Computers & Geosciences 95 (2016) 85–98

Contents lists available at ScienceDirect

Computers & Geosciences

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

Case study

Optimization-based multiple-point geostatistics: A sparse way

Sadegh Kalantari, Mohammad Javad Abdollahifard *

Electrical Engineering Department, Tafresh University, Tafresh, Iran

ARTICLE INFO

Article history:
Received 10 April 2016
Received in revised form
25 June 2016
Accepted 16 July 2016
Available online 19 July 2016

Keywords:
Dictionary learning
Optimization-based synthesis
Texture synthesis
Training image
Sparse representation

ABSTRACT

In multiple-point simulation the image should be synthesized consistent with the given training image and hard conditioning data. Existing sequential simulation methods usually lead to error accumulation which is hardly manageable in future steps. Optimization-based methods are capable of handling inconsistencies by iteratively refining the simulation grid. In this paper, the multiple-point stochastic simulation problem is formulated in an optimization-based framework using a sparse model. Sparse model allows each patch to be constructed as a superposition of a few atoms of a dictionary formed using training patterns, leading to a significant increase in the variability of the patches. To control the creativity of the model, a local histogram matching method is proposed. Furthermore, effective solutions are proposed for different issues arisen in multiple-point simulation. In order to handle hard conditioning data a weighted matching pursuit method is developed in this paper. Moreover, a simple and efficient thresholding method is developed which allows working with categorical variables. The experiments show that the proposed method produces acceptable realizations in terms of pattern reproduction, increases the variability of the realizations, and properly handles numerous conditioning data.

1. Introduction

More than two decades ago, Guardiano and Srivastava (1993) have suggested going beyond bivariate moments to achieve more realistic realizations when simulating environmental variables. Training images (TIs) were introduced as direct models describing the behavior of the field of interest. Since then, different multiple-point statistics (MPS) methods have been proposed aiming at constructing a simulation grid (SG) consistent with the TI and satisfying hard conditioning data. The pattern reproduction capability, as one of the most important factors for comparing different MPS methods, implies that the SG is expected to be composed of patterns similar to those of the TI. Meanwhile, MPS methods are expected to produce diverse realizations to appropriately model the variability of the field (Tan et al., 2014).

Most existing MPS simulation methods are sequential and fill the SG in a specific order. Pixel-based methods fill one pixel at each simulation step and hence are too slow (Strebelle, 2002; Mariethoz et al., 2010; Huang et al., 2013). Patch-based methods fill one patch at a time to accelerate the simulation process, but they have difficulties handling hard conditioning data in large patches (Arpat and Caers, 2007; Honarkhah and Caers, 2010; Mahmud et al., 2014; Abdollahifard, 2016). Error accumulation is the most prominent problem encountered in sequential simulation. Due to the limited spatial extent of data events, the values could be synthesized inconsistently with far pixels. As the simulation proceeds, the accumulation and propagation of such errors lead to short-range unmanageable inconsistencies degrading the pattern reproduction capability (Abdollahifard, 2016).

If the SG is scanned in a raster order, a minimum overlap with previously synthesized pixels is guaranteed, resulting in more consistent realizations (Tahmasebi et al., 2014; Mahmud et al., 2014). However, such methods are also prone to develop inconsistencies with conditioning data. Mahmud et al. (2014) attempted to handle inconsistencies between subsequent patches by allowing the border to be cut along an arbitrarily shaped curve which minimizes the error. It seems easier to handle accumulated errors in textureless areas and/or regions with fewer hard samples. Based on this, Abdollahifard (2016) attempted to prevent inconsistencies by informed selection of points on the scanning path, giving priorities to pixels with more hard neighbors and more explicit structures (stronger edges) and postponing the synthesis of textureless regions.

