

TISIS: A Program to Perform Full Indicator Cosimulation Using a Training Image

Steven Lyster and Clayton V. Deutsch

Centre for Computational Geostatistics
Department of Civil & Environmental Engineering
University of Alberta

A training image (TI) provides all required spatial statistics to inform the spatial law of a random function (at the grid spacing of the training image). A TI is normally used with a multiple point statistics (MPS) based algorithm; however, any algorithm can be used. This paper documents a full indicator cokriging approach with spatial statistics inferred directly from a training image. The results of this approach provides useful models, base case results for comparisons, and advanced starting points for iterative MPS algorithms. There are no problems associated with fitting a valid model of coregionalization between all direct and cross variograms; the set extracted from the TI is positive definite. A GSLIB-like program is documented.

Introduction

Characterization of natural resources such as petroleum reservoirs or complex ore deposits often necessitates the creation of a geologic model of rock types before modeling of continuous properties such as porosity, permeability, and ore grade. Such a model is required because in many cases the properties of interest vary widely from rock type to rock type, or facies to facies. The resulting model, integrating geologic uncertainty as well as uncertainty within facies, should give a better idea of the total uncertainty within a model (Deutsch, 2002).

A common method used to build a rock type model is to code the facies as indicators, characterize the spatial structure of each indicator with variograms and/or covariance functions, and then perform sequential simulation. This method relies on the assumption that the facies all have multivariate Gaussian spatial structure.

In the typical implementation, the indicator variograms of all facies are calculated separately; there is an implicit assumption that the different rock types are independent of one another. In most cases this is not true, and there are in fact spatial relations between the facies. The difficulty in producing a linear model of coregionalization including all direct and cross variograms (or covariances) leads to the simplification. It is often not worth the effort to produce a valid LMC when the number of different facies exceeds three.

Training images are becoming more common as multiple-point geostatistics gain in popularity. Most high-order spatial structures are too complex to characterize as functions, so conditional probabilities and other such statistics are calculated from a training image deemed fully representative of the area in question. With such a resource available, it is possible to derive whatever information is desired. Direct calculation of covariances and cross-covariances for all facies is straightforward with no fitting of functions, and the resulting model of spatial structure is guaranteed to be positive semidefinite as there is at least one realization which honours all of the statistics (the training image).

Using a training image to derive all relevant two-point statistics may be desirable for several reasons. Compared to multiple-point methods, sequential indicator simulation is very fast; features such as locally varying proportions and soft secondary data may be easily integrated; conditioning data is explicitly honoured; and with all cross-covariances included in the kriging equations, the relations between the facies should be correct. The realizations resulting from full indicator cosimulation could be used for comparison to MPS methods, initial images for iterative MPS algorithms, or as final facies models.

Indicator Simulation

The expression of facies as mutually exclusive coded indicators may be accomplished through the transform (Deutsch, 2002):

$$I(u; k) = \begin{cases} 1, & \text{if facies } k \text{ is present at location } u \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

This transform must be applied K times at each data location, where K is the number of different facies. By coding the information in this way, an exhaustive and exclusive distribution is created; all locations have a single facies which occurs at that point (and therefore has an indicator value of 1), and all other facies do not occur at that particular location (and therefore have indicator values of 0). The expected value of an indicator is equal to the global proportion or probability of k , and is denoted as P_k . The variance of a categorical indicator is equal to $P_k(1-P_k)$.

Indicator values may be estimated at unsampled locations, as with any other variable. The linear estimate may be expressed as (Chiles and Delfiner, 1999):

$$I^*(u; k) = \sum_{i=1}^n \lambda_i \cdot [I(u_i; k) - P_k] + P_k \quad (2)$$

where the u_i 's are sampled or previously simulated locations, and the λ_i 's are the linear estimation weights. The optimal weights may be found by simple indicator kriging:

$$\sum_{j=1}^n \lambda_j \cdot \text{Cov}\{I(u_i; k), I(u_j; k)\} = \text{Cov}\{I(u_i; k), I(u; k)\} \quad i = 1, \dots, n \quad (3)$$

Indicator kriging must be performed for all $k = 1, \dots, K$ facies to determine the full conditional probability density function (cpdf). While indicator kriging minimizes the estimation variance, the actual variances themselves are not used in indicator simulation.

After the full conditional distribution has been created, order relations problems must be corrected. The standard method for this is to set any negative probabilities to zero, and then divide each conditional probability by the sum of the cpdf values; this ensures the distribution sums to one.

