture the spatial distribution and the specific surface of the inclusions quite well.

To visualize the variations in the performance measures across the samples, $S_2(r)$ and $L(r)$ of two of the representative samples from each batch are plotted against those of the original structure in Figure 9, and it can be observed that the statistical equivalency is well preserved for the whole range of correlation functions. In addition, the box plots in Figure 9(C) clearly shows that the variations in the specific surface is negligible for all the samples (reconstructed with either of the models) and the ensemble average of both batches is close to the specific surface of the original image. In Figure 9(C) the blue boxes mark the 25th and 75th percentiles, the red lines are the medians and the whiskers extend to include all the samples.

Porous medium

The porous medium of interest (Fig. 10A) is a ceramic microstructure with pores occupying 39.13% of the volume. Compared to the previous example, the VF and connectivity range of the secondary phase (the pores) is significantly higher and therefore it is anticipated that a more complex model is required for effective characterization. In porous materials, voids are the inhomogeneities and affect the solid–liquid interaction (e.g. porosity and specific surface of a catalyst are important for catalytic activity, see Debye *et al.*, 1957). The connectivity of pores in this structure spans up to 20 pixels.

As in the previous example, we fitted two causal models to the training image in Figure 10(A); one with a full 3D neighbourhood and one with a partial 3D neighbourhood

© 2016 The Authors

Table 2. Average errors in volume fraction ($\overline{|\Delta VF|}$), two-point correlation function ($\overline{\Delta S_2(r)}$), lineal-path function ($\overline{\Delta L(r)}$) and specific surface ($\overline{\Delta SS}$) between the original structure and the reconstructed samples. The results are averaged over all the samples in a batch.

Example, Model	$\overline{\lvert\Delta \boldsymbol{VF}\rvert}$	$\overline{\Delta \mathbf{S}_2(\boldsymbol{r})}$	$\overline{\Delta \mathbf{L}(\boldsymbol{r})}$	$\overline{\Delta \boldsymbol{SS}}$
1, Full 3D model	5.22×10^{-5}	3.29×10^{-6}	4.66×10^{-6}	2.18×10^{-5}
1, Partial 3D model	4.49×10^{-5}	6.75×10^{-6}	5.11×10^{-6}	2.80×10^{-5}
2, Full 3D model	1.10×10^{-4}	2.84×10^{-4}	5.48×10^{-4}	1.60×10^{-3}
2, Partial 3D model	1.37×10^{-4}	2.34×10^{-4}	5.51×10^{-4}	1.70×10^{-3}
3, Full 3D model	8.18×10^{-5}	2.59×10^{-4}	6.66×10^{-4}	3.10×10^{-3}
3, Partial 3D model	1.05×10^{-5}	4.31×10^{-4}	3.75×10^{-4}	6.28×10^{-3}

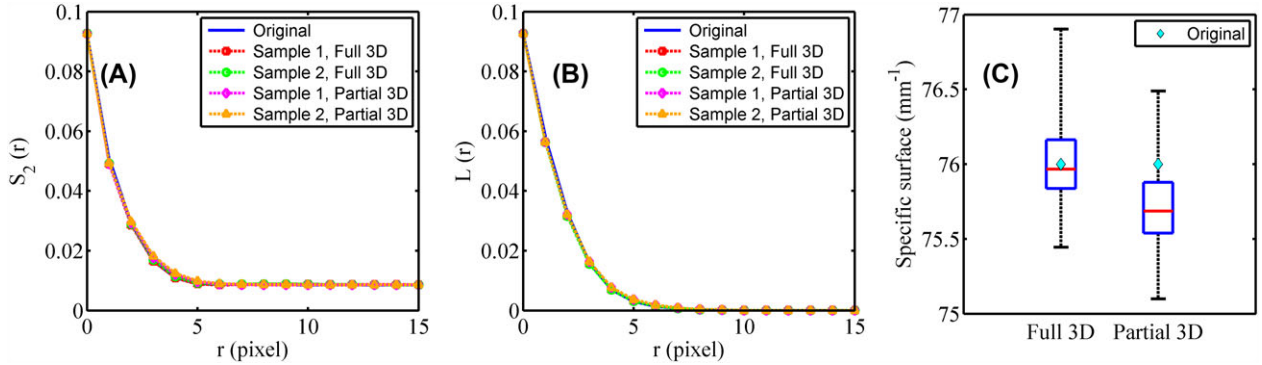

Fig. 9. (Colour online) (A) Two-point correlation and (B) lineal-path functions of the original and four of the reconstructed samples in EX1. Two of the samples are randomly chosen from the batch of the samples reconstructed with a model trained using a full 3D neighbourhood, whereas the other two are randomly chosen from the batch of the samples reconstructed with a model trained using a partial 3D neighbourhood. (C) Box plot of specific surface of all the reconstructed samples (200 samples for each model). The specific surface of the original structure is shown with a cyan diamond.