In sequential simulation methods, the synthesized values have no chance for future refinement. This problem has been overcome using optimization-based methods developed in the context of texture synthesis in computer graphics (Kwatra et al., 2005; Kopf et al., 2007; Peyré, 2009). Optimization-based methods avoid error accumulation by iterative refinement of realizations using an EM-like method. This is achieved at the expense of much further computational effort needed to solve enormous number of template matching problems. Mariethoz and Lefebvre (2014) have

* Corresponding author.
E-mail address: mj.abdollahi@tafreshu.ac.ir (M.J. Abdollahifard).

http://dx.doi.org/10.1016/j.cageo.2016.07.006
0098-3004/

compared texture synthesis and MPS simulation concluding that they have much in common. In fact, MPS is an extended conditional texture synthesis that is applicable to two/three-dimensional fields. Very recently, some researchers attempted to extend the optimization-based methods to geological applications and MPS simulation (Abdollahifard and Ahmadi, 2016; Yang et al., 2016; Pourfard et al., 2016).

Instead of a TI, Abdollahifard and Ahmadi (2016) suggested to employ an analytical edge model allowing the template matching problem to be solved using a few steps of a gradient-descent optimization. However, the method is limited to image reconstruction using high sampling rates (≥1%). Yang et al. (2016) extended the method of Kwatra et al. (2005) to MPS simulation by suggesting solutions for handling conditional data and 3D simulation. Yang et al. (2016) have employed the PatchMatch method of Barnes et al. (2009) to accelerate the search process to some extent.

Although the above-mentioned methods perform very well in terms of pattern reproduction, they have difficulties in producing diverse realizations. Yang et al. (2016) have increased the realizations diversity by enlarging the training data-base, leading to even more computational complexity.

A diverse family of signal frames or image patches could easily be reconstructed as a superposition of a limited number of atoms of a well-chosen dictionary. Such frames or patches are said to have sparse representation in the dictionary space. There exist methods which allow learning a dictionary from a set of exemplar images or signals (Aharon et al., 2006). Peyré (2009) has proposed an optimization-based texture synthesis method whereby, instead of searching a complete TI in the E-step, the image patches were forced to have sparse representation in a dictionary learnt based on training patterns.

Based on the method of Peyré (2009), in this paper a new MPS simulation method is developed. Conditional simulation is accomplished by proposing a new sparse coding method named Weighted Matching Pursuit (WMP), which is an extension for the well-known matching pursuit method (Mallat and Zhang, 1993).

The paper is organized as follows. Section 2 introduces fundamental concepts regarding the sparse model and summarizes the sparse texture synthesis method of Peyré (2009). In the next section the proposed method is described. Section 4 examines the capability of the proposed method in handling different problems in MPS simulation. Finally we conclude in Section 5.

2. Sparse model

Over the past decade, sparse models are widely employed for solving different problems in signal and image processing namely image/signal retrieval, denoising, inpainting, and super-resolution, just to name a few (Bruckstein et al., 2009). Let $\tilde{\boldsymbol{p}}$ denote a $\sqrt{n} \times \sqrt{n}$ patch whose vectorized form is denoted by $\boldsymbol{p} \in \mathcal{R}^n$. $\Phi = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_m] \in \mathcal{R}^{n \times m}$ denotes a dictionary which serves as a representation basis, where $\boldsymbol{\phi}_i$ s are the columns of Φ with a unit norm ($\| \phi_i \|_{\ell_2} = 1$ for $i = 1, \ldots, m$), also known as dictionary atoms ($m \geq n$). $\boldsymbol{p}$ could be represented in the dictionary as follows:

$$\boldsymbol{p} = \Phi\boldsymbol{\alpha} = \sum_{i=1}^{m} \alpha_i \boldsymbol{\phi}_i, \tag{1}$$

where $\boldsymbol{\alpha}$ is the representation of $\boldsymbol{p}$ in the representation domain. The patch $\boldsymbol{p}$ is said to have a sparse representation in Φ if $\boldsymbol{\alpha}$ contains a few (say s) non-zero coefficients, or equivalently, $\boldsymbol{p}$ can be constructed as a superposition of a few atoms of the dictionary.

