

Modified FILTERSIM Algorithm for Unconditional Simulation of Complex Spatial Geological Structures

Peyman Mohammadmoradi, Mohammadreza Rasaei*

Institute of Petroleum Engineering, School of Chemical Engineering,
College of Engineering, University of Tehran, Tehran, Iran
Email: *mrasaei@ut.ac.ir

Received May 10, 2012; revised June 13, 2012; accepted June 21, 2012

ABSTRACT

Facies and fracture network modeling need robust, realistic and multi scale methods that can extract and reproduce complex relations in geological structures. Multi Point Statistic (MPS) algorithms can be used to model these high order relations from a visually and statistically explicit model, a training image. FILTERSIM as a pattern based MPS method attracts much attention. It decreases the complexity of computation, accelerates search process and increases CPU performance compare to other MPS methods by transferring training image patterns to a lower dimensional space. The results quality is not however as satisfactory. This work presents an improved version of FILTERSIM in which pattern extraction, persisting and pasting steps are modified to enhance visual quality and structures continuity in the realizations. Examples shown in this paper give visual appealing results for the reconstruction of stationary complex structures.

Keywords: FILTERAIM; MPS; Pattern Persisting; Training Image; Entropy Plot

1. Introduction

Static modeling of complex reservoirs calls for more than two point statistics. Certain features of these objects such as curvilinearity cannot be expressed via two-point relations [1]. Multi Point Statistics (MPS) is used to produce realistic realizations of complex structures after the pioneer work of Strebelle who applied image processing concepts for this purpose [2]. MPS based on image processing techniques and proposes image construction approach. In this approach, the patterns of the final image to be constructed (facies, fracture network...) are obtained from a training image that depicts relevant geological patterns expected to be found in the subsurface [1]. Furthermore, MPS modeling has the potential to improve estimation in addition to simulation and the so called E-type average value at each node is taken as a local estimated value if training image is appropriate [3].

Training Images (TIs) provide spatial relations, existing patterns and degree of connectivity which must be reproduced in the reconstructed images. The goal is not to reproduce the training image, but to simulate a model that shares some of the multivariate characteristics of the true distribution [4]. Achieve the most appropriate TI rise several challenges like: statistical richness of the training image, the scale of the training image, the grid definition

of the simulated model, and the univariate distribution of the training image. In this area, Boisvert selects the most data representative TI by comparing the distribution of runs and the multiple point density function from TIs [5]. Boogaart applied variography as an auxiliary element to choose TIs which represent the true joint distribution of the field under consideration [6].

Pattern to point SNESIM method is the first algorithm for MPS modeling [7,8]. SNESIM is a combination of traditional sequential simulation with a search tree concept. Multiple point statistics (observed patterns) of a certain size of training images are stored in a tree structure; Node properties of realizations are then assigned one by one in a search process loop. Search tree is extremely memory demanding. Straubhaar proposed a list approach modification of SNESIM [9]. Several case studies of successful application of the method are reported [10-14]. Okabe also modified SNESIM for pore network reconstruction [15].

Pattern to pattern methods such as SIMPAT [1] FILTERSIM [16] and DisPAT [17] are completely isolated from two-point statistics and eliminate the probabilistic paradigm in MPS algorithms. These methods inherently take the probabilities of the whole multiple point patterns conditioned to the same multiple point data event from the training image [18]. SIMPAT produces visual appealing realizations; however complete search in pattern

*Corresponding author.

database is needed. This makes the process of finding the most similar pattern too complex and CPU time consuming. FILTERSIM reduces pattern dimensions by applying spatial filters and transferring patterns to a lower dimensional space. Coded patterns are then clustered and a prototype is chosen for each bin. This accelerates the search process and decreases run time compared to SIMPAT by means of reducing number of analogy loops. Structures continuity in FILTERSIM method however is not preserved as it is in SIMPAT.

Dimensional reduction in patterns is done by applying directional filters [16] or wavelet decomposition on training image [19]. Honarkhah proposed a distance based algorithm DisPAT and automatic information theory based algorithms for selection of optimum template size and dimension of patterns target space [17].

In this paper, we proposed an improved version of FILTERSIM method. Modifications are made to the template size and pattern selection algorithm so that features connectivity enhance in realizations. Although computation cost increases marginally compare to the original method, results quality is far better in our Modified FILTERSIM approach.

