Simulation of Categorical Variables with Multiple Point Statistics and Random Perturbations: MPS-RP

Clayton V. Deutsch

A new MPS simulation algorithm is proposed. The algorithm is iterative with an objective function that controls reproduction of two-point and multiple-point statistics. Convergence is rapid and complex features over large length scales are reproduced. Computer time is reasonable and this algorithm/program is a contender with other algorithms. The most significant advantages are simplicity of parameter specification and simplicity of code. Additional constraints are easily integrated into the code. The disadvantages are that computer time may be significant if very high order statistics are desired or if the model size is very large. Locations are visited in a directed path – spiraling away from the data. The category at each location is randomly perturbed and the change is kept if the model is improved. The program is documented and some small examples are given.

Introduction

Multiple point statistics (MPS) were proposed in geostatistics in the early 90s (Srivastava, 1992; Guardiano and Srivastava, 1992; Deutsch, 1992). Some notable advances have been made since then on the practical implementation (Strebelle, 1998; Ortiz, 2002; Lyster, 2007). MPS is typically applied in the simulation of a limited number of categorical variables (2 to 5). A training image must be available that contains the spatial features of the categorical variable to be simulated. The number of categories and the order of the statistics used in MPS-based simulation cannot be too large or the multiple point patterns will not be reliably found in the training image. Finding training images that look interesting is easy; finding training images that truly represent the subsurface being modeled is not so simple. MPS-based simulation is very applicable in situations with sparse data and a lack of a suitable process-mimicking approach to simulate the geological structures (Pyrcz, 2004). The features that will appear in the final models are fully and explicitly disclosed in a training image and not hidden in some statistics that are hard to understand and inaccessible in any case.

There are two main algorithms currently available for MPS-based simulation: (1) scanning and storing multipoint probabilities from a training image and using them directly in a sequential simulation paradigm, and (2) starting from an initial image and iterating toward a solution. The direct use of multipoint distributions in sequential simulation was popularized by SNESIM (Strebelle, 1998 and many other more recent references). The Gibbs sampler approach of Lyster (Lyster, 2007) is a recent implementation of an iterative approach. There are a number of public domain and commercial programs that implement variants of these two MPS approaches. There are other less common MPS simulation approaches based on other variations including patterns and filtering patterns. A thorough review will have to be found elsewhere.

The algorithm presented in this paper is iterative. The annealing approach of Deutsch (1992) was revisited for the Beauty Contest (presented in paper 107 of this report). The original code compiled and executed fine, but a number of observations were made that led to a new algorithm/program: (1) the simulated annealing decision rule is not required for convergence because local minima are not to be avoided – they are plausible realizations; searching for a global minima is not the goal, (2) a compact arrangement of multiple locations for a multiple point distribution contains a tremendous amount of information on the spatial features, (3) two-point transition probabilities over large distance contain large scale information that may be important, (4) it is straightforward to weight disparate components of an objective function to achieve simultaneous convergence to different objectives, (5) the computer time for a steepest descent-like random approach is quite reasonable, and (6) the simplicity of implementation and parameter specification are attractive.

There may be no need for yet another MPS-based simulation algorithm/program, but the simple and transparent program with open source code (that is quite accessible) will provide competition to the current algorithms/programs that are not always simple, open or accessible for experimentation.

Framework of MPS-RP

Consider a 3-D Cartesian grid defined by Nx, Ny and Nz grid node locations. Each grid cell is to be assigned an integer code from the set k=1,…,K representing a discrete or categorical variable such as facies, lithology or rock type. The number of grid nodes (Nx x Ny x Nz) would likely be between 10^5 and 10^8. The number of categories would likely be between 2 to 7. A training image (TI) with the same set of integer codes and the desired features must be available. The size of the training image need not be the same as the grid being simulated. The proportions of the categories in the training image should be close to that being simulated. An objective function will constrain the simulated realizations to reproduce local proportions coming from vertical proportion curves or trend maps. The objective is to simulate realizations of the categorical variable that reproduce important spatial features. The spatial features of importance include multiple point statistics, two-point statistics, local proportions and conditioning data. Other features could be added to the algorithm as needed.

