Pattern-based geostatistical simulation using discrete wavelet transformation

Snehamoy Chatterjee[1], and Roussos Dimitrakopoulos[2]

Abstract Pattern-based simulation is a growing area within multi-point geostatistical simulation. The successfulness of any pattern based simulation depends on how efficiently a pattern is retrieved from the pattern database. A discrete wavelet-based algorithm is proposed in this paper; this algorithm assists in reducing the dimensionality of the pattern database and preserves maximum pattern information. Wavelets by construction preserve maximum information through a few coefficients termed scaling coefficients. The scaling coefficients of the wavelet decomposed patterns are used for clustering the pattern database and the k-means clustering algorithm is then applied to classify patterns. The proposed algorithm is validated by simulating conditionally and unconditionally sets of categorical and continuous data and analyzing results. The comparative evaluation with *filtersim* shows that the proposed wavelet-based simulation algorithm performs better in all cases.

1. Introduction

Simulation at spatially correlated continuous and/or categorical variables, such as the geological units and metal grades of mineral deposits or sedimentary facies and pertinent attributes of petroleum reservoirs and water aquifers, is a challenging task. The well-known variogram-based two-point statistical techniques [1-2] are limited in their ability to adequately model spatial complexity [3]. To address the limitations of two-point statistical models,there are a number of multi-point simulation techniques proposed in the literature like *simpat* [4], *snesim* [5], *filtersim* [6] and cumulant-based simulation [7]. In multi-point models, a pattern is defined as a set of values spatially distributed over a given template of spatial locations [4]. During simulation, multi-point conditioning data in the form of a template is compared with patterns of the training image (a geological analogue of what is being modeled) and a pattern is selected from the training image.

A pattern-based simulation algorithm using wavelet analysis is proposed in this paper, and termed as *wavesim*. The pattern database is generated in a manner

[2]Department of Mining Engineering, National Institute of Technology Rourkela 769008, India, snehamoy@gmail.com

[3]COSMO Mine Planning Laboratory, Mining and Materials Engineering, McGill University, Montreal, Canada. Roussos.dimitrakopoulos@mcgill.ca

similar to other mp simulation techniques. The pattern database is classified by using wavelet approximate sub-band coefficients of each pattern. The wavelet approximate sub-band can capture most of the pattern variability, and at the same time reduce the dimensionality of the pattern database. Pattern database classification is performed using the k-means clustering technique. For categorical data simulation, the *ccdf* of the individual prototype class for the central node category of the template is developed using the probability of each individual category within the class; however, for continuous data simulation, a random sample is selected from best match class. For simulation, the similarity of the prototype classes with the conditioning data event is calculated. A random pattern is generated from the developed *ccdf* of the best match class. Unlike *filtersim*, *wavesim* is not generating a random pattern from a class; rather it generates a random pattern from a *ccdf* developed for a class. However, for continuous data, no *ccdf* is generated.

2. Method

Define $ti(u)$ as a value of the training image *ti* where $u \in G_{ti}$ and G_{ti} is the regular Cartesian grid discretizing the training image, $ti_T(u)$ indicates a specific multiple-point vector of $ti(u)$ within a template T centered at node u. That is

$$ti_T(u) = \{ti(u+h_1), ti(u+h_2), \ldots, ti(u+h_\alpha), \ldots, ti(u+h_{n_T})\} \quad (1)$$

where, the h_α vectors are the vectors defining the geometry of the n_T nodes of template T and $\alpha = \{1, 2, \ldots, n_T\}$. The vector $h_1 = 0$ represents the central location u of template T. The pattern database is then obtained by scanning ti using template T and stored the multi-point vectors $ti_T(u)$ in the database For a categorical training image with M categories, the training image is first transformed into M sets of binary values $I_m(u), m = 1, \ldots, M, u \in T$,

$$I_m(u) = \begin{cases} 1, & \text{if u belongs to } m\text{th Category,} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

After generating the patdbT irrespective of using a continuous or a categorical training image, the classification of the pattern database will be performed so that during simulation, instead of searching the entire pattern database (patdbT), only some representative members, for example prototypes of the classes, are compared with the conditioning data event. However, when the template

dimension is large, the dimension of patdbT will also be large. Therefore, classification of this large dimensional pattern database patdbT is a computationally demanding task. In previous research, the patdbT classification was performed by reducing the dimensions of the pattern by using few filter scores [6]. Any dimensional pattern in the patdbT is represented by 6*M filter scores (for a two-dimensional image) where M is the number of categories (M =1 for continuous image) present in the training image. A wavelet-based representation of patterns is introduced where the dimension of the pattern-for-pattern classification can be reduced by selecting the scale of wavelet decomposition.