2. Training Image Processing

Processing of training images consists of gridding, scanning and persisting the corresponding multiple point vectors in a database as shown in **Figure 1**. Pattern extraction can be performed by templates in different sizes. Template size affects the extracted statistics, CPU time and scale of structures in results. A pattern of training image is defined as $ti(u)$ and includes a specific multiple point vector of $ti(u)$ values within a template T centered at node u (Equation (1)).

where h_α vectors define the geometry of the template nodes; $\alpha = 1, \dots, n_T$ and n_T is the template size. Content of pattern with a *frequency component*, based on its occurrence frequency in TI, are stored in pattern database $patdb_T$ as the following vector (Equation (2)):

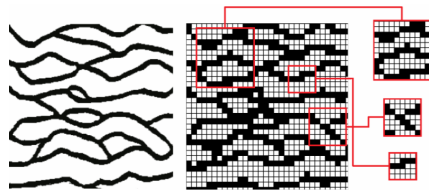


Figure 1. Schematic of training image preprocessing steps consist of gridding, scanning and pattern extraction (As template size increases, pattern stores larger scale spatial information).

Frequency component of $patdb_T$ has the advantage of accelerating selection of appropriate pattern in simulation process. This achieves by elimination of repeated patterns and consequently repeated calculations. Pattern stores larger scale spatial information as the template size increases with the cost of simulation uncertainty and complexity. To find the optimum template size in a single grid simulation, Shannon's entropy versus template size plot is used [20]. In order to calculate entropy in different template sizes, the following equation is applied:

$$H = \sum_{i=1}^K p_i \log(p_i) \quad (3)$$

where K = number of possible outcomes of the random variable, and p_i represents the probability mass function. Two different behaviors are expected with increasing the template size. In the first stage, entropy will sharply increase since the average number of nodes needed to guess the remaining nodes in the patterns is increased. At a later stage, where the template size has passed the optimal range in an elbow that represents the stationary features of the training image, entropy would increase slowly. This is because the information needed for encoding a large pattern with some stationary features is approximately equal the information of the stationary feature of the training image itself [17]. Mathematical details are discussed extensively by Honarkhah [17]. As shown in results of FILTERSIM automation in **Figure 2**, spatial statistics of the results are fixed with the passing of the Elbow in entropy plot.

By applying frequency component and optimum template size concepts, training image is scanned with the optimum template after which the responses of patterns

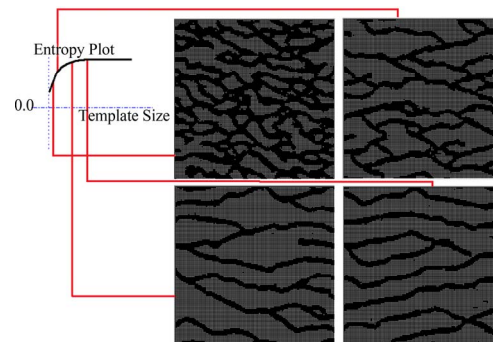


Figure 2. Effect of template size on simulation results (where the template size has passed the optimal window, stationary features in realizations are approximately equal the information of the stationary feature of the training image).

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

$$pat_T^k = \{pat_T^k(h_1), pat_T^k(h_2), \dots, pat_T^k(h_\alpha), \dots, pat_T^k(h_{n_T}), \text{Frequency}^k\} \quad (2)$$

to six directional filters are measured. These filters are listed in **Table 1**. Each filter score is divided into five categories in order to cluster the patterns into 5^6 bins. Many of these created bins are empty and will be eliminated in the search process which is only done on selected prototypes for the filled bins. Each prototype is binary average pattern of patterns in a specific bin.

Table 2 shows the TI processing steps of our Modified FILTERSIM approach.

Having processed the TI and filled the pattern database as just described, reconstruction of realizations can be performed in unconditional and conditional situations. Unconditional simulation procedure is discussed in detail in the next section.

3. Unconditional Simulation Algorithm

Simulation begins by defining a random path on realization nodes. Data event in the template is extracted in each node and its similarity to the existing patterns is then measured. Data event vector $dev_T(u)$ is a set of previously informed nodes in unconditional simulation while hard data is also added to $dev_T(u)$ in conditional simulation. We used the following single point manhattan distance function:

$$d\langle x, y \rangle = \sum_{i=0}^{nr} d_i(x_i, y_i), d_i(x_i, y_i) = |x_\alpha - y_\alpha| \quad (4)$$

