Computer-Aided Design 128 (2020) 102906

Contents lists available at ScienceDirect

Computer-Aided Design

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

Reconstruction of 3D Microstructures from 2D Images via Transfer Learning☆

Ramin Bostanabad

Department of Mechanical and Aerospace Engineering, University of California, Irvine, United States of America

ARTICLE INFO

Article history:
Received 2 February 2020
Received in revised form 8 May 2020
Accepted 13 June 2020

Keywords:
Reconstruction
Microstructure
Statistical equivalency
Transfer learning
Correlation functions

ABSTRACT

Computational analysis, modeling, and prediction of many phenomena in materials require a three-dimensional (3D) microstructure sample that embodies the salient features of the material system under study. Since acquiring 3D microstructural images is expensive and time-consuming, an alternative approach is to extrapolate a 2D image (aka exemplar) into a virtual 3D sample and thereafter use the 3D image in the analyses and design. In this paper, we introduce an efficient and novel approach based on transfer learning to accomplish this extrapolation-based reconstruction for a wide range of microstructures including alloys, porous media, and polycrystalline. We cast the reconstruction task as an optimization problem where a random 3D image is iteratively refined to match its microstructural features to those of the exemplar. VGG19, a pre-trained deep convolutional neural network, constitutes the backbone of this optimization where it is used to obtain the microstructural features and construct the objective function. By augmenting the architecture of VGG19 with a permutation operator, we enable it to take 3D images as inputs and generate a collection of 2D features that approximate an underlying 3D feature map. We demonstrate the applications of our approach with nine examples on various microstructure samples and image types (grayscale, binary, and RGB). As measured by independent statistical metrics, our approach ensures the statistical equivalency between the 3D reconstructed samples and the corresponding 2D exemplar quite well.

1. Introduction

The heart of computational materials science lies in unraveling processing-microstructure–property relations across multiple length and time scales which, in turn, provides the foundation to design materials with unprecedented properties. Computational analysis, modeling, and prediction of many phenomena in materials (such as fatigue [1], fracture [2], transport properties [3], manufacturing-induced residual stresses [4], delamination [5]) require a three-dimensional (3D) microstructure sample[1] that embodies the most important features of the microstructure under study (e.g., size and spatial distribution of inclusions or voids). Recent advances in imaging techniques [6–8] have enabled the collection of high resolution 3D maps of microstructures that can be used in computer simulations. However, obtaining 3D images is extremely costly and time-consuming. Since acquiring 2D microstructural images is relatively simpler and inexpensive, an attractive alternative is (upon making some assumptions) to extrapolate a 2D image (aka exemplar) into a virtual 3D sample and thereafter use the 3D image in simulations. The goal of this paper is to introduce an efficient and novel approach based on transfer learning to accomplish this extrapolation-based reconstruction for a wide range of microstructures (composite, alloy, porous, polycrystalline, etc.) and image types, i.e., single channel (binary or grayscale) or multi-channel (RGB).

From a historical standpoint, computational microstructure reconstruction can be traced back to 1974 where Joshi [9] and then Quiblier [10] reconstructed porous media by level-cutting Gaussian random fields (GRFs) whose correlation functions were fitted to experimental data on the scattering power of electromagnetic radiation [11–14]. Since then, many methods have been developed to reconstruct microstructures from a given exemplar or some data that characterizes the exemplar (e.g., scattering power). In all these methods, microstructure reconstruction is cast as an optimization problem with the goal of minimizing an appropriately defined loss (aka cost or energy) function that measures the differences between the characterized information obtained from the exemplar and the microstructure being reconstructed. Since such information generally has a probabilistic nature (e.g., void spatial distribution or average pore size), the

☆ This paper has been recommended for acceptance by A. Pasko.

E-mail address: Raminb@uci.edu.

[1] The three dimensions refer to height, width, and depth (or thickness). Number of channels is not counted toward the dimensionality of the image in this paper. For instance, an RGB image with H and W pixels along its height and width, respectively, is 2D and needs a 3D array of size $h \times w \times 3$ for storage in a computer.

https://doi.org/10.1016/j.cad.2020.102906
0010-4485/

reconstructed microstructure is said to be *statistically equivalent* to the exemplar. With this viewpoint, different methods can be categorized based on the characterization scheme which, consequently, determines the features that the 3D microstructure should possess. Following this classification, relevant works to this paper that can extrapolate a 2D exemplar into a 3D structure are reviewed below.

A microstructure can be characterized with some statistical functions such as n-point correlation [11,15], lineal-path [16–19], and cluster correlation [20]. These functions essentially quantify microstructural features with some probability distributions. Assuming these probabilities are the same between the 2D exemplar and the desired 3D microstructure, one can build a 3D microstructure by refining an initial random 3D structure such that the differences between its statistical functions and those of the exemplar are minimized. The so-called stochastic reconstruction (SR) method achieves this minimization using a heuristic optimization algorithm (e.g., simulated annealing [21–24] or genetic algorithm [25–27]) where the voxels of the initial 3D image are randomly swapped until the loss function that measures the statistical differences is minimized. The SR method has been previously used to reconstruct 3D porous media from 2D exemplars [25,28–36].

All the variants of the SR method incorporate the two-point correlation function ($S_2(r)$ where r measures the pixel-wise or voxel-wise distance in an image) into the loss function. Depending on the microstructure, additional measures such as the lineal-path function, $L(r)$, are also considered in the loss function to ensure the reconstructed microstructure embodies the important features of the exemplar [16,20,23,37–39], see Appendix for technical details on $S_2(r)$ and $L(r)$. This addition, while increasing the reconstruction fidelity, has two main disadvantages (*i*) it significantly increases the computational costs [40–42], and (*ii*) it is problem dependent since complete characterization of different microstructures (e.g., polycrystalline, particulate, porous, or stochastic) requires different statistical functions; rendering the optimal choice of statistical functions ambiguous in some cases [43]. It is noted that, assuming a microstructure can be completely characterized solely via two-point correlation function, the phase recovery method [44–49] can be used instead of the SR method for reconstruction. The reconstruction cost in SR and phase recovery methods rapidly grows as the number of phases increases. Hence, due to computational reasons, these methods are incapable of reconstructing grayscale or RGB images.

As an alternative to statistical functions, physical descriptors (e.g. average pore or grain size) can be used to characterize the 2D exemplar. In this case, the heuristic optimization process aims to match the characteristics of the descriptors in the reconstructed image to the corresponding ones in the exemplar with some assumptions [50–56]. Characterization via physical descriptors is particularly desirable for microstructure design [57] as it sensibly models topological features and allows to build process-structure–property relations. In the context of microstructure reconstruction, however, this method has some limitations. Firstly, it relies on image analysis to extract the characteristics of the descriptors from the exemplar and is only applicable to certain microstructures (e.g., particulate or semi-particulate) with limited number of phases (i.e., inapplicable to grayscale or RGB images). Additionally, one needs to a priori know the descriptors that should be used during characterization, image analysis, and optimization. For example, because transport processes in particulate heterogeneous materials is sensitive to nearest neighbor distances between particles [58], one has to match the distributional characteristics (e.g., the mean and variance) of this descriptor in the exemplar and the 3D reconstructed sample.

Recently, some methods based on texture synthesis have been employed for microstructure reconstruction [59–63] with the assumption that the exemplar can be characterized as a stationary Markov random field (MRF). Texture synthesis originated in computer graphics with the basic idea of synthesizing a texture (i.e., a stationary image) given a small exemplar. The early works in [64–66] established the general framework of texture synthesis but it was the pioneering works of Efros and Leung [67,68] and Wei and Levoy [69] that resulted in practical solutions by reformulating the problem of texture synthesis to an exemplar-based framework. In short, these works addressed the ill-posed nature of the problem by assuming that the exemplar is a realization of a stationary MRF and has the spatial locality property. These assumptions imply that the conditional probability distribution of brightness value of any pixel (voxel in 3D), given the brightness values of its neighboring pixels, is independent from the rest of the image. This conditional probability distribution, however, is not explicitly modeled and a 3D image is reconstructed by repeatedly querying the given 2D exemplar. Specifically, the reconstruction is usually done voxel-by-voxel in a specific order (e.g. raster scan) where each voxel's value in the reconstructed image is found by searching for the pixel (or set of pixels) in the exemplar whose neighboring pixels best match the neighbors (along three orthogonal planes) of the voxel to be generated. Reconstructing large microstructures that do not satisfy the stationary MRF assumptions via this method is impractical. Additionally, the quality of the reconstructed images greatly depends on the neighborhood size used in the search process (which is oftentimes chosen manually via ad hoc methods) [70] and the reconstruction order has been shown [71] to affect microstructure quality in texture synthesis (e.g. a bottom-up raster-scan vs. a top-down raster-scan).

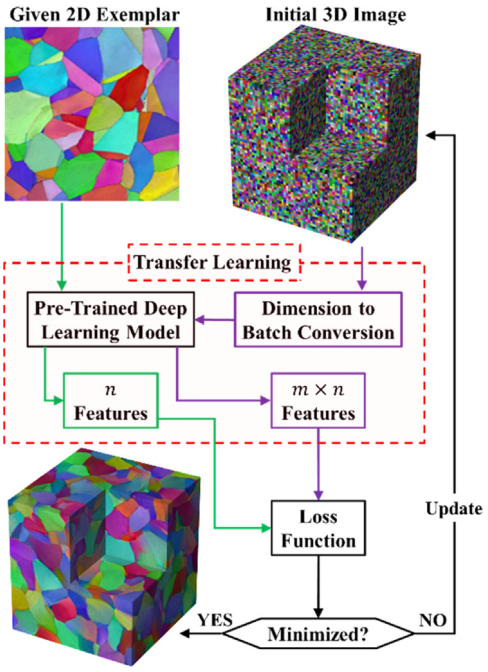

Fig. 1. (Color online) Transfer learning for 3D microstructure reconstruction: The colors (green and purple) distinguish the information flow from the exemplar and initial image to the loss function.

In this paper, a novel approach based on transfer learning is developed to reconstruct a 3D microstructure using a single 2D exemplar. The fundamental idea is, as demonstrated in Fig. 1, to restructure a pre-trained 2D deep learning model[2] in such a way

[2] A 2D deep learning model refers to one that takes one (or a batch of) 2D image(s) as input.