Wavelets analysis can decompose a training image into different frequency components [8]. The wavelet decomposition of an image provides one approximate sub-band image and three high frequency sub-band images after one scale decomposition of a two-dimensional training image. For further decomposition, the approximate image is decomposed to obtain the next scale sub-band images. The approximate sub-band provides average type information about the training image and preserves most of the data variability of the training image. If the high frequency sub-bands are added to the approximate sub-band, then the training image is perfectly reconstructed. It is noted that the amount of data in an approximate sub-band is 2^{j*d} times less than the amount of data in the training image, where *j* is the number of scale in wavelet decomposition, and *d* is the dimension of the original image.

For classification of pattern database patdbT, the approximate sub-band of the patterns, which is reduced in dimension depending on the value of j, is used. The k-means clustering technique [9] is applied to classify the pattern database patdbT. The main idea of k-means clustering is to divide the patdbT into a number of classes such that the sum of the inter-class distance is maximized. The k-means clustering is a simple, unsupervised learning algorithm. In this algorithm, the pattern database is classified based on the selected priory cluster number (k). First, k patterns from the patdbT are randomly selected. These k patterns represent the initial class centroids. Since the patdbT classification is performed by using the approximate sub-band of patterns, randomly selected approximate sub-band of k patterns from patdbT will act as initial centroids. Then each pattern from the patdbT is assigned to a class which has the closest distance to the centroids. After assigning all patterns into any one of those classes, the centroids' positions are recalculated. This is an iterative process and the algorithm stops when the centroids' positions are no longer changed.

After classifying the patdbT prototype calculation, simulation was carried out. During simulation, the similarity between the conditioning data event and the prototypes of the classes is carried out. A sequential simulation algorithm [1] is used for pattern-based simulation in this paper. At each visited node, a conditioning data event is obtained by placing the same template used in the training image, centering at the node to be simulated. The similarity between the conditioning data and prototypes of classes are calculated by a distance function.

A distance function is used to calculate the distance from the prototypes of classes to the conditioning data event. The distance function used in this paper is L_2-norm chosen for its success in template matching.

After measuring the similarities of the conditioning data event with the prototypes of classes, the best matching class is selected. In *filtersim*, a random pattern from the selected class is drawn and pasted in a simulated node. The probability of the central node categories within a class may be different, which has not been considered in *filtersim*. However in *wavesim*, a conditional cumulative distribution function (*ccdf*) is generated for each class. This is developed by calculating the probability of occurrence of a particular category in the central template node, divided by the total number of patterns in that class.

During the simulation process, after finding the best matched class, a uniform random number is generated. From the developed *ccdf*, the category at the central node corresponding to the generated random number is obtained. Then, a random pattern is drawn from the matched class patterns which have the same central node category as the category obtained from the *ccdf*. After pasting the drawn pattern at a simulated node, the next node is visited in a random path. The same distance function and the patterns-drawing algorithm are performed until all nodes are simulated. The algorithm stops when no nodes are left unvisited. It is noted that, for continuous image, a random pattern is drawn from a class; no *ccdf* is generated for the continuous case.

2.1 Application of the proposed method

The *wavesim* algorithm is validated by simulating known categorical and continuous two-dimensional data sets. The exhaustive data sets are obtained from different sources. All runs are performed on a 3.2 Ghz Intel(R) Xeon (TM) PC with 2 GB of RAM. For wavelet decomposition, the Haar basis functions are applied for all cases unless otherwise specified. The results of *wavesim* are compared with *filtersim* results to make a valid comparison.