Having the random path defined, a search among the prototypes is done for each visited node to select the most similar prototype which shows minimum distance function $d\langle x, y \rangle$ with data event as below:

$$d\langle dev_T(u), Prototype_T^k(h_\alpha) \rangle = \sum_{\alpha=0}^{nr} d_\alpha \langle dev_T(u + h_\alpha), Prototype_T^k(h_\alpha) \rangle \quad (5)$$

The bin of selected prototype contains the most similar patterns to the visited data event. FILTERSIM routine is to select a random pattern among these patterns and set it as realization node. This obviously is not the best procedure and may lose the feature connectivity as is a known

$$d\langle dev_T(u), pat_T^k \rangle = \sum_{\alpha=0}^{nr} d_\alpha \langle dev_T(u + h_\alpha), pat_T^k(h_\alpha) \rangle \quad (6)$$

Only the pattern nodes with known values are considered in the above relation.

$$d\langle x_\alpha, y_\alpha \rangle = \begin{cases} |x_\alpha - y_\alpha| & \text{if } x_\alpha \text{ and } y_\alpha \text{ are both known} \\ 0 & \text{if } x_\alpha \text{ or } y_\alpha \text{ is unknown} \end{cases} \quad (7)$$

The pattern with minimum distance function is then pasted entirely on realization nodes. If some patterns have the same distance to the data event vector, the one with the most frequency is selected. Random selection of patterns is only allowed when they all have the same frequency. This pattern selection strategy has improved the FILTERSIM performance and connectivity features of reconstructed realizations compare to the original method. This will be fully explored in the results section. **Table 3** shows the stepwise procedure of Modified FIL-

Table 1. Directional filters for dimension reduction of patterns [16].

Filter Equation	Name
$f_1(u, v) = 1 - \frac{ v }{n}, v = -n, \dots, n$	N-S directional average
$f_2(u, v) = 1 - \frac{ u }{n}, u = -n, \dots, n$	E-W directional average
$f_3(u, v) = \frac{v}{n}$	N-S directional gradient
$f_4(u, v) = \frac{u}{n}$	E-W directional gradient
$f_5(u, v) = 2 \frac{ v }{n} - 1$	N-S directional curvature
$f_6(u, v) = 2 \frac{ u }{n} - 1$	E-W directional curvature

Table 2. Training image preprocessing steps of the Modified FILTERSIM approach.

Training Image Processing	
1.	Load and mesh training image to the desired grid network size G_n .
2.	Find optimum template size (Entropy plot).
3.	Scan TI and store all patterns in <i>patdb</i> along with the frequency of each pattern as the last component of pat_T^k vector.
4.	Apply filters to patterns and calculate responses.
5.	Cluster patterns according to node scores.
6.	Eliminate empty bins.
7.	Select a prototype for each filled bin.

drawback of this method. Here we modified the procedure and performed a second searching step to find the most similar pattern to data event in the selected bin (**Figure 3**). Pattern selection in candidate cluster is based on the following distance function:

TERSIM approach described above.

The last but not the least point is the updatable region of pasted pattern on realization nodes which is used for the first time by Arpat in SIMAPT [1]. The selected pattern after imposing on the visited node of realization is not entirely fixed and some outer parts of it can be replaced by future pasted patterns as shown in part 4 of **Table 3**. This provides the opportunity of superior realizations with closer features of connectivity to those of TI. (**Figure 4**).

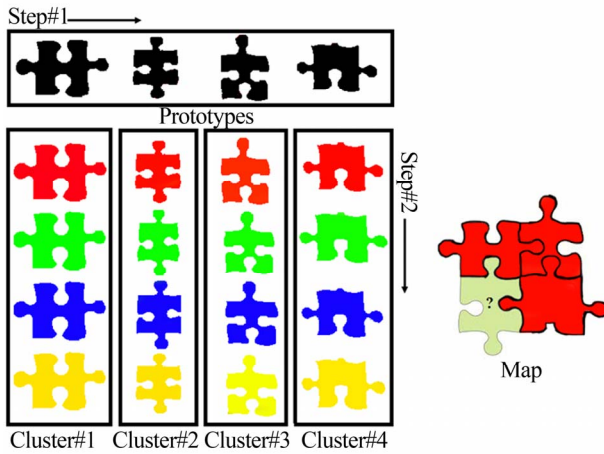


Figure 3. Schematic of searching between prototypes in Step#1 and between selected cluster patterns in Step#2 (In Modified FILTERSIM , Step#1 choose 2nd column and Step#2 chooses the red pattern while in the FILTERIM Step#2 is performed in random).

Table 3. Procedure steps of unconditional Modified FILTERSIM approach.