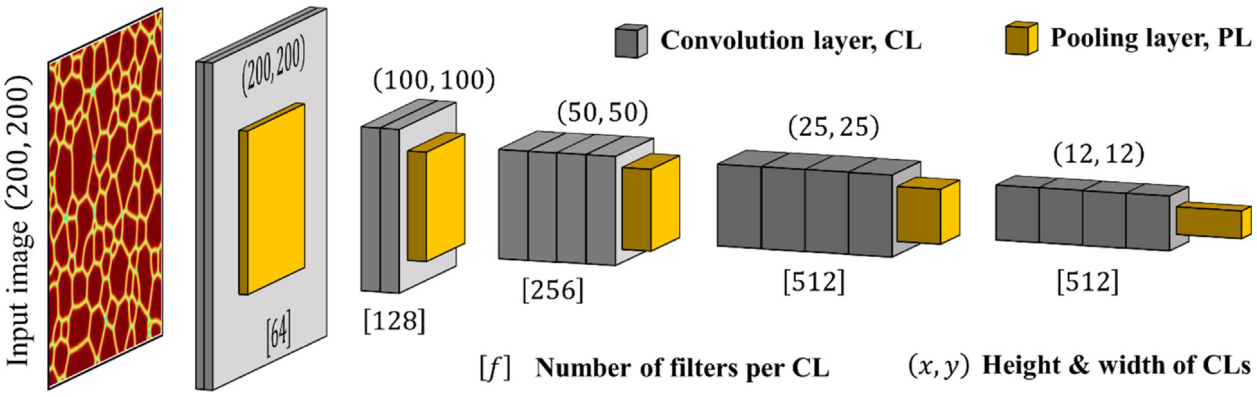

Fig. 2. (Color online) Internal architecture of VGG19: This pre-trained deep learning model has 5 blocks where each block has either two or four convolution layers as well as one average pooling layer at the end. While the size of the layers decreases from left to right (along the depth of VGG19), the number of filters (aka channels) per layer increases. The dense layers at the end of VGG19 are not demonstrated since they are for labeling images and not used in this paper.

that a 3D image can be used as its input. Upon this restructuring, reconstruction is cast as an optimization problem where an initial random image is optimized to match its microstructural features to the exemplar's features. We employ a pre-trained deep learning model for microstructure characterization because it can automatically extract a wide range of features (grains, edges, particles, etc.) from different types of microstructures, see Fig. 6 through Fig. 8 for some examples. We elaborate on the nature of these features and how they are used in our approach in Sections 2.1 and 2.3, respectively.

It is highlighted that if a pre-trained 3D model was available, the abovementioned restructuring would not be required. However, to the author's best knowledge, a general-purpose pre-trained 3D model that can extract a wide range of microstructural features is unavailable. Additionally, a 3D model would need a 3D exemplar as input which violates the premise of this work (i.e., one does not have a 3D image and would like to build one given a 2D exemplar). Due to these reasons, we employ a pre-trained 2D deep learning model in our approach. 2D deep models such as VGG16 and VGG19 [72], Inception [73], and Xception [74] are extremely powerful in quantitatively characterizing a wide range of 2D images and our restructuring mechanism allows to extend their power to 3D.

While transfer learning has been previously used for microstructure reconstruction [40,75–77], the efforts have been limited to reconstructing 2D binary or ternary structures that are statistically equivalent to a given 2D exemplar. The primary differentiating factors between this paper and [40,75–77] are extrapolation from 2D to 3D *and* exclusion of any image pre- and post-processing procedure (e.g., smoothing or refining of any sort). Additional advantages of this work over the state-of-the-art methods (e.g., SR, texture synthesis, and phase recovery) include reasonable computational expenses and applicability to a wide range of microstructures (e.g., composites, polycrystalline, alloys, porous, and inhomogeneous) and image types (binary, grayscale, or RGB) without any modification to the method.

The rest of the paper is organized as follows. In Section 2 some background on transfer learning is provided with more emphasis on the pre-trained deep learning model, VGG19 [72], used in this paper. Then, the proposed method is introduced and details on the restructuring idea, feature extraction, loss function, and optimization are discussed. In Section 3, nine different 3D samples are reconstructed to demonstrate the applicability of the method to different microstructures and image types. Concluding remarks and future works are provided in Section 4.

2. Reconstruction via extrapolative transfer learning

We first describe how 2D microstructural features can be extracted via VGG19. Then, our idea on dimension to batch conversion is introduced to enable the analysis of 3D images with VGG19. In Section 2.3 we employ these features to formulate the loss function and in Section 2.4 we elaborate on the optimization procedure. Further details are provided in the Appendix A.3.

2.1. Feature extraction via transfer learning

The overarching goal of transfer learning is to improve learning a new task by transferring the knowledge obtained from learning a related task [78]. In the context of this paper, knowledge transfer refers to quantifying the statistical differences between two microstructures (task 1) using a deep learning model that is trained to label a massive dataset of images (task 2). The relevance between these two tasks is that the features that a pre-trained model uses for labeling images[3] in task 2 can also be used to measure the statistical differences between two microstructures. To explicitly state what these features are and how they are used for microstructure characterization, we first elaborate on the pre-trained deep learning model employed in this work, i.e., VGG19.

VGG19 is a deep convolutional neural network (CNN) trained on the ImageNet dataset [79] for object classification. The objects in ImageNet are very diverse and include humans, animals, buildings, vehicles, etc. under various imaging conditions such as posing, angle, lightening, etc. As illustrated in Fig. 2, VGG19 consists of five blocks where each block has either two or four convolution layers as well as one average pooling layer at the end. Each convolution layer stores a set of filters (aka channels) which encode specific features of the input image. Once an image is used as input in VGG19, these filters are activated where each filter produces a *response map* that indicates the locations on the input image where the encoded feature by the filter is present, see Fig. 3 for some examples.

It is well documented in the literature [73,74,80] that in a deep CNN the first layer stores a collection of filters that act as various edge detectors. As one goes deeper into the CNN (i.e., from left to right in Fig. 2), the filters begin to detect more complex and less visually interpretable features. Fig. 3 demonstrates this behavior of deep CNNs when two different microstructures are fed into VGG19: The response maps from the filters in the first layer of

[3] These images include many objects (e.g., animals and humans) but exclude microstructures.

block 1 can effectively detect the edges and, in turn, segment different parts of the image (e.g., the grains or inclusions can be clearly detected using these filters). Understanding what filters detect becomes increasingly complex as one goes deeper into the model to the extent that interpreting the features that the filters of block 5 detect is very difficult (see the last column in Fig. 3).

2.2. Model restructuring: Dimension to batch conversion

2D pre-trained deep CNNs such as VGG19 accept a 4D tensor of size $b \times h \times w \times c$ as input where these letters denote, respectively, batch size (the number of 2D images), image height and width (in pixels), and number of channels. Since the most widely used pre-trained 2D models are trained on RGB images, $c = 3$. If a 3D microstructure image stored in a 4D tensor of size $t \times h \times w \times 3$ (t denotes thickness) is used as an input to VGG19, the model will produce features (i.e., response maps) from t images of size $h \times w \times 3$, i.e., VGG19 will operate as if it has received t *independent* 2D images. With such an independent treatment of 2D parallel images along the thickness, 3D microstructural features cannot be extracted.

To address this issue, our novel and yet simple idea is to enforce VGG19 to produce a collection of *dependent* 2D feature maps that approximate an underlying 3D feature map. Producing dependent feature maps can be potentially achieved in many ways. In this work, we accomplish this by adding a *permutation operator* to VGG19's architecture (after the input and before the first convolution layer in Fig. 2) such that it simultaneously considers 2D cross-sections along thickness, height, and width to stand for batch, i.e., the dimensions are converted to batch from VGG19's perspective, see Fig. 4. The advantages of this idea are twofold:

- It can be applied to any pre-trained deep CNN without the need to alter its internal architecture.
- It results in a set of 2D response maps that can be directly compared to the 2D response maps of the exemplar (as opposed to 3D response maps that cannot be directly compared to the exemplar's 2D response maps).

Assuming VGG19 has a total of n filters across its five blocks, this approach will produce a total of $3 \times s \times n$ response maps for a 3D microstructure of size $s \times s \times s$ voxels (note that the response maps are all 2D images but of different size, see Figs. 3 and 5). In Fig. 5 three response maps corresponding to the first filter of each block's first layer are illustrated. For visual clarity, the maps are only obtained for the orthogonal planes overlaid on the input (for the entire 3D input image, each filter produces three 3D response maps). Similar behavior to Fig. 3 is observed across the blocks in Fig. 5, i.e., the response maps become increasingly abstract and difficult to interpret as one goes deeper into VGG19 (i.e., from left to right in Fig. 5).

2.3. Loss function

Since our goal is to reconstruct a 3D microstructure that is statistically equivalent to a 2D exemplar, the response maps discussed in Section 2.2 cannot be directly used in the loss function, $\mathcal{L}(\cdot)$. This is because $\mathcal{L}(\cdot)$ should quantify the differences between some statistical measures while the response maps are deterministic features. In other words, one cannot necessarily obtain a statistically equivalent 3D microstructure if the optimization problem aims to match the response maps of the 3D microstructure (see Fig. 5) with those of the exemplar (see top row in Fig. 3).

Spatial statistics (e.g., void or grain spatial distribution) are significantly important in microstructural analyses and must be considered in reconstruction. Capturing these spatial statistics requires finding the correlations among different response maps. For instance, if two response maps characterize, respectively, grain size and orientation, finding the correlation between these two response maps would determine how grain size and orientation are linked in the microstructure. This requirement motivates employing the correlations between the response maps [81] in $\mathcal{L}(\cdot)$. To formally define these correlations and $\mathcal{L}(\cdot)$, some variables are first introduced.

Let b, l_b, s_b, and f_{lb} denote, respectively, block b in VGG19, the number of layers in block b, size (in pixels) of the layers in block b, and the number of filters within layer l in block b. For example, in the first block: $b = 1, l_b = 2, f_{lb} = 64$, and $s_b = 200$ for an input of size 200×200 (note that in VGG19 all the convolution layers within a block are of the same size and have the same number of filters, see Fig. 2). Once the exemplar is fed into VGG19, a total of f_{lb} response maps are generated by the

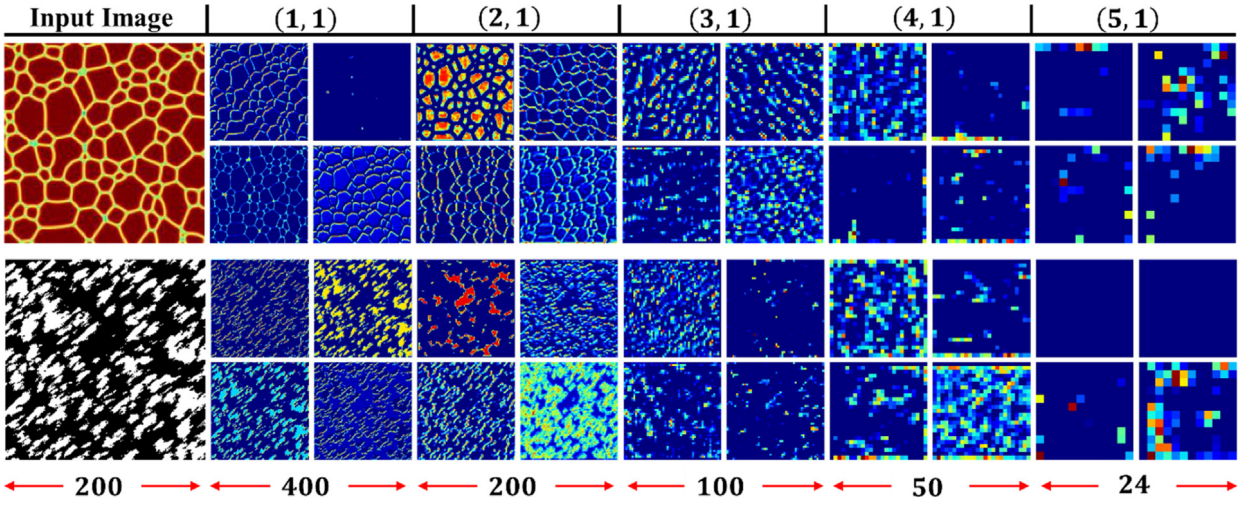

Fig. 3. (Color online) Sample response maps in VGG19 with 2D inputs: The response map of the first four filters of each block's first layer are illustrated for two input images. The pairs on the top of the figure correspond to (block ID, layer ID within the block). Each quadrant under each column indicates the response map of one filter. The images have been resized to facilitate the visual comparison and the actual size can be inferred via the pixel counts at the bottom. For example, each response map under column (1, 1) has a side length of 200 pixels. The colors in each response map indicate the features that the filter detects. To facilitate comparison, the colors in the response maps have been scaled to $[0, 255]$ range where 0 and 255 denote blue and red, respectively.