A classic iterative approach is adopted. A random category from the local proportions is assigned at each location. The conditioning data are fixed and never permitted to change. The grid nodes are all visited in a random order spiraling away from the conditioning data to ensure that they are reproduced without a discontinuity. Normally, 5 to 15 loops through all grid nodes will ensure convergence of the final realization. When a grid node is visited a number of steps are taken: (1) an alternative category is considered, (2) all objective functions are updated and combined into a change in the global objective function, (3) a change may be made to the current category if the objective function improves. These random perturbations are remarkably effective at permitting convergence of an initially random realization toward one that satisfies all of the constraints.

Multiple realizations are generated by starting from a different initial image, following a different random path and selecting a different random category at each location. The framework of the program is remarkably simple. The only challenge is to formulate the components of the objective function that ensure desired features are reproduced in the simulated realizations. The components in the objective function are discussed next, then the weighting of the different components must be considered.

Objective Function Component: Local Category Proportions

The heart of any iterative algorithm is the objective function. The objective function consists of a number of components that are weighted to have equal importance in controlling the convergence of the final realization.

Reproduction of the proportions of each category is important. These first order biases are important for resource assessment and may also influence connectivity for reserve calculation. The global proportions of the training image are contained in the two-point and multiple point statistics. This is advantageous when the simulated realizations are to have the same proportions as the training image, but it is common to aim for slightly different proportions with, perhaps, vertically and areally varying proportions. The implementation documented here requires either the user to (1) simply consider the training image as having the correct proportions, or (2) specify a 3-D proportion cube that could come from a vertical proportion curve, areal proportion maps, a combination of the two, or full 3-D trend models of the proportions. If no 3-D proportion cubes are specified, then no special objective function will be considered for the category proportions. The specification of 3-D proportion cube introduces a number of constraints in the program.

The first constraint is that a category will be forbidden for any location where the local proportion for that category is set to zero (practically less than 0.01). Moreover, a category will be frozen if the proportion for that category is set to one (practically greater than 0.99). This is useful because, in practice, there are almost always large scale trends and other non-stationary features in category proportions.

Secondly, the model is initialized by random sampling from the locally varying proportions. Thus, the initial model would already reproduce important large scale features contained in the local proportion models. This may be enough to ensure that the final model reproduces the proportions, but a component could be added to the objective function to ensure that the local proportions do not “drift” in the final model. Consider n_p equally spaced probability/proportion intervals, e.g., if n_p =10, then consider 0-0.1,

0.1-0.2, … ,0.9-1.0. All locations in the model are classified into one of the n_p classes for each category. Then, the objective is to ensure that the correct number of cells are reproduced in the final realization:

$$O_p = \sum_{k=1}^{K} \sum_{i=1}^{n_p} \left(n_{k,i}^{\text{Training Image}} - n_{k,i}^{\text{Realization}} \right)^2 \quad (1)$$

This objective function would have a different behavior than the other ones, because it should start near zero. Thus, the simulation should just ensure that it does not become too high. Weighting is discussed below. As mentioned this objective function has not been added. Locally varying proportions are, for now, accounted for by the probability of sampling at each location.

Objective Function Component: Two Point Transition Probabilities

Two-point transition probabilities are extracted for many different distance vectors and included in the objective function. A two-point transition probability is completely redundant if a full multiple point statistic contains the same lag vector, but the multiple point statistics considered (see below) are simplified and the directional information of the two-point transition probabilities contain important features. The advantage of explicitly considering two-point transition probabilities is that they can be considered for large distances and their frequency is reliably sampled from any training image. A compact arrangement of lag vectors is considered to ensure there are no artifacts and to simplify parameter selection

The user would simply set an overall anisotropy and the number of lags desired. The program will determine the specific lags. The anisotropy is specified in the units of grid cells. This means that the anisotropy would be more isotropic than the actual subsurface because cells are normally already anisotropic. An area of future work is to automatically determine the anisotropy.