Unconditional Modified FILTERSIM
<ol style="list-style-type: none"> 1. Training image preprocessing. 2. Define a random path on G_{re} grid of realization re. 3. At every node u, extract the data event $dev_T(u)$ from realization re and find the most similar prototype that minimizes $d(dev_T(u), Prototype_T^k)$. <p>Once the most similar bin is found, search most similar pattern in that bin to maximize similarity $s(dev_T(u), pat_T^k)$, if some patterns have the same distance to the data event vector, most frequent of them is selected and if all of them have the same frequency, selection will be done randomly.</p> <ol style="list-style-type: none"> 4. Paste the entire selected pattern to updatable data event: $dev_T(u+h_u) = pat_T^k(h_u)$. Set inner nodes as fixed and outer nodes as updateable. <ol style="list-style-type: none"> 5. Move to the next updatable node of the random path and repeat the above steps until all grid nodes are fixed.

4. Results and Discussion

Modifications are made to the original FILTERSIM algorithm in several ways of optimizing template size, considering pattern frequency component in database and pattern distance function ranking in candidate bin instead of random selection. Performance of this modified algorithm is investigated in this section by applying the method on several binary 2D training images with grid

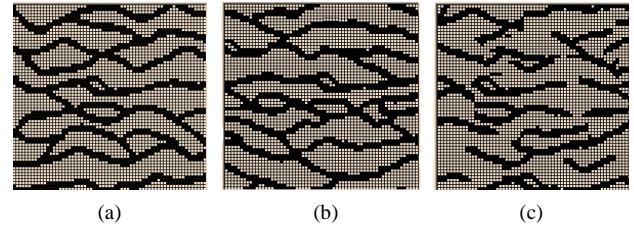


Figure 4. Effect of updatable nodes on structures continuity. (a) Training image; (b) Realization with updatable region; (c) Realization without updatable region.

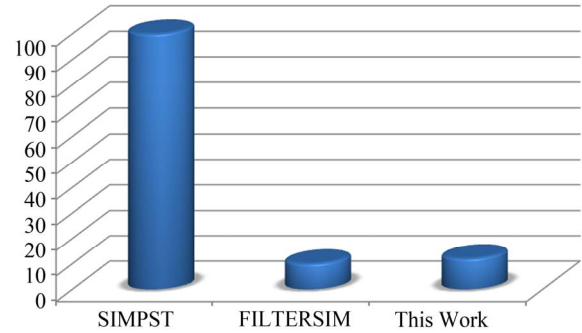


Figure 5. CPU Time Qualitative Comparison of SIMPAT, FILTERSIM and Modified FILTERSIM methods.

size of 128×128 . **Figures 6 and 7** compare reconstructed realizations of 2 different TIs at 3 template sizes with unconditional algorithms of FILTERSIM, SIMPAT and Modified FILTERSIM. Better visual continuity of the Modified FILTERSIM results is clear in all these cases. Entropy changes of the two TIs at various template sizes are shown in **Figure 8**. Elbows are around the optimum template size which is consistent with the superior results of these sizes in **Figures 6 to 7**. For example, for the first training image in **Figure 6**, the best template size of 13×13 is obtained from entropy plot (**Figure 8**). This template size preserves the TI features more precisely in the generated realization compare to other template sizes of 9×9 and 17×17 as shown in **Figure 6**.

Performance of modified FILTERSIM is also investigated by applying the method for a continuous 2D training image with grid size of 128×128 (**Figure 9**). In this case better visual continuity of our algorithm is clear which is obtained the optimum template size of 15.

5. Summary

We proposed a Modified FILTERSIM algorithm to simulate 2D binary stationary structures. We modified the original algorithm with introducing optimum template size for each training image, a pattern frequency parameter in database, and additional search among stored patterns in the selected bin. Shannon entropy concept was applied in FILTERSIM and used to infer optimum template size that can capture local structures in each template. Frequency