Fig. 4. (Color online) Dimension to batch conversion: By adding the permutation operator, VGG19 treats each dimension as batch and produces 2D feature maps along thickness, height, and width. The colors indicate how the permutation operator affects the input.

filters of layer l in block b. By vectorizing these 2D response maps, one can store them in the matrix $\boldsymbol{R}^{lb}$ of size $f_{lb} \times s_b^2$ where the ith row stores the vectorized form of the ith response map. The correlations between the response maps within layer l of block b are calculated (up to a proportionality constant) via [81]:

$$\boldsymbol{G}^{lb} = \boldsymbol{R}^{lb} * \left(\boldsymbol{R}^{lb}\right)^T, \tag{1}$$

where $*$ and T denote the matrix product and transpose operation, respectively. $\boldsymbol{G}^{lb}$ is commonly known as the Gram matrix.

When a 3D image is fed into VGG19, each filter in block b generates $3 \times s_b$ 2D response maps (as opposed to one 2D map when a 2D input is used). The $3 \times s_b$ number is the result of the permutation operation discussed in Section 2.2, see Fig. 5. Hence, each layer in block b will have $3 \times s_b$ Gram matrices each of size $f_{lb} \times f_{lb}$. Denoting these Gram matrices by $\boldsymbol{C}^{dlb}$ for $d = 1, 2, \ldots, 3 \times s_b$ the loss function can be written as:

$$\mathcal{L}(\boldsymbol{x}) = \sum_{b=1}^{5} \sum_{l=1}^{l_b} \sum_{d=1}^{3\times s_b} \sum_{ij} \left(C_{ij}^{dlb} - G_{ij}^{lb}\right)^2, \tag{2}$$

where $\boldsymbol{x}$ denotes the RGB triplets for all the voxels of the 3D microstructure. For example, $\boldsymbol{x} = \left[x_i^r, x_i^g, x_i^b\right]^T$, $i = 1, \ldots, 200$ when reconstructing a 3D colorful microstructure of size $200 \times 200 \times 200$ voxels.

The loss function in Eq. (2) is known to produce high frequency signals (i.e., noise) upon reconstruction. To reduce the noise, a regularization term known as the total variation loss [82] is added to the loss function:

$$\mathcal{L}(\boldsymbol{x}) = \sum_{b=1}^{5} \sum_{l=1}^{l_b} \sum_{d=1}^{3\times s_b} \sum_{ij} \left(C_{ij}^{dlb} - G_{ij}^{lb}\right)^2 + \lambda p(\boldsymbol{x}), \tag{3}$$

where λ is the regularization parameter that can be estimated very efficiently as detailed in Appendix A.3. $\lambda = 10^{12}$ is used for all the examples in Section 3.

We now discuss three important aspects of $\mathcal{L}(\boldsymbol{x})$ in Eq. (3). Firstly, the gram matrices are obtained up to some proportionality constants. These constants are excluded from Eq. (3) as their presence would merely change the estimate on λ. Secondly, the Gram matrices of the exemplar do not change under the summation over d. Since this summation essentially makes VGG19 to *look* at 2D cross-sections of the 3D microstructure from different angles, the invariance of exemplar's Gram matrices under this summation implies that any 2D cross-section taken from the 3D microstructure inherits its statistics from the exemplar. Note that if a 3D microstructure cannot be adequately characterized with a single 2D exemplar (e.g., an anisotropic microstructure), Eq. (3) must be updated to use more than one 2D exemplar in reconstruction. We elaborate more on dealing with anisotropy in Section 3.4. Lastly, the extracted features in each layer are assumed to be equally important since there are no weights inside the summation over l (the weighted summation would have been $\sum_{l=1}^{l_b} w_l \sum_{d=1}^{3\times s_b} \ldots$). This choice reflects the equal importance of all the features in the exemplar. Noting that the filters in the deep layers in VGG19 (e.g., layers in block 5) capture longer range correlations than the filters in the proceeding layers (e.g., layers in block 1), an unweighted sum also implies that short- and long-range correlations are equally important. We believe these choices increase the generality of our method and make it applicable to various microstructures as illustrated in Section 3.

2.4. Optimization

As the flowchart in Fig. 1 illustrates, the goal of optimization is to iteratively adjust a randomly generated 3D image such that it becomes statistically equivalent to the exemplar. This adjustment is achieved by minimizing $\mathcal{L}(\boldsymbol{x})$:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \mathcal{L}(\boldsymbol{x}), \tag{4a}$$

$$s.t.: 0 \le x_i \le 255, i \in \left[1, \ldots, 3 \times z^3\right], \tag{4b}$$

where z is the side-length of the 3D image and $\hat{\boldsymbol{x}}$ is the desired 3D microstructure whose features (as defined in the previous section) match those of the exemplar. The constraints in Eq. (4b) assume that each voxel in the 3D image requires an RGB triplet. For grayscale and binary images the constraints would be, respectively, $0 \le x_i \le 255$ and $x_i = 0$ *or* 1 where $i \in \left[1, \ldots, z^3\right]$ in both cases.

The optimization in Eq. (5) can be solved in different ways. However, a gradient-based technique is always used due to the extremely high-dimensionality of the problem and, more importantly, because the gradient of $\mathcal{L}(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ can be

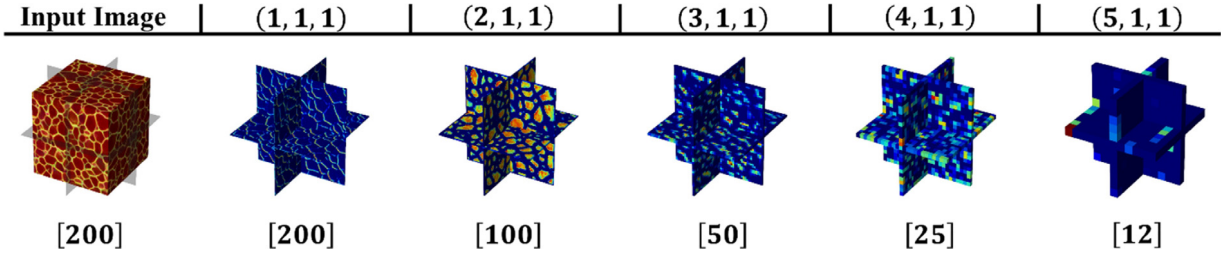

Fig. 5. (Color online) Sample response maps in VGG19 with a 3D input: The response maps of the first filter of each block's first layer are illustrated for a 3D input. The triplets on the top of the figure correspond to block ID, layer ID within the block, and filter ID within the layer. The numbers at the bottom indicate the side length in voxels. For clarity, the response maps are only plotted for the orthogonal planes overlaid on the input image. For the entire image, each filter generates 3×200 2D response maps in total (three of which are shown).

analytically derived and implemented via the standard backpropagation algorithm [81,83].

3. Results and discussions

In this section, the developed method is used to reconstruct eight 3D microstructures that belong to different microstructures (e.g., porous, composite, etc.) and each has unique features that the approach is able to characterize and reconstruct quite well. The example (Ex) sets are categorized based on the image format, i.e., grayscale, binary, and RGB. The method is developed in Python and leverages TensorFlow [84] for transfer learning and the L-BFGS-B algorithm [85] for optimization where the termination criterion and λ are set to 1000 iterations and 10^{12}, respectively. All computations are carried out on GPU (NVIDIA® Quadro RTX™ 8000).

The two-point correlation, lineal-path, and two-point cluster correlation functions are used to validate the statistical equivalency between the reconstructed microstructures and the corresponding exemplars in Ex1 through Ex6 (see Appendix for implementation details). For the polycrystalline microstructures (Ex7 and Ex8), the distribution of the normalized grain size is employed to compare the reconstructed samples with the exemplars. These measures are chosen as validation metrics because they are linked to material properties [13,86,87] and ubiquitously employed in the literature [41,59,61–63,75–77,88–92]. In all examples, the side length of the exemplars and the reconstructed microstructures is 200 pixels/voxels. Details on how to apply the optimization problem in Eq. (5) directly to grayscale and binary images are provided.

3.1. Example set 1: Grayscale images

Grayscale images are ubiquitous in materials science as many devices such as scanning electron microscope and transmission electron microscope produce grayscale images. As reviewed in Section 1, most reconstruction methods are inapplicable to grayscale images and threshold the exemplars to limit their pixel brightness values to a few numbers (e.g., 0/1 or 0/255 for two-phase microstructures). Using three examples, we illustrate that our approach is directly applicable to various grayscale images, see Fig. 6.

The microstructure in Ex1 belongs to steel where the dark regions indicate secondary phase which has various shapes and sizes. The size and spatial distribution of Pearlites are highly important in steel and are directly linked to manufacturing processes and properties. In Ex2, an electron micrograph of tempered martensite is investigated where the small particles and the matrix indicate cementite and α-ferrite phases, respectively. Ex3 is on an aluminum alloy prepared by high pressure die casting [93]. In all these examples, the size, shape, and spatial distribution of the secondary phase (pores, inclusions, defects, etc.) are of particular interest.

The reconstruction results are demonstrated in the second row of Fig. 6 and indicate that the developed method can characterize microstructural features of various nature. For example, the microstructure in Ex1 mostly has randomly oriented large secondary phase while the microstructure in Ex3 has only a few elliptical voids. The micrograph of the tempered martensite in Ex2 has even more interesting features. Firstly, the elliptical particles have various aspect ratios and sizes. Secondly, there seems to be a