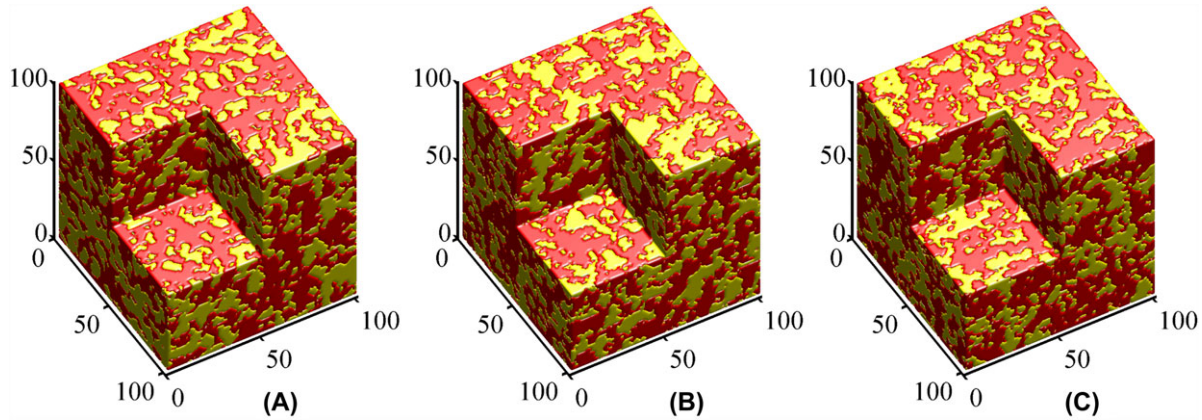

Fig. 10. (Colour online) EX2: (A) The original porous structure with 39.13% of pores, (B) a sample reconstructed image with a model trained using a full 3D neighbourhood and (C) a sample reconstructed image with a model trained using a partial 3D neighbourhood. The numbers are voxel indices and the two colours merely distinguish the two constituents. A small portion of all the images is cropped to reveal the interior of the structures.

and then used these models to reconstruct two batches of 200 microstructure samples. Figures 10(B) and (C) each illustrate a randomly chosen sample from each batch and indicate that both models provide visually appealing results. Contrary to EX1, we conjectured that a large neighbourhood is required to effectively learn the complex spatial voxel dependencies and therefore used a neighbourhood size of 15. However, it was observed that most of the predictors which were further away from the response voxel were not used in the model and hence we reduced the size to those reported in Table 1. With these choices of w, almost all the predictors (including the user-defined ones) are used in both models and the trees have many leaves (see the last two rows of the last column in Table 1).

© 2016 The Authors

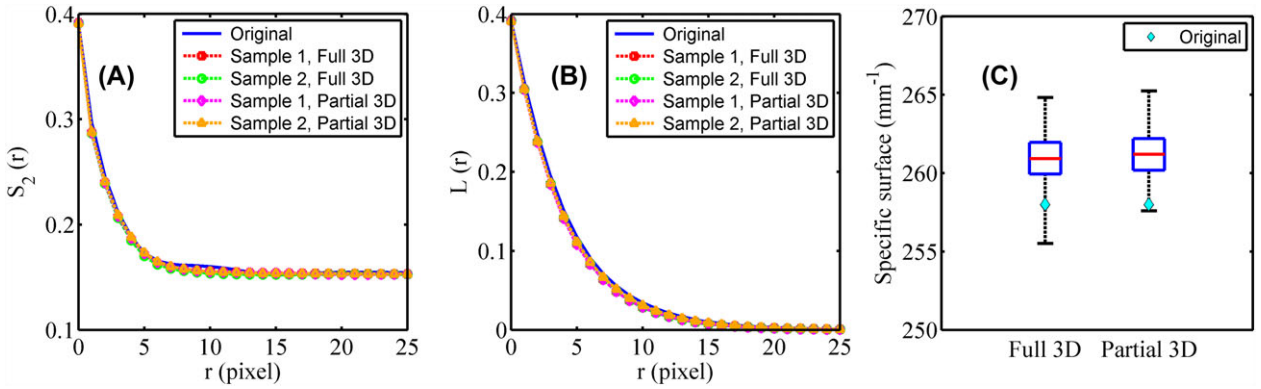

Fig. 11. (Colour online) (A) Two-point correlation and (B) lineal-path functions of the original and four of the reconstructed samples in EX2. See the caption of Figure 9 for more details. (C) Box plot of specific surface of all the reconstructed samples (200 samples for each model). The specific surface of the original structure is shown with a cyan diamond.