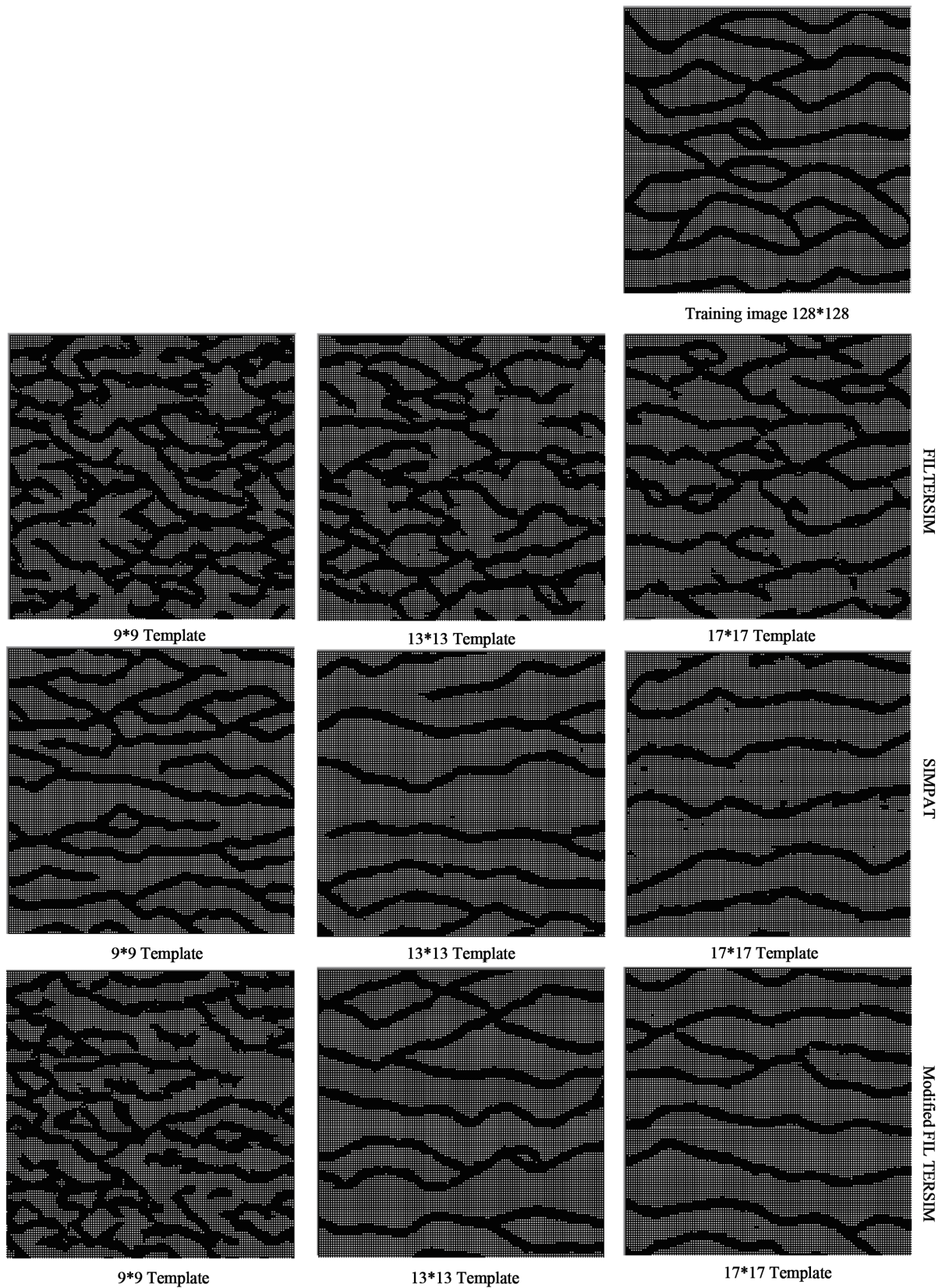


Figure 6. Simulation results with three different template sizes by applying SIMPAT, FILTERSIM and Modified FILTERSIM methods.

	Actual	FIL TERSIM	Modified FIL TERSIM
Porosity %			
Average of 25 realizations	13.64	17.4	13.61