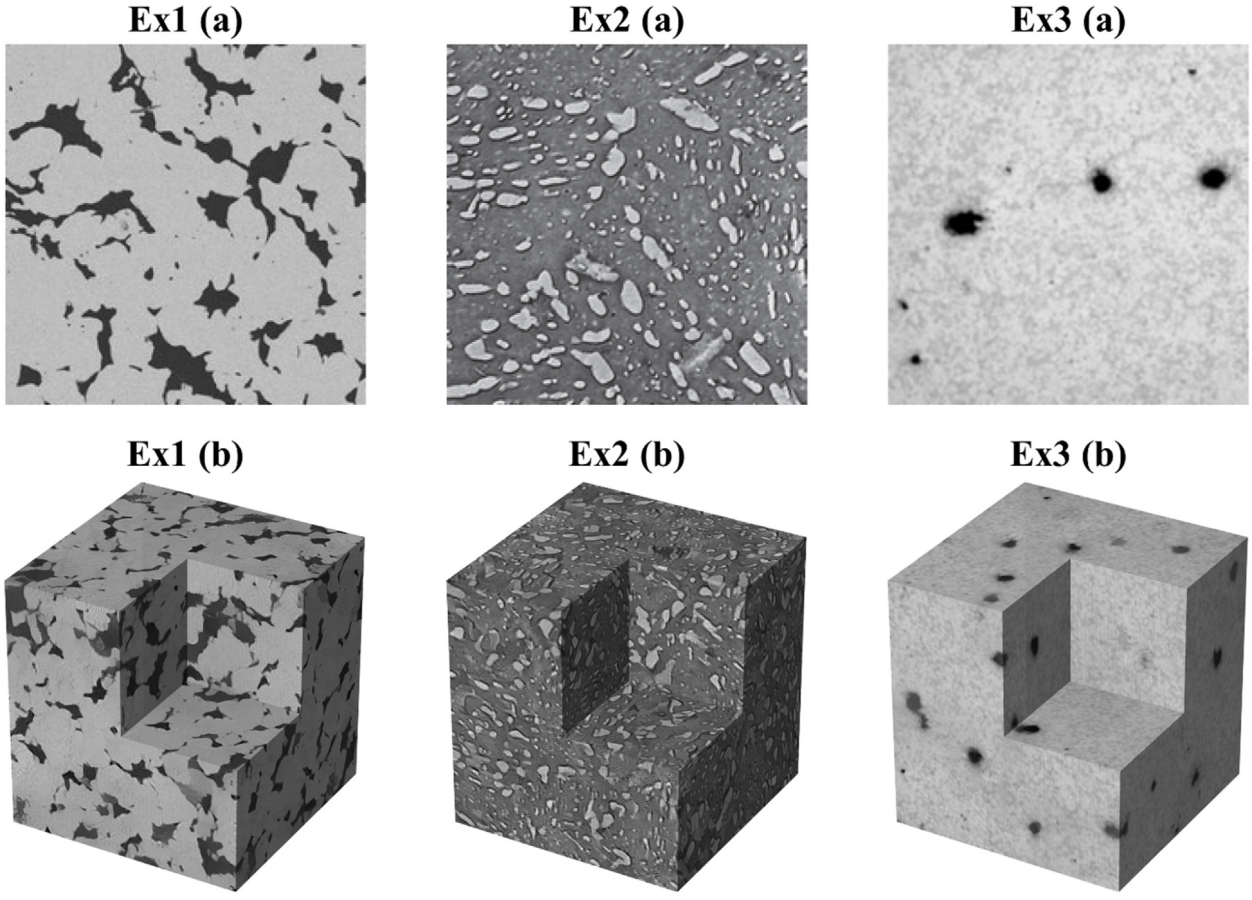

Fig. 6. Examples on grayscale images: **(a)** The 2D exemplars of size 200×200 pixels, and **(b)** the 3D reconstructed microstructures of size $200 \times 200 \times 200$ voxels. Part of the reconstructed images is removed to reveal the interior.

Table 1
Average errors and reconstruction time.

	Ex1	Ex2	Ex3	Ex4	Ex5	Ex6
Error in $S_2\,(r)$	3.21×10^{-3}	1.06×10^{-3}	3.21×10^{-4}	2.30×10^{-5}	1.69×10^{-3}	2.11×10^{-3}
Error in $L\,(r)$	1.59×10^{-3}	1.55×10^{-3}	2.21×10^{-4}	6.93×10^{-5}	3.91×10^{-4}	1.28×10^{-3}
Error in $C_2\,(r)$	3.02×10^{-3}	1.65×10^{-3}	3.17×10^{-4}	7.64×10^{-5}	1.82×10^{-3}	3.62×10^{-3}
Time	110.1	104.9	113.2	104.2	105.2	107.9

The reconstructed microstructures are compared to the exemplars by measuring the root mean squared error (RMSE) in two-point correlation, $S_2\,(r)$, lineal-path, $L\,(r)$, and two-point cluster correlation, $C_2\,(r)$, functions. The reconstruction time is in minutes. In Ex6, calculations are done along the anisotropy direction. The data are averaged over ten simulations. The variations are small and hence no reported.

correlation between the orientation of the particles' major axes and their spatial location. For example, the particles on the top left corner are aligned horizontally while those close to the right edge (in the middle of the image) are aligned vertically. Such features render the microstructure in Ex2 non-stationary. These interesting features in Ex1 through Ex3 are all preserved quite well in the reconstructed samples.

Quantitative comparisons based on $S_2\,(r)$, $L\,(r)$, and $C_2\,(r)$ are summarized in Table 1. Each number in the first row indicates the root mean squared error (RMSE) between the $S_2\,(r)$ of the exemplar and the $S_2\,(r)$ of the corresponding reconstructed microstructure. Given the small value of RMSEs in $S_2\,(r)$, $L\,(r)$, and $C_2\,(r)$ we can conclude that the statistical equivalency is preserved quite well.

Table 1 also includes the computational costs (last row), indicating that each reconstruction takes roughly 110 min. The primary reasons behind such consistency in costs are the imposed optimization settings on termination criterion (i.e., 1000 iterations) and initialization (optimization starts by a random image as per the flowchart in Fig. 1). The small differences in reconstruction costs are mainly due to the adaptive scheme that the L-BFGS-B algorithm employs to update the variable values while imposing the constraints in Eq. (5).

Lastly, it is noted that a 2D grayscale image requires a one-channel representation, i.e., a total of s^3 brightness values need to be estimated for reconstructing a microstructure of size $s \times s \times s$ voxels (one value for each voxel). However, we use a three-channel (i.e., RGB) representation even for grayscale images where the brightness values are the same across the channels in the exemplar (e.g., a pixel brightness value of 2.34 in a grayscale image is represented as [2.34, 2.34, 2.34] in the corresponding RGB image). Although this modeling choice triples the number of optimization variables, it is adopted primarily because VGG19 is trained on RGB images and does not accept grayscale images. It is highlighted that this additional constraint is *not* added to Eq. (5), that is, the algorithm is expected to reconstruct a 3D microstructure where the RGB values in each voxel are the same.

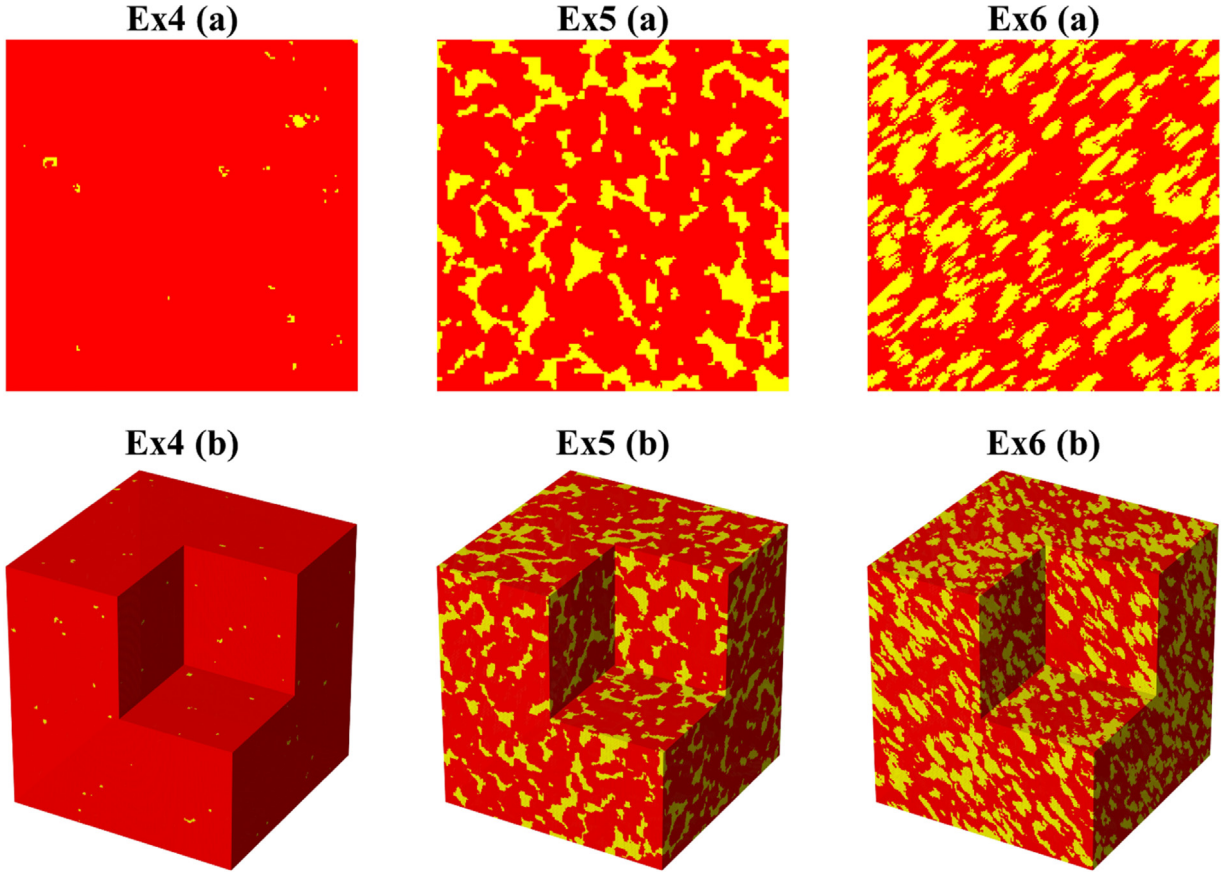

Fig. 7. Examples on binary images: **(a)** The 2D exemplars of size 200×200 pixels, and **(b)** the 3D reconstructed microstructures of size $200 \times 200 \times 200$ voxels. Part of the reconstructed images is removed to reveal the interior. The red and yellow colors distinguish the matrix and the secondary phase.

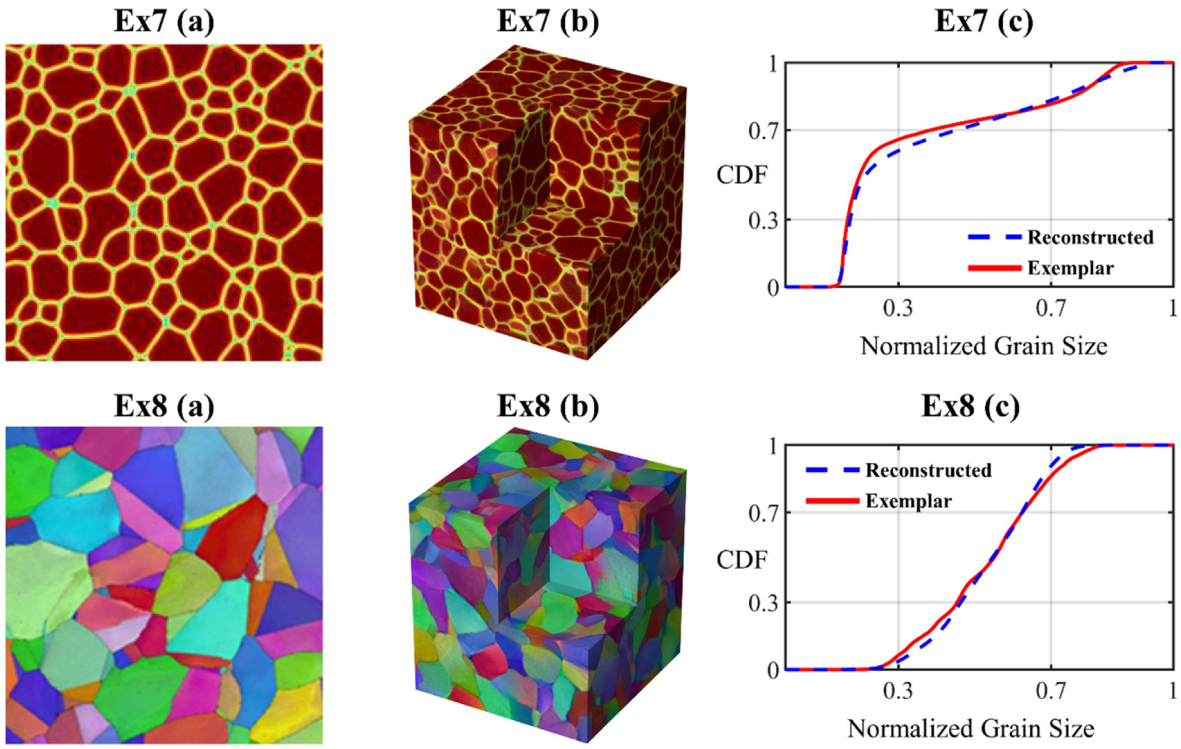

Fig. 8. (Color online) Examples on polycrystalline images: **(a)** The 2D exemplars, **(b)** the 3D reconstructed microstructures, and **(c)** the cumulative distribution function (CDF) vs. normalized grain size. Part of the reconstructed images in **(b)** is removed to reveal the interior.