The objective function is to reproduce all K^2 transition probabilities for each lag. The transition probabilities are asymmetric, that is, the transition from k to k' for vector $\mathbf{h}_l$ is not the same as the transition from k' to k for $\mathbf{h}_l$. Of course the transition probability from k' to k for vector $-\mathbf{h}_l$ is the same as the transition probability from k to k' for vector hl. The lag selection is such that both $\mathbf{h}_l$ and $-\mathbf{h}_l$ are not chosen. The objective function:

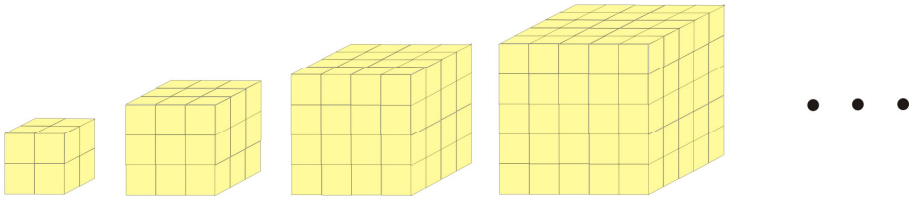
$$O_{TP} = \sum_{l=1}^{n_L} \sum_{k=1}^{K} \sum_{k'=1}^{K} \left(p(k,k',l)^{\text{Training Image}} - p(k,k',l)^{\text{Realization}} \right)^2 \quad (2)$$

As an implementation detail, the transition probabilities that involve original conditioning data are assigned greater importance than those that involve two simulated locations. The conditioning data are frozen from the beginning and this weighting ensures that the conditioning data are not reproduced with a discontinuity. This objective function will start very high if an initially random image is chosen. Convergence toward the transition probabilities of the training image will be rapid and must be balanced against convergence toward the locally varying proportions and the multiple point statistics.

Objective Function Component: Multiple Point Statistics

To maintain simplicity and transparency, a cubic multiple point configuration is considered by MPS-RP. One could question this choice in presence of strong anisotropy, but most anisotropy is accounted for in the selection of the cell size. Consider the common choice of a grid spacing of 50m horizontally and 0.5m vertically in a petroleum reservoir – this is a 100:1 anisotropy consistent with many clastic environments. The *cubic* configuration would really have a 100:1 anisotropy.

Arbitrary non-cubic configurations that could be tailored to the TI were considered during the development and experimentation phase of program development; however, strong anisotropy is captured in the anisotropy of the grid cells and a compact square or cubic arrangement was found to work well. The size of the cube would normally be set between 2 to 8.



The full multiple point histogram for each configuration could be quite large. The dimension would be K^M, where M is the number of locations in the multiple point configuration. A large case would have K=8 and M=8x8x8 that would lead to a dimension of 8^{512} which is inconceivably large. Instead of retaining the full multiple point histogram, the precise location of the rock types within the configuration is not considered, but simply the frequency of observing 1,…,M of each RT within the configuration. This reduces the dimension to KxM, which is much more manageable (8x512=4096).

These simple multiple point statistics contain a great deal of information about the organization and structure of the categorical variable. A high frequency of all points M to be in certain categories would indicate that there is a high degree of structure in that category. Categories that are mixed would not have high order patterns within a single rock type. Figure 1 illustrates the multiple point distributions for a training image (and a random image not shown) for a 5x5 pattern. The black dots on the figure to the right are for the training image and the red dots are for a random image with the same proportion of white/black rock types. Notice that there is a significant number of times in the training image that all 25 locations are simultaneously black. There is a lesser frequency of times that none of the 25 are black, but still reasonable. The probability of all black or white in a random image is very small – notice the Gaussian like curve for the random image.