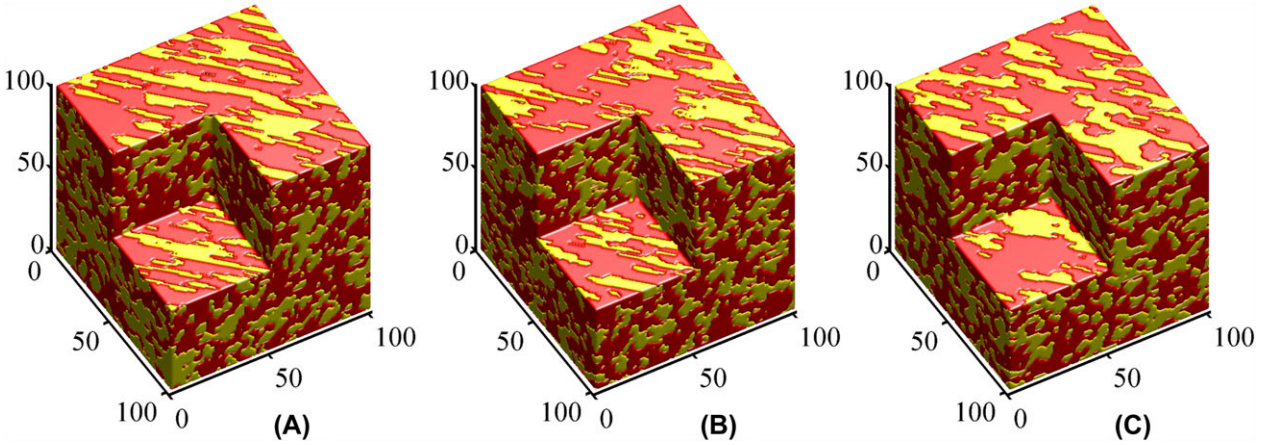

Fig. 12. (Colour online) EX3: (A) The original anisotropic structure with VF of 39.16%, (B) a sample reconstructed image with a model trained using a full 3D neighbourhood and (C) a sample reconstructed image with a model trained using a partial 3D neighbourhood. The numbers are voxel indices and the two colours merely distinguish the two constituents. A small portion of all the images is cropped to reveal the interior of the structures.

Table 1 demonstrates that the total computational cost is quite small for both models, with characterization cost of the full 3D model being higher than that of the partial 3D model as it has more predictors in its neighbourhood. Similar to EX1, to achieve comparable performance between the two models, a larger neighbourhood is required for the partial 3D model.

Two hundred microstructures were reconstructed with each model and all were compared with the original structure. Figures 11(A) and (B) illustrate sample results for two randomly chosen microstructures from each batch. It can be observed that the correlation functions match quite well and the general trends are captured. The average errors in performance measures for the samples in each batch are summarized in Table 2 (the associated standard deviations were negligible and therefore are not reported). It is evident that the average errors are quite small but have increased compared to EX1; which was expected since the original microstructure in Figure 10 is more complex than that in Figure 8. It can also be observed that the model fitted with a partial 3D neighbourhood performs no worse than the one fitted with a full 3D neighbourhood. This finding is of particular interest if one intends to characterize a high resolution microstructure and cannot afford fitting a model with a full 3D neighbourhood. Finally, Figure 11(C) illustrates that the variations in the specific surface is quite small across the samples (reconstructed with either of the models) and the ensemble average of both batches is close to the specific surface of the original image.

Anisotropic structure

The previous two examples were isotropic, and so they might not have fully demonstrated the potentials of the reduced model and the user-defined predictors. In this section, we test the algorithm on the anisotropic structure illustrated in Figure 12(A). It can be observed that the connectivity of the secondary phase is well beyond 20 pixels in this microstructure and the direction of anisotropy is in none of the orthogonal planes.

© 2016 The Authors

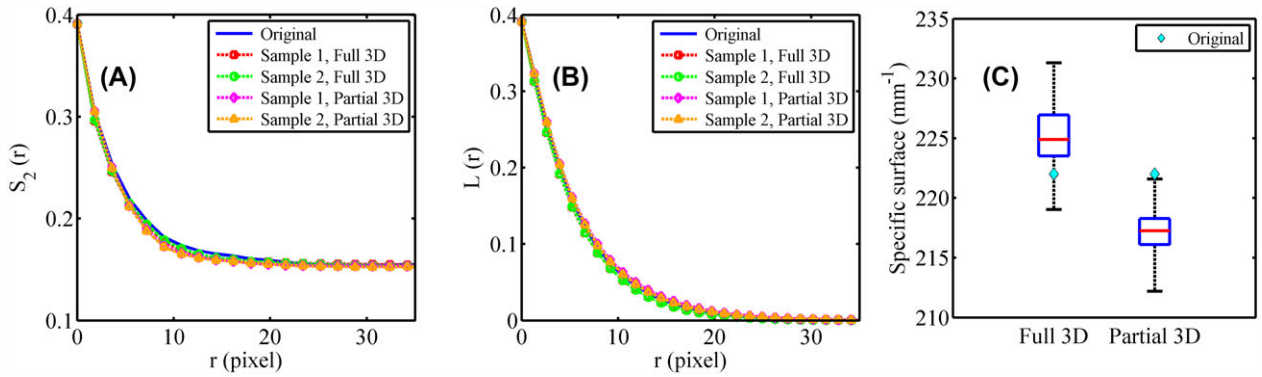

Fig. 13. (Colour online) (A) Two-point correlation and (B) lineal-path functions of the original and four of the reconstructed samples in EX3. (C) Box plot of specific surface of all the reconstructed samples (200 samples for each model). See the caption of Figure 9 for more details.