In the author's experience, such an implementation can consistently satisfy the equality constraint across the RGB channels even though the constraint is not used in optimization.[4]

3.2. Example set 2: Binary images

In grayscale or RGB images, pixel brightness values fall in the continuous region $[0, 255]$. In two-phase microstructures, however, pixels only take two integers (e.g., 0 and 255) which results in sharp gradient changes during optimization. The examples of this subsection demonstrate the robustness of the developed approach to these sharp changes. As illustrated in Fig. 7, the microstructures correspond to drastically different microstructures. Ex4(a) belongs to a dielectric material where the secondary phase represents silicon and controls the dielectric behavior of the system. The area fraction of silicon in Ex4(a) is roughly 1% which makes it very difficult to learn its statistical features. Ex5(a) is the microstructure of ceramics and contains many pores of random shape, size, orientation, and spatial distribution. Ex6(a) demonstrates an anisotropic microstructure where the major axis of the inclusions is aligned diagonally.

The reconstructed microstructures are shown in the second row of Fig. 7, indicating that our method can characterize and reconstruct a wide range of binary images quite well. It is worth noting that Ex4(a) is a nonstationary microstructure so reconstruction methods that rely on stationarity assumptions (e.g., random field-based approaches) are inapplicable to it. Additionally, we highlight that the anisotropy direction in Ex6(a) is along none of the principal directions (i.e., height, width, and thickness) along which the response maps are obtained via VGG19. We believe preservation of these features indicates the generality of our approach.

[4] Adding this constraint causes convergence issues and increases the computation times.

The reconstructed microstructures are quantitatively compared to the exemplars by calculating the RMSEs in $S_2(r)$, $L(r)$, and $C_2(r)$. The results are summarized in Table 1 and indicate that the errors are negligible (in Ex6, these functions are calculated along the anisotropy direction). The computational costs are also provided in Table 1 which are very close to the previous examples.

We highlight that in our examples we treat binary images akin to grayscale images, i.e., we use a three-channel representation where the brightness values (upon optimization) are the same across the RGB channels. However, the brightness values must be integer (as opposed to real numbers in grayscale images) so after optimization, the microstructure is thresholded. The threshold value is not critically important because (assuming the phases are represented with 0 and 255) most brightness values are very close to either 0 or 255. In all the examples in this subsection, a threshold value of 125 is used.

3.3. Example set 3: Polycrystalline images

The last set of examples is on polycrystalline microstructures that contain grains of various shapes, sizes, and orientations. In Fig. 8, Ex7(a) illustrates a 2D microstructure generated via the phase field method that models grain growth [40]. Ex8(a) is obtained by electron backscattered diffraction and demonstrates the microstructure of a polycrystalline material where the colors distinguish different grains and represent unique crystallographic orientations.

The reconstruction results are shown in the second column of Fig. 8 and demonstrate that the essential features of both exemplars are reproduced in the corresponding reconstructed images. Given the importance of grain size, the cumulative distribution function (CDF) of the normalized grain size is used to quantitatively compare the exemplars and the reconstruction results. As shown in the third column of Fig. 8, the CDF of the normalized grain size is preserved quite well. The grain sizes are obtained by

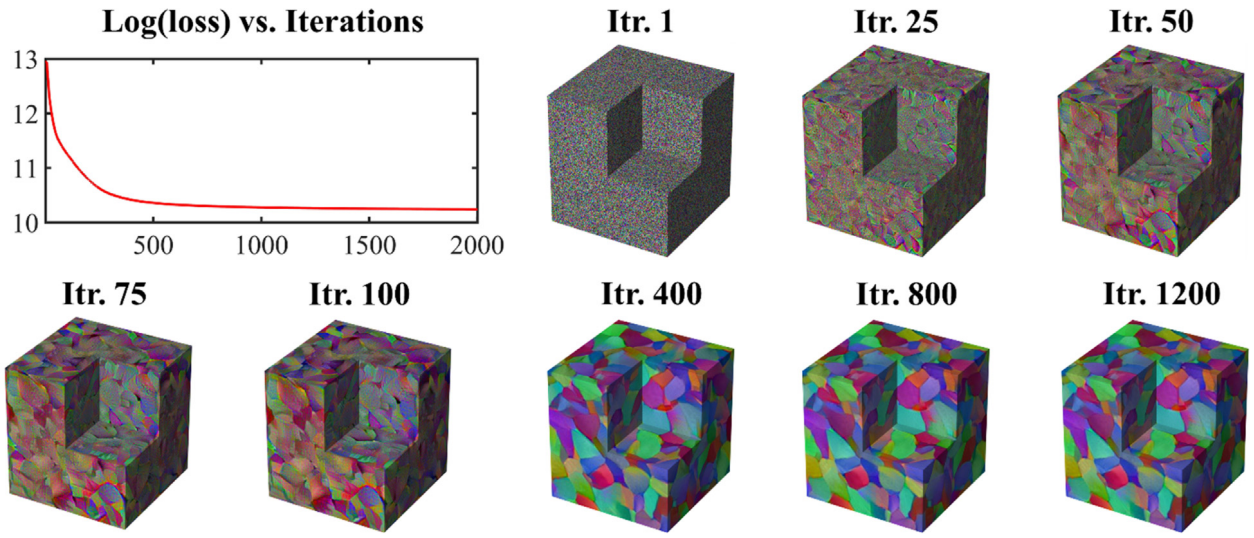

Fig. 9. (Color online) Evolution of the initial 3D image during optimization in Ex8: As the microstructures indicate, the optimization converges with less than 1000 iterations.

counting the number of pixels/voxels within a grain and then the largest grain size (either in terms of area or volume) is used to normalize them.

The computational costs are similar to those of the previous examples: the microstructures in Ex7 and Ex8 are reconstructed in 101.4 and 100.5 min, respectively. The slight reduction in the computational costs is primarily because the gradients of the loss function in Eq. (3) are larger (hence a faster convergence) with an RGB image where the brightness values are different across the channels.

3.4. Discussions

In this subsection, some detailed discussions on convergence and computational costs are provided. Ex8 is used throughout but the discussions are applicable to all the examples.

Fig. 9. illustrates the evolution of the initial random 3D image used in Ex8. As it can be observed, log 10 ($\mathcal{L}(\boldsymbol{x})$) decreases quickly at first but after about 500 iterations the convergence rate drops significantly. This trend is due to the fact that a gradient-based optimization technique is used in the minimization where the gradient norm is large at the early stages of the optimization. A large gradient norm enables L-BFGS-B to take a large step in updating the variables (i.e., $\boldsymbol{x}$) which, in turn, reduces $\mathcal{L}(\boldsymbol{x})$ quickly. As the minimization progresses, this norm reduces and so do the changes in the variables as well as the objective function. Note that Fig. 9 is in log scale to better demonstrate the convergence rate.

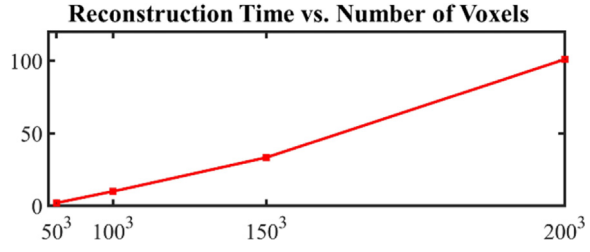

Fig. 10. Effect of image size on reconstruction cost: Time is in minutes and the numbers on the horizontal axis determine the voxel counts. Reconstruction time is reported after 1000 optimization iterations. Each number on the plot is the average value across 5 simulations. The variations are very small and hence not reported.

Fig. 9. also shows that the termination criterion (1000 iterations) used in all the examples is more than sufficient. The advantage of this criterion is that it provides a reliable means to schedule simulations on a server. In particular, the computation time per iteration is almost constant which simplifies resource allocation and scheduling. We highlight that since the optimization problem is a bounded one (see Eq. (5)), the simulation costs of the first few iterations are slightly higher than the rest of the iterations: The random initialization coupled with the large gradient norm may update some of the variables such that they fall outside of the feasible [0, 255] range. Enforcing these constraints increases the computational costs.

The number of iterations as well as the image size directly influence the computational costs. Fig. 10 illustrates the effect of the microstructure size on the costs and indicates that there is almost a linear relation between the number of voxels and the reconstruction time. We believe these costs are acceptable given the robustness (in terms of reproducing the 2D exemplar's features in the 3D reconstructed microstructure) and flexibility (i.e., applicability to various microstructures) of our approach. We recommend some future works to decrease the reconstruction costs in Section 4.

The permutation operator in Fig. 4 is limited to three orthogonal directions which means that the phase value[5] of a voxel is determined by comparing three orthogonal planes centered at that voxel with the given 2D exemplar. This comparison mechanism is schematically illustrated in Fig. 11 where the phase value of voxel V1 is predicted by comparing the three orthogonal blue planes to the given 2D exemplar (note that each plane is partly shown for clarity). Even though we are only using three orthogonal planes, Fig. 7**(c)** indicates that anisotropy is quite well preserved in the non-orthogonal directions. We justify this observation by noting that in each iteration of optimization, the phase values of *all* the voxels are updated simultaneously (as opposed to one voxel at a time employed in [39,59,61,94]). In each update, the relation between any two voxels that do not lie in each other's orthogonal planes is captured through their shared neighboring voxels. For instance, the relation between V1 and V2 in Fig. 11 is captured via the voxels that belong to both the orthogonal planes centered at V1 and the orthogonal planes centered at V2 (the latter planes are not shown in Fig. 11).

[5] The RGB triplet for a colorful image.

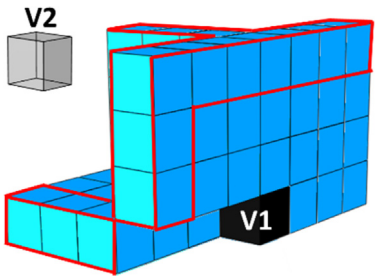

Fig. 11. (Color online) Preserving anisotropy: The phase value of voxel V1 is predicted by comparing the three orthogonal blue planes to the given 2D exemplar (part of each plane is shown for clarity). The spatial relation between V1 and V5 is characterized through their neighboring voxels which belong to both the orthogonal planes centered at V1 and the orthogonal planes centered at V2. Some of these shared voxels are indicated by red lines.