There is no need to consider all KxM values for all categories in an objective function because there is redundancy in the information, particularly with only two categories; however, for simplicity of coding, the objective function considers the multiple point distribution for all KxM, that is:

$$O_m = \sum_{k=1}^{K} \sum_{m=1}^{M} \left(p_{k,m}^{\text{Training Image}} - p_{k,m}^{\text{Realization}} \right)^2 \tag{3}$$

This will ensure that all of the local organization and structure contained in the multiple point distributions are reproduced in the final model.

Weighting Objective Function Components

Each component of the objective function will have different units and a different impact on the acceptance/rejection of changes in the random perturbation. Weighting the components is necessary to ensure that each component will contribute to ultimate convergence. One idea would be to standardize the components by the initial objective function values; however, the acceptance/rejection decision is based on a change to the objective function and not the absolute value of the objective function. Also, some objective functions may start near zero yet should not be given inordinate weight.

The weighting scheme of the author's PhD (Deutsch, 1992) was adapted to the MPS-RP program with good results. Consider the objective function as a sum of multiple components:

$$O = \sum_{i=1}^{n} w_i O_i \tag{4}$$

Some possible objective functions were presented above, but any objective function components could be considered. The weights *wi, i=1,…n* are to be determined such that each component contributes equally to acceptance decisions based on the global objective function thus permitting convergence of all components. Also, the weights are standardized such that the initial objective function is one. The weights are calculated as:

$$s_i = \frac{1}{\sum_{i=1}^{npert} \left| O_i - O_i^p \right|} \qquad w_i = \frac{s_i}{\sum_{i=1}^{n} s_i O_i^0} \tag{4}$$

Where si is the sensitivity of the i^{th} component determined by some number of perturbations (*npert*), O^p is the new unscaled objective function with a random change, the O^{0}'s are the initial objective function values. The initial combined objective function will be exactly 1 and each component will have equal importance during the optimization process. This approach is heuristic, but has worked in all of the cases considered by the program.

Program Parameters

The source code and executable for the program is provided in the directory accompanying this paper. As claimed above, the parameters are relatively simple (18 lines of parameters):

```
1                     MPS-RP Simulation Program
2                     *************************
3
4  START OF PARAMETERS:
5  1                                -number of realizations
6  200 100   1  5                   -nx, ny, nz, number of categories
7  1 2 3 4 5                        -  categories
8  lvprop.dat                       -file with local proportions (optional)
9  1 2 3 4 5                        -  columns for proportions
10 69069                            -random number seed
11 mpsrp.out                        -file for realizations
12 mpsrp.dbg                        -file for debugging
13 3   50000   50                   -debug level, reporting, number of loops
14 1                                -starting image option (1=random, 3=file)
15 initimage.dat                    -  file with starting images
16 nocond.dat                       -file with conditioning data
17 1  2  3  4                       -  columns for IX, IY, IZ, category
18 train.dat                        -file with training image
19 200 100   1                      -  nx, ny, nz
20 40      10                       -two point hist: nlag, cwt
21 0.0 0.0 0.0  50.0 50.0 10.0      -  ang1,ang2,ang3,rad1,rad2,rad3
22 4                                -size of multipoint histogram cube (2-5)
```

The number of realizations, the grid size, the number of categories and the category values are evident. A 3-D file of locally varying proportions could be specified (Lines 8 and 9) at the exact 3-D grid specification as being simulated. The random number seed, file for the realizations and file for debugging are evident. The parameters on Line 13 are the debugging level (between 0 to 5), the reporting number (50000 as a default) is how often the program will write the current objective function values to standard output, and the number of loops (50 as a default, but 25 is more than adequate) is the number of times the program will loop over all grid nodes in the realization. A different path will be followed for each realization.