We followed the same procedure as in the previous two examples, fitting a full and a partial model to the original sample and subsequently using each of them to reconstruct a batch of 200 samples (see Tables 1 and 2 for simulation details and performance evaluations). Figure 12(B) and (C) illustrate sample results for two randomly chosen microstructures from each batch. Similar to EX1 and EX2, it is observed that (1) the correlations, specific surface and phase connectivity are quite well preserved (see Figure 13) across the reconstructed samples, (2) the partial model requires a slightly larger neighbourhood to adequately learn the morphology and (3) almost all the predictors (including all the user-defined ones) were kept in the model.

We note that the use of the P_{Ci}'s has improved the partial 3D model more in this example as compared to the previous two examples (compare the performance measures in Tables 2 and A1), and this highlights the importance of incorporating physics-based predictors into the supervised learning model.

Conclusion and future work

The primary goal of this paper is to extend the recently developed approach in Bostanabad *et al.* (2016) on supervised learning-based C&R in three major directions: extension to 3D, incorporation of user-defined predictors into the supervised learner and addressing computational issues with a reduced model. The method begins by converting the original 3D microstructure image into a set of training data, to which a supervised learning model is fitted. This relatively compact model learns the inherent stochasticity of the structure and is used to reconstruct any number of statistically equivalent 3D microstructure samples.

In the 'Reconstruction in 3D' section, we argued that, under the MRF assumption, our model provides an implicit representation of the full joint distribution $f(\boldsymbol{X})$ of the phase values of all the voxels in the microstructure image. In theory, $f(\boldsymbol{X})$ provides the most generic representation possible of the microstructure nature. In practice, the quality of this implicit representation depends on the ability of the supervised learning model to capture the conditional distributions $f(X_{ij}|\boldsymbol{M}_{ij})$, as well as on the validity of the stationary MRF assumptions. In the 'Results and discussions' section, we demonstrated that the fitted classification tree did a reasonable job in learning the stochastic microstructure behaviour in the examples, and the reconstruction algorithm did a reasonable job in preserving it in the reconstructed images.

The tree-fitting cost depends on the size (rows and columns) of the training dataset (D): the number of rows of D depends on the size of the original image, whereas the number of its columns depends on the number of predictors (the number of voxels in the neighbourhood plus the number of user-defined predictors). For general supervised learning applications, trees are widely regarded as being computationally very efficient, and our studies show that for a given training image (fixed number of rows of D), the fitting cost increases linearly with the total number of predictors in the neighbourhood. In other words, the cost is quite manageable even for complex microstructures which (potentially) require a large neighbourhood for effective learning. Our investigations indicate that the aforementioned linear computational complexity does not depend on the microstructure image. In addition, given a fitted model, the reconstruction cost increases linearly with the number of voxels in the reconstructed image, and hence the reconstruction cost is also quite manageable for building an ensemble of large microstructure images.

A major challenge in microstructure C&R is to build, with isotropy and homogeneity assumptions, a 3D sample via a single 2D image. The extension of our approach to tackle this challenge is nontrivial as the joint probability in 3D cannot be easily inferred from the one in 2D. However, we showed in this paper that a partial 3D neighbourhood with only three orthogonal planes can do a reasonable job in both learning and reconstruction. We are now working to use this finding and integrate it with ensemble/voting methods of supervised learning models to learn a 2D image but reconstruct a 3D one, for applications in which 3D images are difficult to obtain.

© 2016 The Authors

Acknowledgements

The authors are grateful to the anonymous referees for the insightful comments that have helped to improve the paper. This work is supported by the U.S. Department of Commerce, National Institute of Standards and Technology as part of the Center for Hierarchical Materials Design (CHiMaD) award 70NANB14H012, National Science Foundation Award No. CMMI-1265709 and the Air Force Office of Scientific Research (AFOSR) Award No. FA9550-12-1-0458.

References

Baghgar, M., Barnes, A.M., Pentzer, E., Wise, A.J., Hammer, B.A., Emrick, T., Dinsmore, A.D. & Barnes, M.D. (2014) Morphology-dependent electronic properties in cross-linked (P3HT-b-P3MT) block copolymer nanostructures. *ACS Nano*. **8**, 8344–8349.