By slightly modifying Eq. (3) we can reconstruct a 3D microstructure by fusing multiple 2D exemplars. For instance, if a 3D cubic sample is imaged on its three orthogonal plans (i.e., *XY*, *YZ*, and *XZ* planes), we fuse the resulting 2D exemplars accordingly to reconstruct a 3D microstructure. In particular, we replace the summation over dimensions (i.e., d in Eq. (3)) with three summations where each one is over one exemplar (one summation for *XY* plane, and so on):

$$\mathcal{L}(\boldsymbol{x}) = \sum_{b=1}^{5} \sum_{l=1}^{l_b} \sum_{p=1}^{3} \sum_{d=1}^{s_b} \sum_{ij} \left(C_{ij}^{dplb} - G_{ij}^{plb} \right)^2 + \lambda p(\boldsymbol{x}), \tag{5}$$

Application of Eq. (5) is demonstrated in Fig. 12 where we use three 2D exemplars from the microstructure in Fig. 12**(a)** to reconstruct the sample in Fig. 12**(b)**. By comparing these two images in terms of their 2D cross sections as well as the CDF of the normalized grain sizes, we can conclude that the statistical equivalency is preserved quite well.

Similar to many other microstructure reconstruction techniques, our method can generate multiple 3D images that are, while visually different, statistically equivalent to a 2D exemplar, see Fig. 13. The 3D microstructures represented throughout Section 3 are representative results, that is, multiple 3D reconstructions of a 2D exemplar achieved very statistics. The reported errors and tun times in Table 1 are averaged over ten simulations.

4. Conclusion and future works

Computational analysis and design of many materials rely on having a 3D sample that embodies the salient microstructural features of the material under study. We introduce a generic, robust, and efficient approach for reconstructing a 3D microstructure using a single 2D exemplar. In our approach, reconstruction is cast as an optimization problem where a random 3D image is iteratively refined to match its statistical features to those of the 2D exemplar. We obtain these features for both the exemplar and the to-be-built 3D microstructure via VGG19 which is an extremely powerful pre-trained deep learning model.

VGG19 can only analyze 2D images with RGB channels. To enable the use of 3D images with VGG19, we augment its structure with a permutation operator located before the first convolution

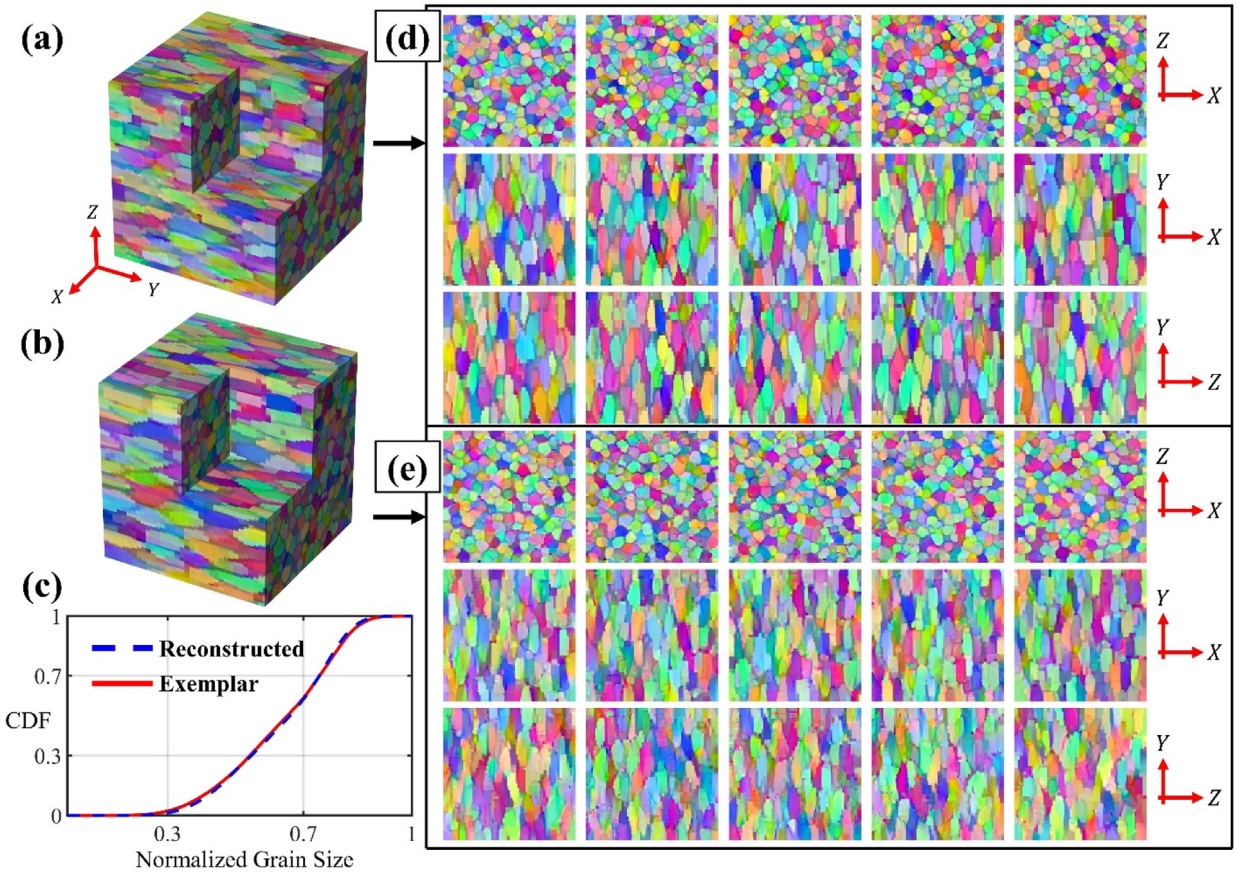

Fig. 12. (Color online) Reconstructing a 3D anisotropic sample via multiple 2D exemplars: **(a)** Original 3D sample. **(b)** Reconstructed sample. Three exemplars (along *XY*, *XZ*, and *YZ* planes) are chosen from (a) to build the microstructure in (b). **(c)** The cumulative distribution function (CDF) vs. normalized grain size for the original and reconstructed microstructures. **(d)** And **(e)** 2D sections from, respectively, the original and reconstructed microstructures. Part of the images in (a) and (b) is removed to reveal the interior.

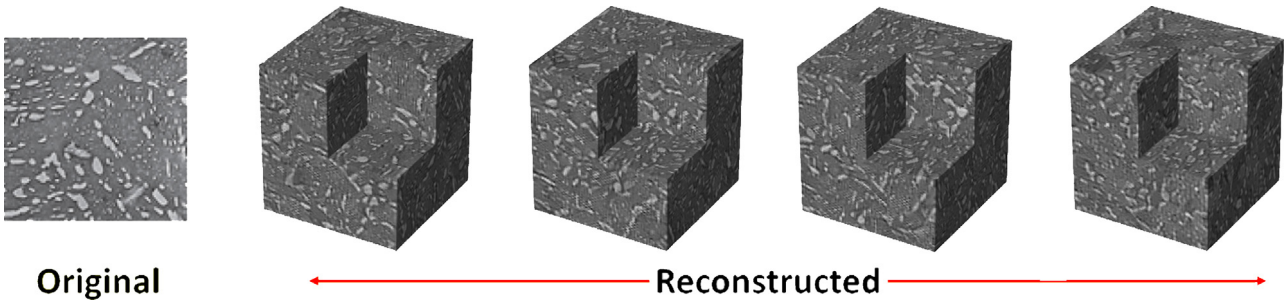

Fig. 13. Reconstruction of multiple statistically equivalent microstructures: All 3D images are generated using the 2D exemplar.

layer. Upon this addition, VGG19 generates correlated 2D features across the 3D microstructure's height, width, and thickness that approximate a 3D feature space. The primary advantages of our simple strategy are threefold. Firstly, it produces 2D features that can be directly compared to the exemplar's 2D features. Secondly, it is not intrusive in that there is no need to re-train VGG19 or alter its internal structure in any way. Thirdly, it is extremely straightforward to implement, adds minimal computational overhead, and is applicable to other deep learning models such as Inception and Xception.

VGG19 produces deterministic features in the form of some 2D response maps. To convert these features to statistical measures and ensure solution existence during optimization, we calculate the Gram matrices associated with these response maps and employ them in formulating the objective function. Since the gradient of the objective function is readily available with standard backpropagation algorithms, the L-BFGS-B method is employed in optimization.

The applications of our method are illustrated with nine examples on different microstructures (alloy, composite, porous, and polycrystalline) and image formats (grayscale, binary, and RGB). The results show that the statistical equivalency is achieved between the reconstructed samples and the corresponding exemplars. We believe these appealing results are primarily due to VGG19's power in detecting features of various forms as well as careful formulation of the objective function that effectively models long- and short-range correlations.

Our approach assumes that any 2D cross-section of the 3D reconstructed sample resembles the 2D exemplar, i.e., its features match (in a statistical sense) to those of the exemplar. As our examples demonstrate, these features can involve rare, non-stationary, or highly correlated statistics. The reconstruction time of our approach primarily depends on size of the microstructure and the number of optimization iterations. Calculating the cost function in Eq. (3) and its gradient with respect to $\boldsymbol{x}$ is $O\left(N^5\right)$ which indicates that the computational complexity of our method is (while similar to texture synthesis) at least an order of magnitude larger than phase recovery. We believe that the generality and accuracy of our approach justify these higher costs. All the examples in Section 3 are cubes of size $200 \times 200 \times 200$ voxels which is sufficiently large for practical applications in, e.g., finite element analysis. The reconstruction time of a cube of size 200^3 voxels is approximately 100 min and decreases almost linearly as the number of voxels is reduced (e.g., the cost is roughly $\frac{1}{8} \times 100$ when reconstructing a cube of size $\left(\frac{1}{2} \times 200\right)^3$). Our analyses and examples demonstrate that the reconstruction cost is reliably predictable given the number of voxels and optimization iterations. This property helps in resource allocation and monitoring the simulations.

We control the convergence by limiting the number of optimization iterations to 1000. As shown in Section 3.4, this is an overkill and a more efficient termination criterion is needed which can also reduce the reconstruction costs. Other means to decrease the costs include choosing an initial 3D image whose features are close to those of the exemplar (instead of a random initial image) and using multiple GPUs in optimization. A multi-resolution reconstruction scheme based on Gaussian pyramids [95] can provide a systematic procedure for implementing these ideas and is the topic of our future work.

The regularization parameter, λ, is the only calibration parameter of our approach. λ directly affects the reconstruction results: a too small λ introduces noise while a very large λ prevents the optimization task from introducing the exemplar's features into the reconstructed microstructure. We propose a very efficient and intuitive way to estimate λ by reconstructing a $2D$ image (which is very fast) for different values of λ and then choosing the best value by visual inspection. In the author's experience, the optimum estimate of λ depends primarily on the deep learning model and how it is used: All our experiments are carried out via VGG19 where all its layers are used for feature extraction (these choices result in $\lambda = 10^{12}$). If another model (or part of VGG19) is used, a different value of λ maybe needed. Regardless of the model and how it is used, our simple approach can be used to estimate λ. A quantitative and reliable approach for estimating λ (either a constant value or one that varies during the optimization) can certainly benefit our approach.