The starting image can be generated randomly from the local proportions or the global proportions taken from the training image (this is option 1 on Line 14). The initial image could also come from a file (option 2 or 3 on Line 14). Option 2 means that the first realization in the input file will be used for all realizations and option 3 means that the file with starting images (Line 15) contains as many starting images as the number of realizations being simulated.

The conditioning data file and column specifications are given on Lines 16 and 17. Note that the coordinates must be in grid index. This may require the input data to be processed ahead of time to convert from X to IX and so on; this is straightforward. Lines 18 and 19 define the training image (file and size). The size of the training image need not be the same as the size of the domain being simulated.

The two-point transition probabilities that will enter the objective function are specified on Lines 20 and 21. The number of two-point histograms is the first number on Line 20. The closest will be taken according to the anisotropy specification on Line 21. The anisotropy is the standard three angles and three ranges definition of GSLIB. An area of future work is to automate the determination of anisotropy from the training image using a measure of entropy. The second number on line 21 is a factor that will increase the importance of two-point transition probabilities associated to conditioning data. The default of 10 is quite large. A value between 2 to 10 would be reasonable.

The size of the multipoint histogram cube is specified on the last Line 22. A size between 2 and 5 is suggested. In 3-D this would correspond to multiple point statistics of orders 8, 27, 64 or 125. A great deal of information is contained in this size of template.

Examples

A synthetic training image was presented in Deutsch's PhD thesis. It was used as a TI for MPS-RP and three images were created, see Figure 2. The images appear to reasonably reproduce the features of the training image. There are some subtle artifacts at the edges of the model. An area of future work may be to expand the grid, simulate, then clip the grid. This could be done manually.

Another 2-D training image was considered in a second example, see Figure 3. This image comes from a core photo of breccias facies in the McMurray facies. The simulated realizations appear close to the training image. The noise from the image processing is also reproduced. Of course, all features from the training image are enforced in the simulated realizations.

Yet another 2-D training image and three realizations are shown in Figure 4. The training image came from scanning and categorizing a small hand sample cut from aeolian sediments in the Southwestern US. The realizations look reasonable. There are some small-scale curvilinear features in the training image that are not reproduced. If these features are deemed important for the final model application, specific multiple point probabilities would have to be introduced to reproduce them.

Some additional examples of the program are included in the Beauty Contest paper in this report. Some slices through a 8 million cell 3-D training image and realization are shown on Figure 5. Features are reasonably reproduced.

Conclusions

Another MPS simulation program is developed and documented in this paper. The reasons to consider another alternative is simplicity – the number and complexity of parameters is limited, the code is short and robust, and additional objective function components could be added easily. The algorithm is iterative with some of the disadvantages that go along with that; unknown CPU time and no guarantee of convergence; however, practice shows that these are not a concern. Realizations converge within a well understood time in all of the examples considered.

The program is ready for application, but there are some areas for future work that could be considered. First, the explicit control on the local proportions in the objective function would be needed in most practical applications. Seeding the model with the local proportions and drawing from the local proportions in the perturbation mechanism are not enough to ensure reproduction of local proportions. Second, the updating of the objective function components could be made more efficient resulting in a significant improvement in CPU speed. Other areas include the automatic selection of the lag vectors and MPS template by scanning the training image for low entropy features.

References

Deutsch, C.V. (1992) Annealing Techniques Applied to Reservoir Modeling and the Integration of Geological and Engineering (Well Test) Data. Ph.D. Thesis, Stanford University, 304 p.

Deutsch, C.V. and Journel, A.G., 1998, *GSLIB: Geostatistical Software Library and User's Guide*, Oxford University Press, New York, 2nd Ed., 369 pp.

Arpat, G.B. and Caers, J. (2007) Conditional Simulation with Patterns. Mathematical Geology, Vol. 39, No. 2, Feb. 2007, pp 177-203.