2.1. Sparse coding

Given $\boldsymbol{p}$, the problem of computing $\boldsymbol{\alpha}$ is called sparse coding. In general, this problem is an underdetermined one with infinite number of possible solutions (note that $m \geq n$). To handle such a problem a model is required to confine the solution space. The sparsity model assumes that the patch has a sparse representation in a well-chosen dictionary. Considering the sparsity assumption, the problem can be formulated as follows:

$$\boldsymbol{\alpha} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \; \| \boldsymbol{p} - \Phi\boldsymbol{\alpha} \|_{\ell_2}^2 \quad \text{subject to} \quad \| \boldsymbol{\alpha} \|_{\ell_0} \leq s \tag{2}$$

where $\|\cdot\|_{\ell_a}$ denotes the ℓ_a norm:

$$\| \boldsymbol{\alpha} \|_{\ell_a} = \sqrt[a]{\sum_{i=1}^{m} |\alpha_i|^a} \tag{3}$$

and $\| \boldsymbol{\alpha} \|_{\ell_0}$ is the number of non-zero coefficients of $\boldsymbol{\alpha}$. Approximate algorithms are developed by either relaxing the problem (replacing the ℓ_0 norm with a ℓ_1 norm) or solving the problem in a greedy manner (Mallat and Zhang, 1993).

The greedy algorithms find one coefficient at a time. Matching Pursuit (MP), as the most fundamental method in this category, starts by finding the atom that best matches with the input patch. To do so, the error is defined as follows:

$$E(i) = \min_{z} \| z\boldsymbol{\phi}_i - \boldsymbol{p} \| \tag{4}$$

where z is a scalar variable. The value of z which minimizes the above error function is obtained as the inner product of the dictionary atom $\boldsymbol{\phi}_i$ and the input patch $\boldsymbol{p}$:

$$z^* = \langle \boldsymbol{\phi}_i, \boldsymbol{p} \rangle. \tag{5}$$

$z^*\boldsymbol{\phi}_i$ is the projection of $\boldsymbol{p}$ on $\boldsymbol{\phi}_i$. The index of the best match atom is obtained as follows:

$$i_0 = \underset{i}{\operatorname{argmin}} \; E(i). \tag{6}$$

Then, the i_0th coefficient in $\boldsymbol{\alpha}$ is updated as $\alpha_{i_0} = z^*$. After that, the residual patch is computed as the difference between the original patch and the projection vector. The algorithm continues by doing the same procedure for the residual vector. The Matching Pursuit algorithm is summarized in Table 1.

2.2. Sparse texture synthesis

Our method is founded on the texture synthesis algorithm of Peyré (2009), which we rapidly summarize in the following. As indicated before, the sparse modeling relies on the assumption

Table 1
Matching Pursuit algorithm.

$\boldsymbol{r} \leftarrow \boldsymbol{p}$, $\boldsymbol{\alpha}^1 \leftarrow \boldsymbol{0}$, $k \leftarrow 1$,
while $\{ \| r \| > \epsilon, k \leq s \}$ **do**
  $E(i) \leftarrow \min_z \| z\boldsymbol{\phi}_i - \boldsymbol{r} \|$,
  $i_0 \leftarrow \operatorname{argmin}_i E(i)$,
  $$\alpha_i^k \leftarrow \begin{cases} \alpha_i^{k-1}: & i \neq i_0 \\ \alpha_i^{k-1} + \langle \boldsymbol{\phi}_i, \boldsymbol{r} \rangle: & i = i_0 \end{cases}$$
  .
  $\boldsymbol{r} \leftarrow \boldsymbol{p} - \Phi\boldsymbol{\alpha}^k$
  $k \leftarrow k + 1$,
end while
$\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha}^{k-1}$