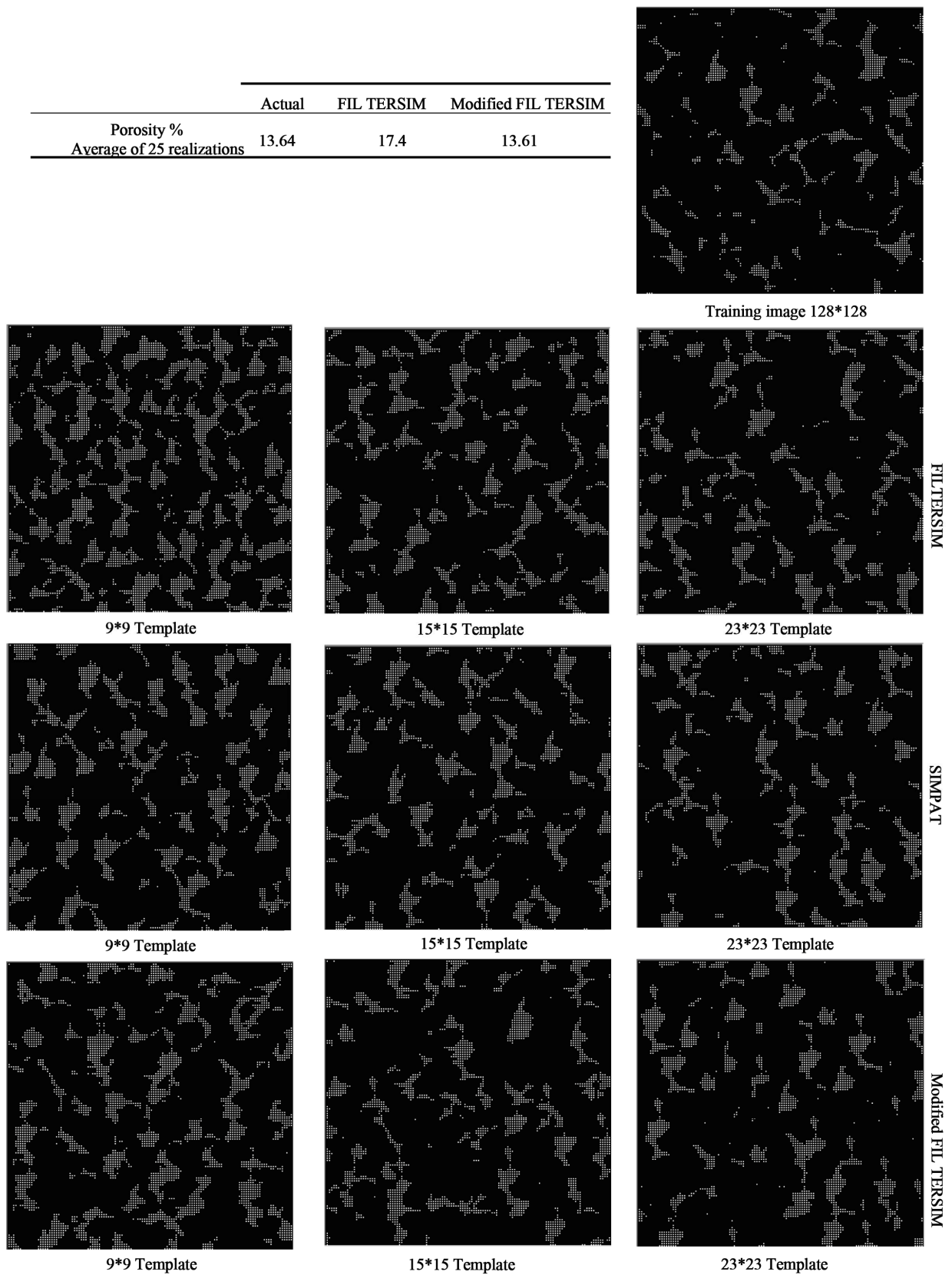


Figure 7. Simulation results with three different template sizes by applying SIMPAT, FILTERSIM and Modified FILTESIM methods (2D thin section image is used as a training image).