Deutsch, C.V. and Journel, A.G. (1998) GSLIB: Geostatistical Software Library and User's Guide, 2nd Ed. Oxford University Press, New York, 369 p.

Guardiano, F.B. and Srivastava, R.M. (1993) Multivariate Geostatistics: Beyond Bivariate Moments. Soares, A., Editor, Geostatistics Troia '92, Vol. 1, pp 133-144.

Lyster, S. and Deutsch, C.V. (2008) MPS Simulation in a Gibbs Sampler Algorithm. 8th International Geostatistics Congress, 10 p.

Lyster, S.J., 2009, Simulation of geological phenomena using multiple-point statistics in a Gibbs sampler algorithm, Ph.D. Thesis, University of Alberta, Canada, 223 p.

Ortiz, J.M., Characteration of high order correlation for enhanced indicator simulation. PhD thesis, University of Alberta, 2003.

Pyrcz, M.J., 2004, The integration of geological information into geostatistical models, Ph.D. Thesis, University of Alberta, 250p.

Srivastava, M. (1992) Iterative Methods for Spatial Simulation. Stanford Center for Reservoir Forecasting, No. 5, 24 p.

Strebelle, S. (2002) Conditional Simulation of Complex Geological Structures Using Multiple-Point Statistics. Mathematical Geology, Vol. 34, No. 1, Jan. 2002, pp 1-21.

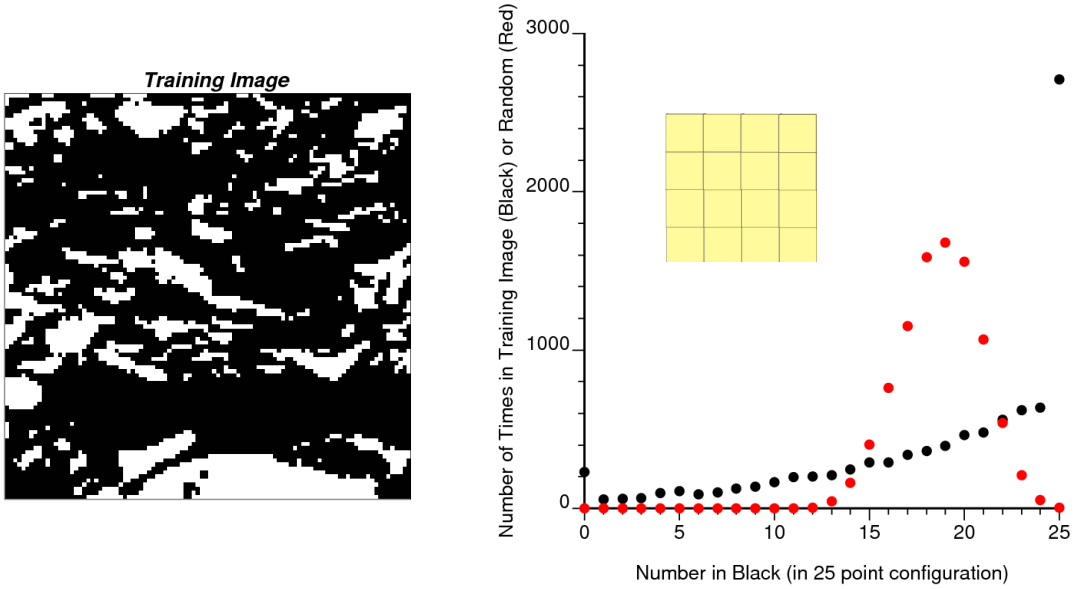


Figure 1: Training image and multiple point distribution within a 5x5 template (black dots) and for a random image with the same proportion of black (red dots).