We use the entire convolution layers of VGG19. However, one can choose certain layers (or add weights to them) while building the loss function. This approach is potentially useful if the user is primarily interested in some microstructural features in the exemplar but not all. While such an implementation may provide computational savings and reproduce certain features very efficiently, directly linking individual layers with microstructural features requires some trial and error and will be pursued in our future works.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

A.1. The two-point correlation function

Let $\boldsymbol{X}$ denote a binary microstructure image on a square lattice. The pixel (voxel in 3D) states in $\boldsymbol{X}$ can be identified by:

$$\boldsymbol{X}_{kl} = \begin{cases} 1 & \text{if } kl \in \text{phase } 1 \\ 0 & \text{otherwise,} \end{cases} \tag{6a}$$

where kl is the pixel index and determines its location within the image.

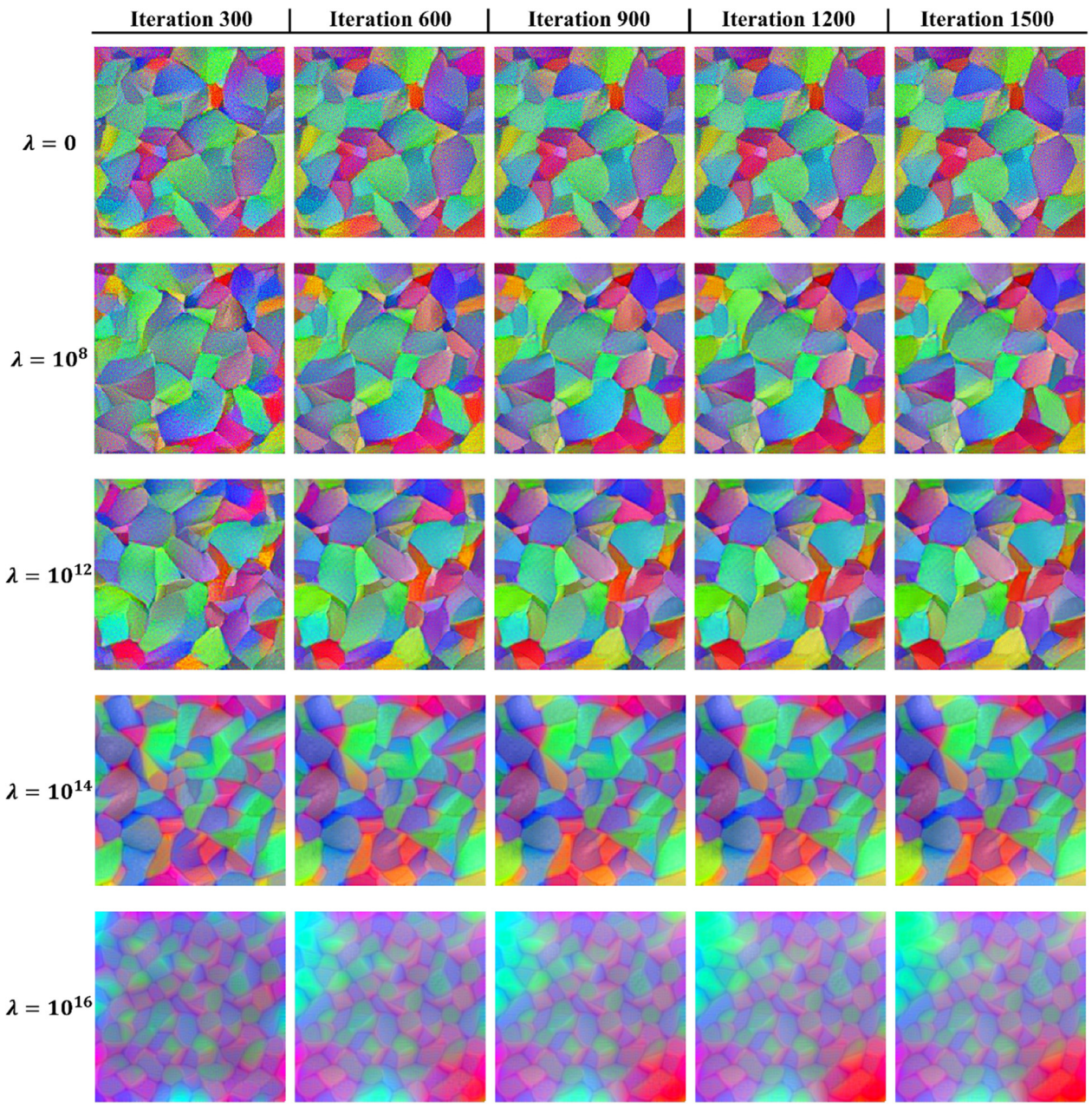

Fig. 14. (Color online) Effect of regularization on reconstruction: A 2D microstructure is reconstructed for each value of λ. The optimization is terminated once the loss function is evaluated 1500 times. The evolution of the microstructure is illustrated every 300 iterations. Reconstruction cost in each case is less than three minutes. The side length of all the images is 200 pixels.

Denoting this location by the vector $\boldsymbol{r}$, the two-point (auto)correlation function [11,12,15] for phase i is defined as:

$$S_2^i(\boldsymbol{r}_1, \boldsymbol{r}_2) = \left\langle \boldsymbol{X}_{\boldsymbol{r}_1}, \boldsymbol{X}_{\boldsymbol{r}_2} \right\rangle, \tag{6b}$$

where $\langle\cdot\rangle$ denotes the expectation operator. If $\boldsymbol{X}$ is stationary and isotropic, S_2^i will only depend on the distance between the two points:

$$S_2^i(\boldsymbol{r}_1, \boldsymbol{r}_2) = S_2^i(\Delta\boldsymbol{r}_{12}) = S_2^i(|\Delta\boldsymbol{r}_{12}|) = S_2^i(r), \tag{6c}$$

Hence, for a stationary and isotropic material, $S_2^i(r)$ has a simplified formulation and can be efficiently calculated via fast Fourier transform (FFT) [44,96,97], Monte Carlo [39], orthogonal sampling [21,22], or lattice-point algorithm [98]. In this paper, FFT is used in calculations and the superscript i is dropped as $S_2^i(r)$ is obtained only for the secondary phase whose spatial distributions are of interest.

A.2. The lineal-path function

The lineal-path function [16], denoted by $L(r)$ for an isotropic and stationary microstructure, quantifies the amount of *clustered-ness* along straight lines within an image. $L^i(r)$ is a two-point quantity and measures the probability of throwing a line on the microstructure image and having the entire line land on phase i.

$L(r)$ can be calculated either with a Monte Carlo approach [39] or by building the histogram of chords [38]. In this paper, the Monte Carlo approach is used.

A.3. Effect of regularization

The regularization parameter, λ, noticeably effects the reconstruction results and should be estimated carefully. Given the relatively high computation cost of reconstructing a large 3D microstructure (see the bottom row Table 1) we recommend to estimate λ by reconstructing a 2D microstructure for various values of λ to find the appropriate value, see Fig. 14. The procedures for such a reconstruction are similar to those outlined in Section 2 except that there is no need for dimension to batch conversion which, in turn, eliminates the summation over d in Eq. (3). As Fig. 14 illustrates, the reconstructed microstructure is noisy when $\lambda = 0$ (i.e., no regularization). As λ increases, the noise reduces and with $\lambda = 10^{12}$ the noise seems to be minimized. Further increasing λ adversely affects the results where first the correlations are lost (e.g., grain size) and then the colors (which represent grain orientation).

In the author's experience, the appropriate value of λ mostly depends on the employed deep learning model (VGG19 in this paper) rather than the microstructure. As noted in Section 3, $\lambda = 10^{12}$ is used throughout this paper. Note that, if a part of VGG19 (e.g., only blocks 1 and 2) is used in reconstruction, the same part should be used to estimate λ.

Another method for estimating λ is to reconstruct a small 3D microstructure for different values of λ and then choose the most appropriate value. Given the success of the approach described above, this estimation method is not pursued in this work.