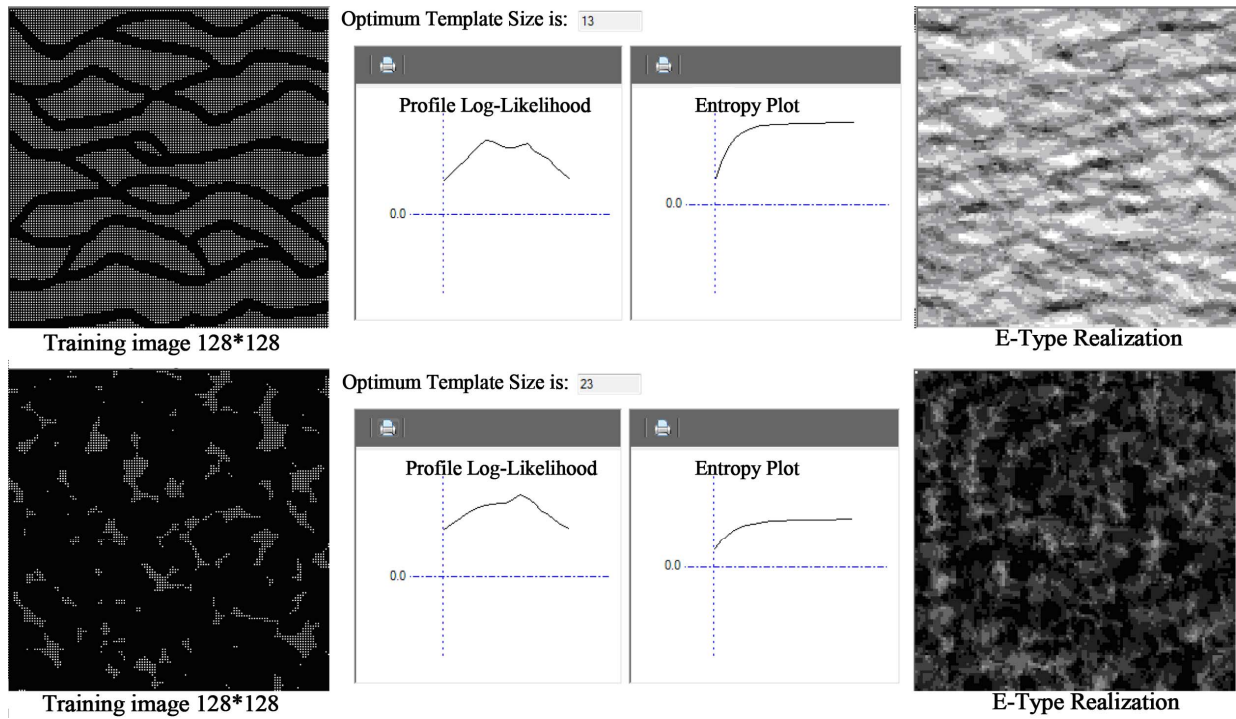


Figure 8. E-Type realization obtained from 14 realizations by optimum sized template.

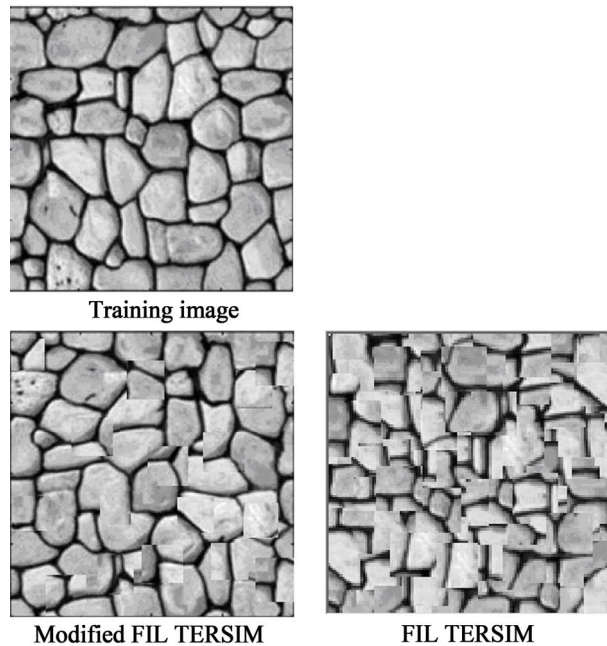


Figure 9. 2D continuous training image and simulation result (Optimum Template Size is 15).

component increase the chance of more repeated patterns to be selected; and prototypes and patterns are chosen based on their minimum manhattan distance function with data events within the realization under construction. These modifications considerable improved the algorithm performance and produced more visual appealing images compare to FILTERSIM. These improvements

add marginal computation cost to FILTERSIM and yet still much faster than SIMPAT.

Our algorithm, SIMPAT and original FILTERSIM have been implemented in C#. We have made our code stand alone and provided a user interface for uploading training and hard data to display simulation results. In the future, we plan to extend Modified FILTERSIM algo-

rithm to reconstruction of non-stationary images and 3D pore space structures. The later will be based on 2D real thin section images.