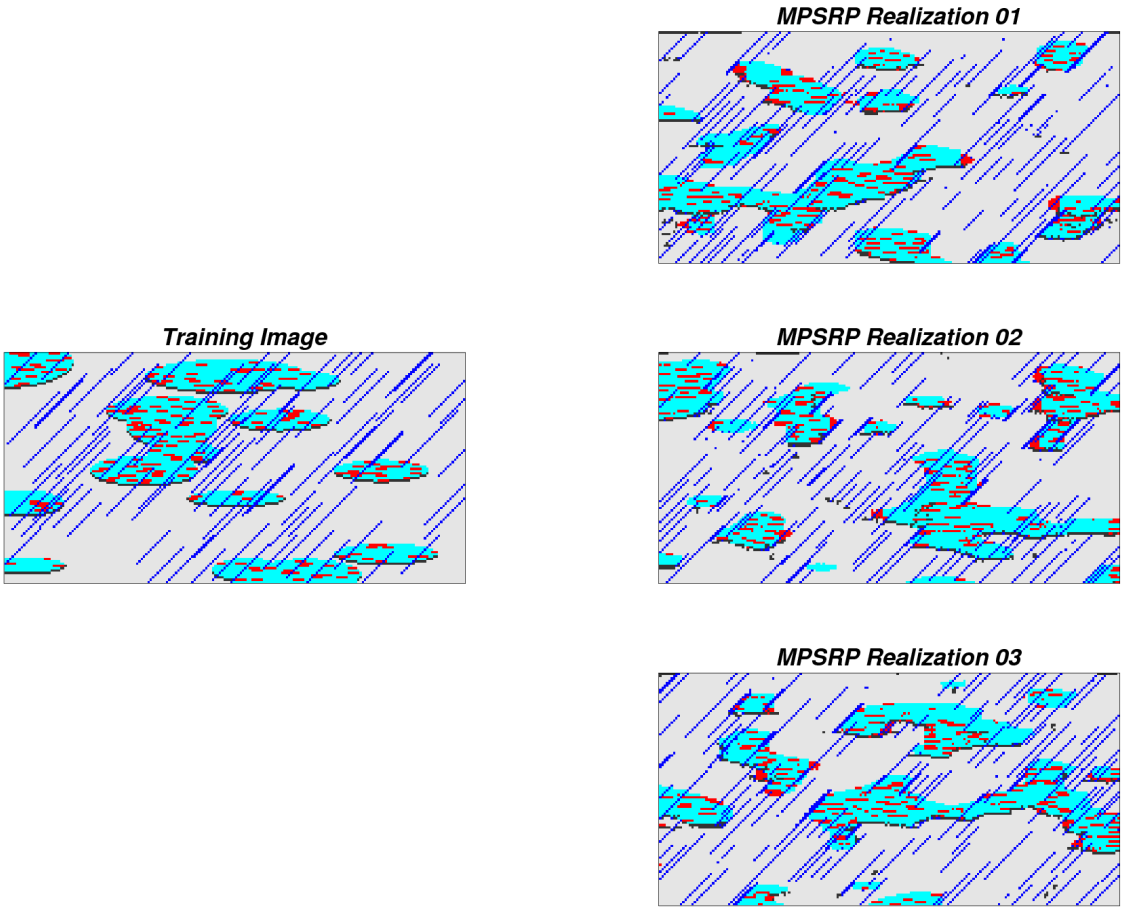


Figure 2: Training image and three realizations.