To perform unconditional simulation, binary training image is considered. For unconditional simulation of categorical image, the wavelet decomposition is performed after generating the pattern database to reduce the dimensionality of the pattern database. The training image is presented in Fig. 1(a). This training image represents complex channels presents in a deposit. The template size is selected using the method proposed in [10] and it is 9×9. The patterns are extracted from the training image and up to 4 scale wavelet decomposition was performed. The unconditional simulation is then performed and compared to the results obtained from *filtersim*. The parameters used for the simulations from the *wavesim* and *filtersim* are the same. Note that the inner patch size is 5×5. The *k*-means clustering algorithm is used with number of classes at 100. An example of unconditionally generated realizations using *wavesim* and *filtersim* are presented in Fig. 1. It is observed from the figure that the *wavesim* can reproduce channels

present in the training image. On the other hand, *filtersim* fails to reproduce the continuity of the channels. The main difference between the *wavesim* and *filtersim* is the way of classifying the patterns in patterns database. The example shows that when classifying patterns using only few filter scores it is not always possible to capture the complexity present in the available patterns; resulting in discontinuities of the channels when unconditional simulations are performed.

The Stanford V Reservoir Data Set [11] is used for the conditional simulation example. One slice of the three-dimensional reservoir data is used as a reference image from which conditioning data are sampled. Another slice is used as the training image. The size of the domain to be simulated is 100×128. The reference image to be simulated is presented in Fig. 2(a). The hard data set consists of 100 data at irregular spacing scattered all over the domain (Fig. 2(b)). The training image used in this study is presented in Fig. 2(c). The *k*-means clustering with cluster number 300 is used for training pattern classification. The weights of hard data, previously simulated node point, and patch data are 0.5, 0.3, and 0.2, respectively, for distance calculation. When the conditioning data set is fully informed, only approximate sub-band coefficients after wavelet decomposition are used. The conditionally simulated realizations generated by *wavesim* and *filtersim* are presented in Figus 2(d) and 2(e). The realizations show that the high valued channels are well reproduced. The comparison study with *filtersim* realizations shows that the channels continuity is well reproduced using *wavesim* as compared to *filtersim*. The histogram and variogram of the simulated realizations are compared with the data histogram and variogram. The results revealed that the first- and second-order statistics are well reproduced using *wavesim*.

A contributor to the success of a simulation algorithm, in terms of use for real world applications, is its computational efficiency. The proposed algorithm is implemented in the MATLAB environment, which makes it difficult to compare the CPU time taken in our various examples to *filtersim* or other algorithm which are implemented in the C++ environment. The main different between our proposed algorithm and *filtersim* is the dimensionality reduction. Both the algorithms are using the same clustering algorithm and simulation steps are almost the same. The computing time depends on the number of reduced dimensions.

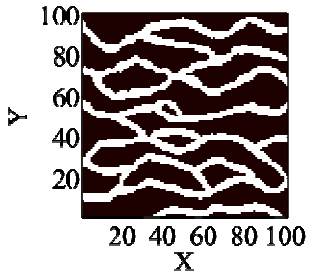
3. Conclusions

A pattern-based conditional simulation algorithm, *wavesim*, is presented. The algorithm uses wavelet basis function for dimensional reduction of patterns. The technique is based on pattern classification and pattern matching; the dimensional reductions of the patterns were performed by wavelet decomposition. The pattern classification was performed by the *k*-means clustering algorithm. The algorithm is verified by two-dimensional conditional and unconditional simulation using different data sets like two-class categorical data and continuous complex channels data. The algorithm reproduced the continuity of the channels using

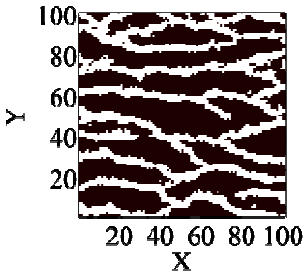
conditional and unconditional simulation. The comparative study with the *filtersim* algorithm showed that the *wavesim* performed better than the *filtersim* for reproducing the continuity of the channels for all examples.

The major advantages of the *wavesim* algorithm are (a) the pattern classification of the high dimensional pattern database can be performed successfully with less computational effort due to the nature of the approximate sub-band of the wavelet decomposition, which reduces the dimensionality of the pattern and captures most of the data variability; and (b) since the *ccdf* is developed for each class for categorical simulation, the pattern drawing from a class is performed based on a probability law, rather than random drawing, which may help with the reproduction of channels better.