REFERENCES

- [1] G. Arpat and J. Caers, "A Multiple-Scale, Pattern-Based Approach to Sequential Simulation," *GEOSTAT 2004 Proceedings, 7th International Geostatistics Congress*, Banff, October 2004.
- [2] S. Strebelle, "Conditional Simulation of Complex Geological Structures Using Multiple-Point Statistics," *Mathematical Geology*, Vol. 34, No. 1, 2002, pp. 1-21. [doi:10.1023/A:1014009426274](https://doi.org/10.1023/A:1014009426274)
- [3] A. G. Journel and T. Zhang, "The Necessity of a Multiple-Point Prior Model," *Mathematical Geology*, Vol. 38, No. 5, 2006, pp. 591-610.
- [4] J. Ortiz, "Characterization of High Order Correlation for Enhanced Indicator Simulation," PhD Thesis, University of Alberta, Edmonton, 2003.
- [5] J. Boisvert, M. J. Pyrcz and C. V. Deutsch, "Multiple-Point Statistics for Training Image Selection," *Natural Resources Research*, Vol. 16, No. 4, 2007, pp. 313-321.
- [6] K. G. Boogaart, "Some Theory for Multiple Point Statistics: Fitting, Checking and Optimally Exploiting the Training Image," *International Association for Mathematical Geology XIth International Congress*, 2006.
- [7] F. Guardiano and M. Srivastava, "Multivariate Geostatistics: Beyond Bivariate Moments," In: A. Soares, Ed., *Geostatistics Troia*, Kluwer Academic Publications, Dordrecht, 1993, pp. 133-144. [doi:10.1007/978-94-011-1739-5_12](https://doi.org/10.1007/978-94-011-1739-5_12)
- [8] S. Strebelle, "Sequential Simulation Drawing Structures from Training Images," Ph.D. Thesis, Stanford University, Stanford, 2000.
- [9] J. Straubhaar, P. Renard, G. Mariethoz, R. Froidevaux and O. Besson, "An Improved Parallel Multiple-Point Algorithm Using a List Approach," *Mathematical Geosciences*, Vol. 43, No. 3, 2011, pp. 305-328. [doi:10.1007/s11004-011-9328-7](https://doi.org/10.1007/s11004-011-9328-7)
- [10] J. Caers, "Geostatistics: From Pattern Recognition to Pattern Reproduction," *Developments in Petroleum Science*, Vol. 51, 2003, pp. 97-115. [doi:10.1016/S0376-7361\(03\)80010-9](https://doi.org/10.1016/S0376-7361(03)80010-9)
- [11] X. Y. Liu, C. Y. Zhang, Q. S. Liu and J. Birkholzer, "Multiple-Point Statistical Prediction on Fracture Networks at Yucca Mountain," *Environmental Geology*, Vol. 57, No. 9, 2004, pp. 1361-1370.
- [12] M. J. Ronayne, S. M. Gorelick and J. Caers, "Identifying Discrete Geologic Structures That Produce Anomalous Hydraulic Response: An Inverse Modeling Approach," *Water Resources Research*, Vol. 44, 2008, Article ID: W08426, 16 p. [doi:10.1029/2007WR006635](https://doi.org/10.1029/2007WR006635)
- [13] M. Stien, P. Abrahamsen, R. Hauge and O. Kolbjørnsen, "Modification of the Snesim Algorithm," *EAGE, Petroleum Geostatistics*, 2007.
- [14] M. Huysmans and A. Dassargues, "Application of Multiple-Point Geostatistics on Modelling Groundwater Flow and Transport in a Cross-Bedded Aquifer," *GEOENV VII — Geostatistics for Environmental Application*, Vol. 16, 2010, pp. 139-150.
- [15] H. Okabe and J. Blunt, "Pore Space Reconstruction Using Multiple-Point Statistics," *Journal of Petroleum Science and Engineering*, Vol. 46, No. 1-2, 2005, pp. 121-137. [doi:10.1016/j.petrol.2004.08.002](https://doi.org/10.1016/j.petrol.2004.08.002)
- [16] T. Zhang, P. Switzer and A. Journel, "Filter-Based Classification of Training Image Patterns for Spatial Simulation," *Mathematical Geology*, Vol. 38, 1, 2006, pp. 63-80. [doi:10.1007/s11004-005-9004-x](https://doi.org/10.1007/s11004-005-9004-x)
- [17] M. Honarkhah and J. Caers, "Stochastic Simulation of Patterns Using Distance-Based Pattern Modeling," *Mathematical Geosciences*, Vol. 42, No. 5, 2010, pp. 487-517. [doi:10.1007/s11004-010-9276-7](https://doi.org/10.1007/s11004-010-9276-7)
- [18] M. Honarkhah and J. Caers, "Classifying Existing and Generating New Training Image Patterns in Kernel Space," *21st SCRF ALIATE Meeting*, Stanford University, 2008.
- [19] S. Chatterjee, R. Dimitrakopoulos and H. Mustapha, "Dimensional Reduction of Pattern-Based Simulation Using Wavelet Analysis," *Mathematical Geosciences Work to be Submitted*, 2011.
- [20] Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, Vol. 27, 1948, pp. 379-423, 623-656.