Once a cpdf has been built, the values are used to create a conditional cumulative density function (ccdf); a random number between zero and one is drawn; and the corresponding indicator value in the ccdf is set as the rock type at location u in the model. The next location is then simulated in the same way; this continues until the entire model has been populated.

Cosimulation

Indicator values other than the one being estimated are not usually used for sequential indicator simulation. Because the categorical indicators are mutually exclusive, the collocated covariance is always zero between facies; this precludes the use of collocated cosimulation, which is a popular option for integrating multiple data types. Also, as there are often four or more facies types, a large number of cross-variograms would have to be calculated in addition to the direct; this discourages most practitioners and is often not worth the effort. With the use of training images, the inference of cross-covariances for complex relations is not a problem; therefore, full indicator cosimulation may be undertaken.

Considering all k facies at all n data locations within the search radius, the linear estimate now becomes:

$$I^*(u; \hat{k}) = \sum_{k=1}^K \sum_{i=1}^n \lambda_i^k \cdot [I(u_i; k) - P_k] + P_{\hat{k}} \quad (4)$$

where \hat{k} is the facies for which the probability is being estimated. The optimal estimation weights may be found by solving the simple indicator cokriging system:

$$\sum_{k=1}^K \sum_{j=1}^n \lambda_j^{k'} \cdot Cov\{I(u_i; k), I(u_j; k')\} = Cov\{I(u_i; k), I(u; \hat{k})\} \quad i = 1, \dots, n \quad k = 1, \dots, K \quad (5)$$

It is notable that in Equation 5, the left-hand side matrix of the cokriging system is identical for all indicator values; only the right-hand side covariance vector must be changed to solve for the linear weights of other indicators. Once the full cpdf has been determined, the sequential simulation proceeds as described above.

TISIS Program

The program that has been written to perform full indicator cosimulation with covariances and cross-covariances derived from a training image is called TISIS (Training Image Sequential Indicator Simulation). The basic framework is similar to the GSLIB programs (Deutsch and Journel, 1998). The user is required to enter the parameters for simulation into a parameter file; all input and output files follow the standard GSLIB file structure. The procedure used by TISIS is described below.

Read Parameters

Upon executing the program, the user is prompted for the name of a parameter file; this name defaults to `tisis.par` if nothing is entered. If the default filename is used and no file is found, a generic template parameter file is created. The parameter file is laid out as follows:

```

Parameters for TISIS
*****

Line  START OF PARAMETERS:
  1   tifile.dat                -Training Image file
  2   1                        -Column for TI data
  3   256 256 128              -Size of TI field
  4   conddata.dat            -Conditioning data file

```

```

5  1  2  0  3          -Columns for X,Y,Z,Data
6  1                  -Number of multiple grids
7  2  2  2          -Discretization of grid
8  5                  -Number of codes/facies
9  0  1  2  4  5     -Indicators for codes/facies
10 0                  -Number of codes to freeze
11 initial.out       -If >0, file with initial image
12 1                  -Column for data
13 0 1               -Codes to freeze
14 tisis.out         -Output file
15 10                -Number of realizations
16 256 5.0 10.0     -nx xmin xsiz
17 256 5.0 10.0     -ny ymin ysiz
18 128 0.5 1.0      -nz zmin zsiz
19 2                  -0=Global proportions,1=local,2= from TI
20 0.1 0.1 0.2 0.4 0.2 -If 0, global pdf
21 localpdf.dat      -If 1, local pdf file
22 1 2 3 4 5         -Columns for local pdf values
23 12                -Maximum data to use
24 515               -Number of points in template
25 template.dat      -File containing point offsets
26 1 2 3             -Columns for X,Y,Z offsets
27 69069             -Random number seed value

```

The first three lines of parameters define the training image. The TI may be three-dimensional and the size is limited only by the memory requirements of the data. The next two lines specify the file containing the conditioning data, as well as the columns for the coordinates and facies code; if the conditioning data file does not exist, the realizations will automatically be unconditional. Line 6 defines the number of multiple grids to use; the next *nmg* lines define the discretization of the grids (shown as Line 7 here). The grids must be in order from coarsest to finest. If the value in Line 6 is zero, then there should not be any lines specifying the grids.

Line 8 specified the number of rock codes or facies types contained in the simulation field, while Line 9 contains the codes used to identify the different facies. All data within the training image, conditioning data, and multigridding data files must be identified on Line 9, or an error will occur.

Line 10 specifies the number of rock codes to freeze. If a hierarchical simulation is being performed, Line 10 will have a number greater than zero and Line 11 will specify the input file containing the initial realization; this file will be read in, then the specified codes frozen in place and the remaining locations repopulated. Line 12 contains the column number for the data in the initial image file. Line 13 contains the indicators for the codes to be frozen.