Ballani, F. & Stoyan, D. (2015) Reconstruction of random heterogeneous media. *J. Micros*. **258**, 173–178.

Bauchy, M., Qomi, M.A., Ulm, F.-J. & Pellenq, R.-M. (2014) Order and disorder in calcium–silicate–hydrate. *J Chem. Phys*. **140**, 214503-1–214503-9.

Beran, M. (1965) Statistical continuum theories. *Trans. Soc Rheol. (1957-1977)* **9**, 339–355.

Berryman, J.G. (1985) Measurement of spatial correlation functions using image processing techniques. *J. Appl. Phys.* **57**, 2374–2384.

Berryman, J.G. & Blair, S.C. (1986) Use of digital image analysis to estimate fluid permeability of porous materials: application of two-point correlation functions. *J. Appl. Phys.* **60**, 1930–1938.

Bostanabad, R., Bui, A.T., Xie, W., Apley, D.W. & Chen, W. (2016) Stochastic microstructure characterization and reconstruction via supervised learning. *Acta Materialia*. **103**, 89–102.

Breiman, L., Friedman, J., Stone, C.J. & Olshen, R.A. (1984) *Classification and Regression Trees*. CRC Press, Chapman & Hall/CRC. USA.

Breneman, C.M., Brinson, L.C., Schadler, L.S., *et al.* (2013) Stalking the materials genome: a data-driven approach to the virtual design of nanostructured polymers. *Adv. Function. Mat*. **23**, 5746–5752.

Burt, P.J. & Adelson, E.H. (1983) The Laplacian pyramid as a compact image code. *IEEE Trans. Commun.* **31**, 532–540.

Caers, J. & Journel, A.G. (1998) Stochastic reservoir simulation using neural networks trained on outcrop data. In: *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, New Orleans, Louisiana, USA.

Casella, G. & George, E.I. (1992) Explaining the Gibbs sampler. *Am. Stat*. **46**, 167–174.

Chen, D., Teng, Q., He, X., Xu, Z. & Li, Z. (2014) Stable-phase method for hierarchical annealing in the reconstruction of porous media images. *Phys. Rev. E*. **89**, 013305-1–013305-10.

Corson, P.B. (1974a) Correlation functions for predicting properties of heterogeneous materials. I. Experimental measurement of spatial correlation functions in multiphase solids. *J. Appl. Phys.* **45**, 3159–3164.

Corson, P.B. (1974b) Correlation functions for predicting properties of heterogeneous materials. II. Empirical construction of spatial correlation functions for two-phase solids. *J. Appl. Phys.* **45**, 3165–3170.

Debye, P., Anderson, Jr, H. & Brumberger, H. (1957) Scattering by an inhomogeneous solid. II. The correlation function and its application. *J. Appl. Phys.* **28**, 679–683.

Debye, P. & Bueche, A. (1949) Scattering by an inhomogeneous solid. *J. Appl. Phys.* **20**, 518–525.

Edelman, R.R. & Warach, S. (1993) Magnetic resonance imaging. *N. Eng. J. Med.* **328**, 708–716.

Efros, A.A. & Freeman, W.T. (2001) Image quilting for texture synthesis and transfer. In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. ACM, 341–346.

Efros, A.A. & Leung, T.K. (1999) Texture synthesis by non-parametric sampling. In: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. IEEE, USA.

Elfeki, A. & Dekking, M. (2001) A Markov chain model for subsurface characterization: theory and applications. *Math. Geol.* **33**, 569–589.

Fullwood, D., Kalidindi, S., Niezgoda, S., Fast, A. & Hampson, N. (2008a) Gradient-based microstructure reconstructions from distributions using fast Fourier transforms. *Mat. Sci. Eng. A*. **494**, 68–72.

Fullwood, D.T., Niezgoda, S.R., Adams, B.L. & Kalidindi, S.R. (2010) Microstructure sensitive design for performance optimization. *Progress Mat. Sci.* **55**, 477–562.

Fullwood, D.T., Niezgoda, S.R. & Kalidindi, S.R. (2008b) Microstructure reconstructions from 2-point statistics using phase-recovery algorithms. *Acta Materialia*. **56**, 942–948.

Greene, M.S., Liu, Y., Chen, W. & Liu, W.K. (2011) Computational uncertainty analysis in multiresolution materials via stochastic constitutive theory. *Comput. Methods Appl. Mech. Eng.* **200**, 309–325.

Grigoriu, M. (2003) Random field models for two-phase microstructures. *J. Appl. Phys.* **94**, 3762–3770.

Guardiano, F.B. & Srivastava, R.M. (1993) Multivariate geostatistics: beyond bivariate moments. In: *Geostatistics Troia'92*. Springer, CA, USA.