References

[1] Frazier WE. Metal additive manufacturing: A review. J Mater Eng Perform 2014;23(6):1917–28.
[2] Tian R, et al. A multiresolution continuum simulation of the ductile fracture process. J Mech Phys Solids 2010;58(10):1681–700.
[3] Kwiecien M, Macdonald I, Dullien F. Three-dimensional reconstruction of porous media from serial section data. J Microsc 1990;159(3):343–59.
[4] Bostanabad R, et al. Uncertainty quantification in multiscale simulation of woven fiber composites. Comput Methods Appl Mech Engrg 2018;338:506–32.
[5] Yuan MJ, Turng LS. Microstructure and mechanical properties of microcellular injection molded polyamide-6 nanocomposites. Polymer 2005;46(18):7273–92.
[6] Salvo L, et al. X-ray micro-tomography an attractive characterisation technique in materials science. Nucl Instrum Methods Phys Res B 2003;200:273–86.
[7] Edelman RR, Warach S. Magnetic resonance imaging. New Engl J Med 1993;328(10):708–16.
[8] Kastner J, et al. Advanced X-ray tomographic methods for quantitative characterisation of carbon fibre reinforced polymers. In: Proc. of 4th international symposium on NDT in Aerospace. Augsburg, Deutschland. 2012.
[9] Joshi MY. A class of stochastic models for porous media. University of Kansas; 1974.
[10] Quiblier JA. A new three-dimensional modeling technique for studying porous media. J Colloid Interface Sci 1984;98(1):84–102.
[11] Corson PB. Correlation functions for predicting properties of heterogeneous materials. II. Empirical construction of spatial correlation functions for two-phase solids. J. Appl Phys 1974;45(7):3165–70.
[12] Corson PB. Correlation functions for predicting properties of heterogeneous materials. I. Experimental measurement of spatial correlation functions in multiphase solids. J Appl Phys 1974;45(7):3159–64.
[13] Corson PB. Correlation functions for predicting properties of heterogeneous materials. III. Effective elastic moduli of two-phase solids. J Appl Phys 1974;45(7):3171–9.
[14] Corson PB. Correlation functions for predicting properties of heterogeneous materials. IV. Effective thermal conductivity of two-phase solids. J Appl Phys 1974;45(7):3180–2.
[15] Torquato S, Stell G. Microstructure of two-phase random media. I. The n-point probability functions. J Chem Phys 1982;77(4):2071–7.
[16] Lu B, Torquato S. Lineal-path function for random heterogeneous materials. Phys Rev A 1992;45(2):922.
[17] Quintanilla J, Torquato S. Lineal measures of clustering in overlapping particle systems. Phys Rev E 1996;54(4):4027–36.
[18] Singh H, et al. Image based computations of lineal path probability distributions for microstructure representation. Mater Sci Eng A 2008;474(1–2):104–11.
[19] Turner DM, Niezgoda SR, Kalidindi SR. Efficient computation of the angularly resolved chord length distributions and lineal path functions in large microstructure datasets. Modelling Simulation Mater Sci Eng 2016;24(7). 075002.
[20] Torquato S, Beasley J, Chiew Y. Two-point cluster function for continuum percolation. J Chem Phys 1988;88(10):6540–7.
[21] Yeong C, Torquato S. Reconstructing random media II.three-dimensional media from two-dimensional cuts. Phys Rev E 1998;58(1):224.
[22] Yeong C, Torquato S. Reconstructing random media. Phys Rev E 1998;57(1):495.
[23] Jiao Y, Stillinger FH, Torquato S. A superior descriptor of random textures and its predictive capacity. Proc Natl Acad Sci USA 2009;106(42):17634–9.
[24] Aarts E, Korst J. Simulated annealing and Boltzmann machines. 1988.
[25] Collins BC, Matous K, Rypl D. Three-dimensional reconstruction of statistically optimal unit cells of multimodal particulate composites. Int J Multiscale Comput Eng 2010;8(5):489–507.
[26] Kumar NC, Matous K, Geubelle PH. Reconstruction of periodic unit cells of multimodal random particulate composites using genetic algorithms. Comput Mater Sci 2008;42(2):352–67.
[27] Deb K, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 2002;6(2):182–97.
[28] Jiao Y, Stillinger FH, Torquato S. Modeling heterogeneous materials via two-point correlation functions. II. Algorithmic details and applications. Phys Rev E 2008;77(3 Pt 1):031135.
[29] Tang T, et al. A pixel selection rule based on the number of different-phase neighbours for the simulated annealing reconstruction of sandstone microstructure. J Microsc 2009;234(3):262–8.
[30] Li DS, et al. 3D Reconstruction of carbon nanotube composite microstructure using correlation functions. J Comput Theor Nanosci 2010;7(8):1462–8.
[31] Baniassadi M, et al. Three-phase solid oxide fuel cell anode microstructure realization using two-point correlation functions. Acta Mater 2011;59(1):30–43.
[32] Wu W, Jiang F. Simulated annealing reconstruction and characterization of the three-dimensional microstructure of a LiCoO 2 lithium-ion battery cathode. Mater Charact 2013;80:62–8.
[33] Chen D, et al. Stable-phase method for hierarchical annealing in the reconstruction of porous media images. Phys Rev E 2014;89(1). 013305.
[34] Guo EY, et al. Accurate modeling and reconstruction of three-dimensional percolating filamentary microstructures from two-dimensional micrographs via dilation-erosion method. Mater Charact 2014;89:33–42.
[35] Roberts AP. Statistical reconstruction of three-dimensional porous media from two-dimensional images. Phys Rev E 1997;56(3):3203–12.
[36] Turner DM, Kalidindi SR. Statistical construction of 3-D microstructures from 2-D exemplars collected on oblique sections. Acta Mater 2016;102:136–48.
[37] Torquato S, Lu B. Chord-length distribution function for two-phase random media. Phys Rev E 1993;47(4):2950–3.
[38] Pant LM, Mitra SK, Secanell M. Stochastic reconstruction using multiple correlation functions with different-phase-neighbor-based pixel selection. Phys Rev E 2014;90(2). 023306.
[39] Bostanabad R, et al. Stochastic microstructure characterization and reconstruction via supervised learning. Acta Mater 2016;103:89–102.
[40] Bostanabad R, et al. Computational microstructure characterization and reconstruction: Review of the state-of-the-art techniques. Prog Mater Sci 2018;95:1–41.
[41] Hasanabadi A, et al. 3D microstructural reconstruction of heterogeneous materials from 2D cross sections: A modified phase-recovery algorithm. Comput Mater Sci 2016;111(Supplement C):107–15.
[42] Liu Y, et al. Computational microstructure characterization and reconstruction for stochastic multiscale material design. Comput Aided Des 2013;45(1):65–76.
[43] Bargmann S, et al. Generation of 3D representative volume elements for heterogeneous materials: A review. Prog Mater Sci 2018;96:322–84.
[44] Fullwood DT, Niezgoda SR, Kalidindi SR. Microstructure reconstructions from 2-point statistics using phase-recovery algorithms. Acta Mater 2008;56(5):942–8.

[45] Fullwood D, et al. Gradient-based microstructure reconstructions from distributions using fast fourier transforms. Mater Sci Eng A 2008;494(1):68–72.
[46] Niezgoda SR, Yabansu YC, Kalidindi SR. Understanding and visualizing microstructure and microstructure variance as a stochastic process. Acta Mater 2011;59(16):6387–400.
[47] Niezgoda SR, et al. Optimized structure based representative volume element sets reflecting the ensemble-averaged 2-point statistics. Acta Mater 2010;58(13):4432–45.
[48] Choudhury A, et al. Quantification and classification of microstructures in ternary eutectic alloys using 2-point spatial correlations and principal component analyses. Acta Mater 2016;110:131–41.
[49] Gupta A, et al. Structure-property linkages using a data science approach: Application to a non-metallic inclusion/steel composite system. Acta Mater 2015;91:239–54.
[50] Karasek L, Sumita M. Characterization of dispersion state of filler and polymer-filler interactions in rubber-carbon black composites. J Mater Sci 1996;31(2):281–9.
[51] Xu H, et al. A descriptor-based design methodology for developing heterogeneous microstructural materials system. J Mech Des 2014;136(5). 051007.
[52] Wilson S, et al. Microstructural characterization and evolution in 3D. In: Proc. 31st RisøInt. symp. on materials science. Technical University of Denmark; 2010.
[53] Tewari A, Gokhale A. Nearest-neighbor distances between particles of finite size in three-dimensional uniform random microstructures. Mater Sci Eng A 2004;385(1):332–41.
[54] Holotescu S, Stoian F. Prediction of particle size distribution effects on thermal conductivity of particulate composites. Mater wiss Werkst tech 2011;42(5):379–85.
[55] Sintay S, Rollett A. Testing the accuracy of microstructure reconstruction in three dimensions using phantoms. Modelling Simulation Mater Sci Eng 2012;20(7). 075005.
[56] Rollett AD, et al. Three-dimensional characterization of microstructure by electron back-scatter diffraction. Annu Rev Mater Res 2007;37:627–58.
[57] Breneman CM, et al. Stalking the materials genome: A data-driven approach to the virtual design of nanostructured polymers. Adv Funct Mater 2013;23(46):5746–52.
[58] Torquato S. Statistical description of microstructures. Annu Rev Mater Res 2002;32(1):77–111.
[59] Liu XC, Shapiro V. Random heterogeneous materials via texture synthesis. Comput Mater Sci 2015;99:177–89.
[60] Liu X, Shapiro V. Multiscale shape–material modeling by composition. Comput Aided Des 2018;102:194–203.
[61] Sundararaghavan V. Reconstruction of three-dimensional anisotropic microstructures from two-dimensional micrographs imaged on orthogonal planes. Integr Mater Manuf Innov 2014;3(1):1–11.
[62] Kumar A, et al. A Markov random field approach for microstructure synthesis. Modelling Simulation Mater Sci Eng 2016;24(3). 035015.
[63] Acar P, Sundararaghavan V. A Markov random field approach for modeling spatio-temporal evolution of microstructures. Modelling Simulation Mater Sci Eng 2016;24(7). 075005.
[64] De Bonet JS. Multiresolution sampling procedure for analysis and synthesis of texture images. In: Proceedings of the 24th annual conference on computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co; 1997.
[65] Heeger DJ, Bergen JR. Pyramid-based texture analysis/synthesis. In: Proceedings of the 22nd annual conference on computer graphics and interactive techniques. ACM; 1995.
[66] Zhu SC, Wu YN, Mumford D. Filters random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. Int J Comput Vis 1998;27(2):107–26.
[67] Efros AA, Leung TK. Texture synthesis by non-parametric sampling. In: Computer vision, 1999. The proceedings of the seventh IEEE international conference on. IEEE; 1999.
[68] Efros AA, Freeman WT. Image quilting for texture synthesis and transfer. In: Proceedings of the 28th annual conference on computer graphics and interactive techniques. ACM; 2001.
[69] Wei L-Y, Levoy M. 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; 2000, p. 479–88.
[70] Liu X, Shapiro V. Random heterogeneous materials via texture synthesis. Comput Mater Sci 2015;99:177–89.
[71] Harrison P. A non-hierarchical procedure for re-synthesis of complex textures. 2001.
[72] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014, arXiv preprint arXiv:1409.1556.
[73] Szegedy C, et al. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[74] Chollet F. Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
[75] Cang R, et al. Microstructure representation and reconstruction of heterogeneous materials via deep belief network for computational material design. J Mech Des 2017;139(7). 071404.
[76] . Li X, et al. A transfer learning approach for microstructure reconstruction and structure–property predictions. 2018, arXiv preprint arXiv:1805.02784.
[77] Yang Z, et al. Microstructural materials design via deep adversarial learning methodology. J Mech Des 2018;140(11). 111416.
[78] Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng 2009;22(10):1345–59.
[79] Deng J, et al. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. Ieee; 2009.
[80] Chollet F, python Deeplearningwith. Deep learning with python. Manning Publications Co; 2017.
[81] Gatys LA, Ecker AS, Bethge M. Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks, Vol. 12. 2015, p. 4, arXiv preprint arXiv:1505.07376.
[82] Rudin LI, Osher S, Fatemi E. Nonlinear total variation based noise removal algorithms. Physica D 1992;60(1–4):259–68.
[83] Gatys LA, Ecker AS, Bethge M. A neural algorithm of artistic style. 2015, arXiv preprint arXiv:1508.06576.
[84] Abadi M, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. 2016, arXiv preprint arXiv:1603.04467.
[85] Zhu C, et al. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. ACM Trans Math Softw 1997;23(4):550–60.
[86] Debye P, Bueche AM. Scattering by an inhomogeneous solid. J Appl Phys 1949;20(6):518–25.
[87] Debye P, Anderson Jr H, Brumberger H. Scattering by an inhomogeneous solid. II. The correlation function and its application. J Appl Phys 1957;28(6):679–83.
[88] Xu HY, et al. Descriptor-based methodology for statistical characterization and 3D reconstruction of microstructural materials. Comput Mater Sci 2014;85:206–16.
[89] Ballani F, Stoyan D. Reconstruction of random heterogeneous media. J Microsc 2015;258(3):173–8.
[90] Cang RJ, Ren MY. Deep network-based feature extraction and reconstruction of complex material microstructures. In: Proceedings of the asme international design engineering technical conferences and computers and information in engineering conference, 2016, Vol. 2b, 2016. pp. 95–104.
[91] Izadi H, et al. Application of full set of two point correlation functions from a pair of 2D cut sections for 3D porous media reconstruction. J Petrol Sci Eng 2017;149(Supplement C):789–800.
[92] Tahmasebi P. Accurate modeling and evaluation of microstructures in complex materials. Phys Rev E 2018;97(2). 023307.
[93] Granath O, Wessén M, Cao H. Porosity reduction possibilities in commercial Aluminium A380 and Magnesium AM60 alloy components using the RheoMetalTM process. Metall Sci Tecnol 2010;28(1).
[94] Bostanabad R, Chen W, Apley DW. Characterization and reconstruction of 3D stochastic microstructures via supervised learning. J Microsc 2016;264(3):282–97.
[95] Burt PJ. Fast filter transforms for image-processing. Comput Graph Image Process 1981;16(1):20–51.
[96] Berryman JG. Measurement of spatial correlation-functions using image-processing techniques. J Appl Phys 1985;57(7):2374–84.
[97] Fullwood DT, et al. Gradient-based microstructure reconstructions from distributions using fast Fourier transforms. Mater Sci Eng A 2008;494(1–2):68–72.
[98] Jiao Y, Stillinger FH, Torquato S. Modeling heterogeneous materials via two-point correlation functions: Basic principles. Phys Rev E 2007;76(3 Pt 1). 031110.