(a) Training image



(b) wavesim # Realization 1



(c) filtersim # Realization 1

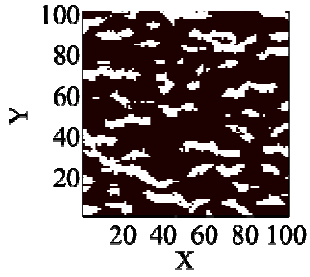
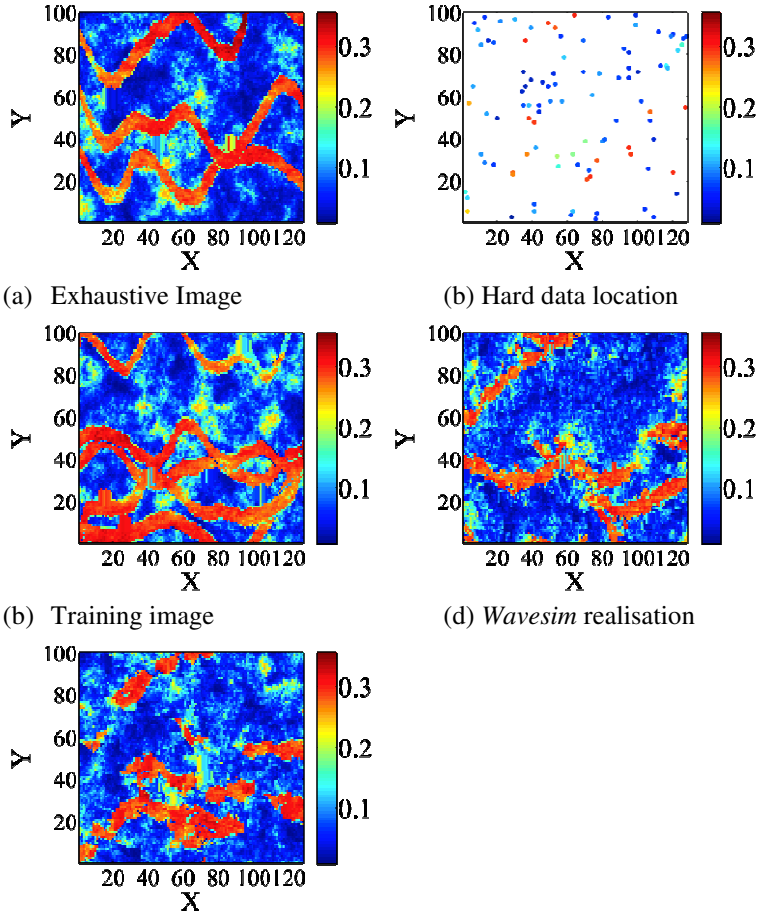


Fig. 1 Training image and unconditionally simulated realisations of the proposed method and *filtersim*



(a) Exhaustive Image (b) Hard data location

(b) Training image (d) *Wavesim* realisation

(e) *Filtersim* realisation

Fig. 2 Exhaustive image, hard data locations, training image, and simulated conditional realisations of our proposed method and *filtersim*

Acknowledgements

The work in this paper was funded from NSERC Discovery Grant 239019 and the members of McGill's COSMO Lab, AngloGold Ashanti, Barrick Gold, BHP Billiton, De Beers, Newmont Mining, and Vale.

Bibliography

1. Goovaerts P (1997) Geostatistics for natural resources evaluation. Oxford University Press, New York

2. Deutsch CV, Journel A (1998) GSLIB: Geostatistical software library and user's guide. Oxford University Press, New York

3. Journel A, Alabert F (1989) Non-Gaussian data expansion in the earth sciences. Terra Nova 1:123–134

4. Arpat G, Caers J (2007) Conditional simulation with patterns. Math Geol 39(2):177-203

5. Strebelle S (2002) Conditional simulation of complex geological structures using multiplepoint statistics.. Math Geol 34 (1): 1-21

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

7. Mustapha H, Dimitrakopoulos R, High-order stochastic simulations for complex non-Gaussian and non-linear geological patterns. Math Geosci 42 (5): 457-485

8. Mallat S A Wavelet Tour of Signal Processing., Academic Press, San Diego

9. MacQueen JB (1967) Some methods for classification and analysis of multivariate observations. In Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability 1: 281-297

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

11. Mao S, Journel A (1999) Generation of a reference petrophysical and seismic three-dimensional data set: The Stanford V reservoir. Stanford Center for Reservoir Forecasting Annual Meeting. Available at: http://ekofisk.stanford.edu/SCRF.html