Guo, E.-Y., Chawla, N., Jing, T., Torquato, S. & Jiao, Y. (2014) Accurate modeling and reconstruction of three-dimensional percolating filamentary microstructures from two-dimensional micrographs via dilation-erosion method, *Materials Characterization*. **89**, 33–42.

Hajizadeh, A., Safekordi, A. & Farhadpour, F.A. (2011) A multiple-point statistics algorithm for 3D pore space reconstruction from 2D images. *Adv. Water Res*. **34**, 1256–1267.

Harrison, P. (2001) A non-hierarchical procedure for re-synthesis of complex textures. The 9-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision. University of West Bohemia, Campus Bory, Plzen, Czech Republic, 190–197.

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J. & Tibshirani, R. (2009) *The Elements of Statistical Learning*. Springer, USA.

Holotescu, S. & Stoian, F. (2011) Prediction of particle size distribution effects on thermal conductivity of particulate composites. *Materialwissenschaft Und Werkstofftechnik*. **42**, 379–385.

Jiang, Z., Chen, W. & Burkhart, C. (2013) Efficient 3D porous microstructure reconstruction via Gaussian random field and hybrid optimization. *J. Microsc*. **252**, 135–148.

Jiao, Y., Stillinger, F. & Torquato, S. (2007) Modeling heterogeneous materials via two-point correlation functions: basic principles. *Phys. Rev. E*. **76**, 031110-1–031110-15.

Jiao, Y., Stillinger, F. & Torquato, S. (2008) Modeling heterogeneous materials via two-point correlation functions. II. Algorithmic details and applications. *Phys. Rev. E*. **77**, 031135-1–031135-15.

Jiao, Y., Stillinger, F. & Torquato, S. (2009) A superior descriptor of random textures and its predictive capacity. *P. Natl. Acad. Sci. U.S.A*. **106**, 17634–17639.

Karasek, L. & Sumita, M. (1996) Characterization of dispersion state of filler and polymer-filler interactions in rubber-carbon black composites. *J. Mat. Sci*. **31**, 281–289.

© 2016 The Authors

Kastner, J., Plank, B., Reh, A., Salaberger, D. & Heinzl, C. (2012) Advanced X-ray tomographic methods for quantitative characterisation of carbon fibre reinforced polymers. In: *Proceedings of the 4th International Symposium on NDT in Aerospace*. Augsburg, Deutschland.

Levitz, P. (1998) Off-lattice reconstruction of porous media: critical evaluation, geometrical confinement and molecular transport. *Adv. Colloid. Interfacac.* **76**, 71–106.

Li, D., Tschopp, M., Khaleel, M. & Sun, X. (2012) Comparison of reconstructed spatial microstructure images using different statistical descriptors. *Comput. Mat. Sci.* **51**, 437–444.

Liu, X. & Shapiro, V. (2015) Random heterogeneous materials via texture synthesis. *Comput. Mat. Sci.* **99**, 177–189.

Liu, Y., Steven Greene, M., Chen, W., Dikin, D.A. & Liu, W.K. (2013) Computational microstructure characterization and reconstruction for stochastic multiscale material design. *Computer-Aided Design*. **45**, 65–76.

March, W.B., Czechowski, K., Dukhan, M., *et al.* (2012) Optimizing the computation of n-point correlations on large-scale astronomical data. In: *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. IEEE Computer Society Press, USA.

Niezgoda, S.R., Turner, D.M., Fullwood, D.T. & Kalidindi, S.R. (2010) Optimized structure based representative volume element sets reflecting the ensemble-averaged 2-point statistics. *Acta Materialia*. **58**, 4432–4445.

Niezgoda, S.R., Yabansu, Y.C. & Kalidindi, S.R. (2011) Understanding and visualizing microstructure and microstructure variance as a stochastic process. *Acta Materialia*. **59**, 6387–6400.

Okabe, H. & Blunt, M.J. (2005) Pore space reconstruction using multiple-point statistics. *J. Petrol. Sci. Eng.* **46**, 121–137.

Olson, G.B. (1997) Computational design of hierarchically structured materials. *Science*. **277**, 1237–1242.

Olson, G.B. (2000) Designing a new material world. *Science*. **288**, 993–998.

Pant, L.M., Mitra, S.K. & Secanell, M. (2014) Stochastic reconstruction using multiple correlation functions with different-phase-neighbor-based pixel selection. *Phys. Rev. E*. **90**, 023306-1–023306-13.

Pedregosa, F., Varoquaux, G., Gramfort, A., *et al.* (2011) Scikit-learn: machine learning in Python. *J. Mach. Learning Res.* **12**, 2825–2830.

Piasecki, R. & Olchawa, W. (2012) Speeding up of microstructure reconstruction: I. Application to labyrinth patterns. *Model. Simulat. Mat. Sci. Eng.* **20**, 055003-1–055003-11.