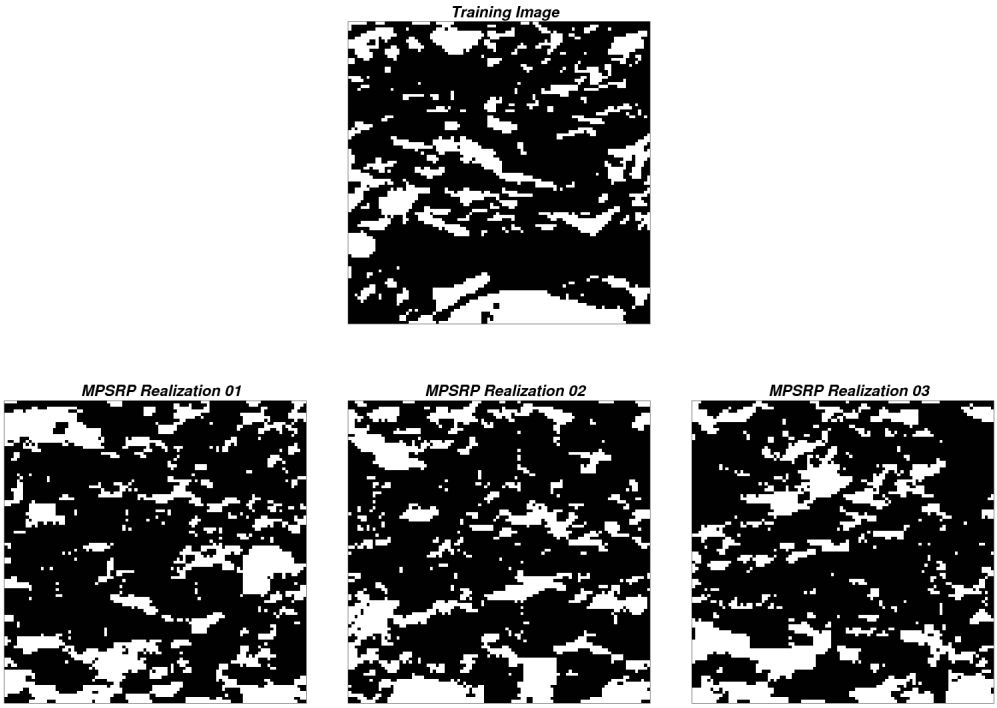


Figure 3: Another training image and three realizations.

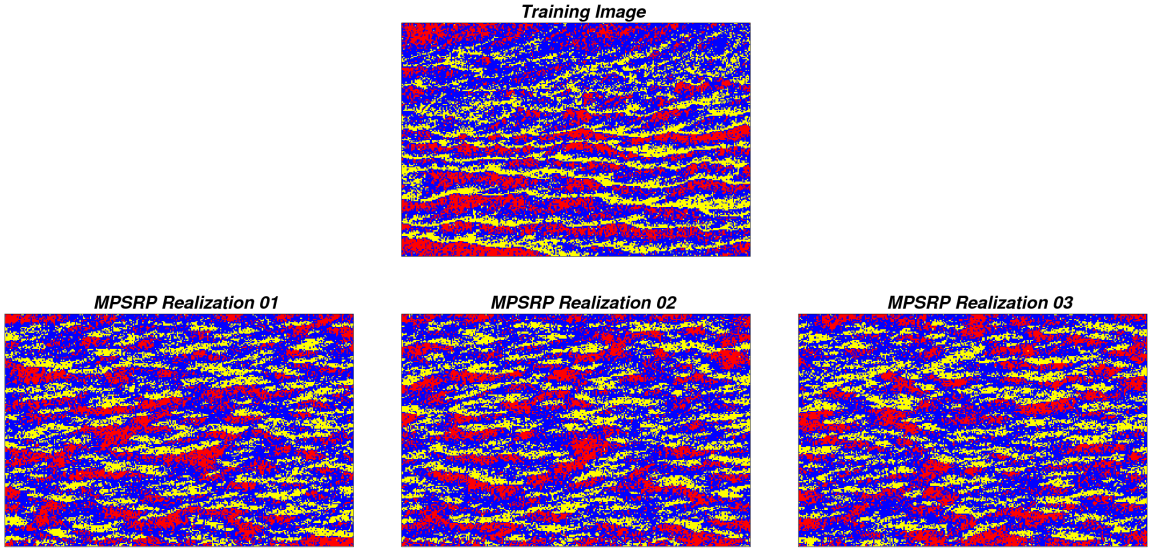


Figure 4: Yet another training image and three realizations.