Qomi, M.A., Krakowiak, K.J., Bauchy, M., *et al.* (2014) Combinatorial molecular optimization of cement hydrates. *Nat. Commun.* **5**, 1–10.

Qomi, M.J.A., Ulm, F.-J. & Pellenq, R.J.-M. (2015) Physical origins of thermal properties of cement paste. *Phys. Rev. Appl.* **3**, 064010-1–064010-17.

Quiblier, J.A. (1984) A new three-dimensional modeling technique for studying porous media. *J. Colliod. Interf. Sci.* **98**, 84–102.

Rahman, S. (2008) A random field model for generating synthetic microstructures of functionally graded materials. *Intl. J. Numer. Methods Eng.* **76**, 972–993.

Rintoul, M.D. & Torquato, S. (1997) Reconstruction of the structure of dispersions. *J. Colliod. Interf. Sci.* **186**, 467–476.

Ripley, B.D. (1996) *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, UK.

Rollett, A.D., Lee, S.-B., Campman, R. & Rohrer, G. (2007) Three-dimensional characterization of microstructure by electron backscatter diffraction. *Annu. Rev. Mater. Res.* **37**, 627–658.

Salvo, L., Cloetens, P., Maire, E., *et al.* (2003) X-ray micro-tomography an attractive characterisation technique in materials science. *Nucl. Instrum. Meth. B*. **200**, 273–286.

Sintay, S. & Rollett, A. (2012) Testing the accuracy of microstructure reconstruction in three dimensions using phantoms. *Model. Simulat. Mat. Sci. Eng.* **20**, 075005-1–075005-18.

Strebelle, S. (2002) Conditional simulation of complex geological structures using multiple-point statistics. *Math. Geol.* **34**, 1–21.

Sundararaghavan, V. (2014) Reconstruction of three-dimensional anisotropic microstructures from two-dimensional micrographs imaged on orthogonal planes. *Integr. Mater. Manuf. Innov.* **3**, 1–11.

Tahmasebi, P., Hezarkhani, A. & Sahimi, M. (2012) Multiple-point geostatistical modeling based on the cross-correlation functions. *Computational. Geos.* **16**, 779–797.

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

Talukdar, M., Torsaeter, O., Ioannidis, M. & Howard, J. (2002) Stochastic reconstruction, 3D characterization and network modeling of chalk. *J. Petrol. Sci. Eng.* **35**, 1–21.

Tang, T., Teng, Q.-z. & He, X.-h. (2008) A hybrid reconstruction method of sandstone from 2D section image. In: *International Conference on Neural Networks and Signal Processing, 2008*. IEEE, USA.

Tang, T., Teng, Q.-z., He, X.-h. & Luo, D. (2009) A pixel selection rule based on the number of different-phase neighbours for the simulated annealing reconstruction of sandstone microstructure. *J. Microsc.* **234**, 262–268.

Tewari, A. & Gokhale, A. (2004) Nearest-neighbor distances between particles of finite size in three-dimensional uniform random microstructures. *Mat. Sci. Eng. A*. **385**, 332–341.

Torquato, S. (2002a) *Random Heterogeneous Materials: Microstructure and Macroscopic Properties*. Springer, USA.

Torquato, S. (2002b) Statistical description of microstructures. *Ann. Rev. Mat. Res.* **32**, 77–111.

Torquato, S. (2006) Necessary conditions on realizable two-point correlation functions of random media. *Ind. Eng. Chem. Res.* **45**, 6923–6928.

Torquato, S. (2010) Optimal design of heterogeneous materials. *Ann. Rev. Mat. Res.* **40**, 101–129.

Torquato, S. & Stell, G. (1982) Microstructure of two-phase random media. I. The n-point probability functions. *J. Chem. Phys.* **77**, 2071–2077.

Ward, C. (2012) Materials genome initiative for global competitiveness. In: *23rd Advanced Aerospace Materials and Processes (AeroMat) Conference and Exposition*. ASM, USA.

Wei, L.-Y. & Levoy, M. (2000) Fast texture synthesis using tree-structured vector quantization. In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press/Addison-Wesley Publishing Co, New Orleans, LA, USA.

Wilson, S., Hefferen, C., Li, F., Suter, R. & Rollett, A. (2010) Microstructural characterization and evolution in 3D. In: *Proceedings of the 31st Risø International Symposium on Materials Science*. Technical University of Denmark, Denmark.

Wu, K., Nunan, N., Crawford, J.W., Young, I.M. & Ritz, K. (2004) An efficient Markov chain model for the simulation of heterogeneous soil structure. *Soil Sci. Soc. Am. J.* **68**, 346–351.

Xu, H., Greene, M.S., Deng, H., *et al.* (2013) Stochastic reassembly strategy for managing information complexity in heterogeneous materials analysis and design. *J. Mech. Design*. **135**, 101010-1–101010-12.

Xu, H., Li, Y., Brinson, C. & Chen, W. (2014) A descriptor-based design methodology for developing heterogeneous microstructural materials system. *J. Mech. Design*. **136**, 051007-1–051007-12.

© 2016 The Authors

Yeong, C. & Torquato, S. (1998a) Reconstructing random media. *Phys. Rev. E.* **57**, 495–506.

Yeong, C. & Torquato, S. (1998b) Reconstructing random media. II. Three-dimensional media from two-dimensional cuts. *Phys. Rev. E.* **58**, 224–233.

Zachary, C.E. & Torquato, S. (2011) Improved reconstructions of random media using dilation and erosion processes. *Phys. Rev. E.* **84**, 056102.

Zhang, Y., Zhao, H., Hassinger, I., Brinson, L.C., Schadler, L.S. & Chen, W. (2015) Microstructure reconstruction and structural equation modeling for computational design of nanodielectrics. *Integr. Mater. Manuf. Innov.* **4**, 1–26.

Appendix

Two-point correlation and lineal-path functions

Following the notation introduced in the 'Review of the 2D supervised learning approach for stochastic microstructure C&R' section, we have:

$$X_{ijk} = \begin{cases} 1 \text{ if } ijk \in \text{phase } 1 \\ 0 \qquad \text{otherwise} \end{cases},$$

here $i\,j\,k$ is the voxel index and determines its location within the image. Denoting this location by the vector $\boldsymbol{r}$, the two-point correlation function for phase i is defined as:

$$S_2^{(i)}(\boldsymbol{r}_1, \boldsymbol{r}_2) = X(\boldsymbol{r}_1)\, X(\boldsymbol{r}_2),$$

where the angular brackets denote the expectation operator. $S_2^{(i)}$ can be thought of as the probability of tossing a line on $\boldsymbol{X}$ and having both its ends land on phase i. If $\boldsymbol{X}$ is statistically isotropic, $S_2^{(i)}$ will only depend on the Euclidian distance between the two points and the orientation will not matter. Furthermore, if $\boldsymbol{X}$ is homogeneous, $S_2^{(i)}$ will be translationally invariant. Hence, for a homogeneous and isotropic material, $S_2^{(i)}$ has a simplified formulation ($S_2^{(i)}(\boldsymbol{r}_1, \boldsymbol{r}_2) = S_2^{(i)}(\Delta\boldsymbol{r}_{12}) = S_2^{(i)}(|\Delta\boldsymbol{r}_{12}|)$) and can be efficiently calculated via fast Fourier transform (Berryman, 1985; Fullwood *et al.*, 2008a).

Other methods such as Monte Carlo (MC) can also be used. For example, for a homogeneous and isotropic structure a sample MC procedure is as follows: A randomly oriented line with length l_k ($0 \leq k \leq K$) is first randomly thrown on $\boldsymbol{X}$ for a total of N times. Next, the number of times that the thrown line has both its ends in phase i is calculated and then divided by N. This process is done for all k (we chose the maximum length as half of the original image size).

Lineal-path function ($L^{(i)}$) is also similar to $S_2^{(i)}$ in that it captures the probability of randomly throwing a line on $\boldsymbol{X}$ and having the whole line land on phase i. $L^{(i)}$ contains partial information on connectedness and hence sometimes underestimates the clusteredness (i.e. if two points are connected via a curved path, they will not contribute to $L^{(i)}$).

Simulation results without the user-defined predictors

Table A1 summarizes the statistical evaluation results for all the examples when the user-defined predictors are not incorporated into the fitted models. All the other parameters (i.e. the neighbourhood size) are kept the same as those reported in Table 1.

Table A1. Performance of the full and partial models without incorporating the user-defined predictors into the supervised learner. See Table 1 for more details on the measures.

Example, Model	$\overline{\lvert\Delta VF\rvert}$	$\overline{\Delta \mathbf{S}_2(r)}$	$\overline{\Delta \mathbf{L}(r)}$	$\overline{\Delta SS}$
1, Full 3D model	8.42×10^{-4}	4.17×10^{-6}	1.21×10^{-5}	2.12×10^{-4}
1, Partial 3D model	6.40×10^{-4}	9.70×10^{-6}	2.01×10^{-5}	3.14×10^{-4}
2, Full 3D model	9.18×10^{-4}	3.83×10^{-4}	1.16×10^{-3}	2.22×10^{-3}
2, Partial 3D model	8.38×10^{-4}	5.32×10^{-4}	1.15×10^{-3}	2.31×10^{-3}
3, Full 3D model	1.10×10^{-4}	9.77×10^{-4}	8.67×10^{-4}	4.40×10^{-3}
3, Partial 3D model	1.03×10^{-3}	2.02×10^{-3}	3.33×10^{-3}	9.08×10^{-3}

© 2016 The Authors