The desired name of the file for output is entered on Line 14. If the file already exists, it will be overwritten. The number of realizations is given on Line 15; any number may be entered, up to the practical limits based on CPU time and storage. The simulation grid is specified on Lines 16 through 18.

The pdf option is on Line 19. If this value is 0, then Line 20 specifies the global P_k values for each rock code; if the option is set to 1, a file containing local pdf values for every facies must be given in Line 21, with the columns stated on Line 22; if the pdf option is set as 2, a global pdf is used and calculated directly from the training image.

Line 23 specifies the maximum number of data to retain for the indicator cokriging. There is no explicit maximum set, but fewer retained data leads to faster simulation speeds and often does not adversely affect the quality of realizations. The number of points in the statistical template is on Line 24. The file with the point offsets is designated on Line 25, and the columns for X, Y, and Z coordinate offsets are on Line 26. The template file must contain at least as many lines as the number of points specified, though fewer may be used than are in the file if so desired.

The initial random seed value for the simulation is given on Line 27. Once the parameter file is read in, the preparation for simulation is executed.

Acquire Training Image

The first step once the parameters have been read in is to read in the training image information. The size of the TI is specified by the user, and is not explicitly limited. The training image must be at the same resolution as the field to be simulated for the results to be valid; this must be checked by the user. If there is an indicator code within the TI file that is not one of those specified in the parameter file, the program returns an error message and stops. An error is also returned if the TI file is not found.

Read Statistical Template

A user-specified template is utilized to calculate and store all necessary statistics. The X, Y, and Z offsets of the template are read from a file. If the column number given for the Z offset values is less than one, the template is assumed by the program to be two-dimensional and all Z offsets are set to zero.

Find Unique Lag Vectors (Offsets)

To speed up calculation of statistics and reduce storage requirements, only unique lag vectors are used. To ensure all vectors are unique, first all vectors are pointed in a “positive” direction: the X offset between the two points must be greater than or equal to zero; if it is equal to zero, the Y offset must be greater than or equal to zero; if both the X and Y coordinates of the two points are the same then the Z offset must be greater than or equal to zero. If the lag vector is oriented in a negative direction then it is reversed and pointed in a positive direction instead.

Once the positiveness of the lag vector is ensured, each combination of points in turn is compared to all previous unique vectors. The comparison is done by seeing if the X, Y, and Z offsets of the vector are identical to any of those already stored.

While the process of finding all unique lag offsets (or vectors) can be somewhat cumbersome, it is very fast and greatly reduces the number of covariances and cross-covariances that must be calculated and stored.

Build Local PDF

A local pdf model must be built, regardless of the pdf option specified in the parameters. If the pdf option is not 1, then the local pdf values are identical to the global ones; otherwise, the local proportions are read in from the specified file. If the file is not found (and the option is 1), then an error is returned and the program stops. The columns for the local pdf values for each code may be specified as desired. Technically, the pdf values do not even need to sum to 1.0; the program should function if the proportions sum to some other value, but this could lead to erratic behaviour in the estimation.

Calculate Statistics

Next, the TISIS program scans the training image field and calculates all of the required covariances and cross-covariances. Variograms are not used because cross-variograms are symmetric whereas cross-covariances do not necessarily have this property (Journel, 1978, p.41). Covariances and cross-covariances are only calculated for the unique offsets/lag vectors found earlier. The values are found by using the following equation:

$$\begin{aligned} \text{Cov}\{I(u_i; k), I(u_j; k')\} &= E\{[I(u_i; k) - P_k] \cdot [I(u_j; k') - P_{k'}]\} \\ &= E\{I(u_i; k) \cdot I(u_j; k')\} - P_k \cdot P_{k'} \end{aligned} \quad (6)$$

The P_k and $P_{k'}$ values are taken from the training image for use in Equation 6, regardless of the specified pdf option. In the cases where $k = k'$ the equation becomes that for direct covariance.

Get Conditioning Data

The final step before the loop over all realizations begins is to read in the conditioning data from the specified file. If the file is not found, all realizations are unconditional. The conditioning data locations are not visited in the spiral path, and remain unchanged from one realization to the next. The columns for the X, Y, and Z coordinates as well as for the rock code may be specified. If the column for the Z coordinate is less than 1, then the data are assumed to be two-dimensional and the block index for all conditioning data is set to one.

Develop Spiral Path

The path followed for simulation begins at the closest points to the conditioning data, and then moves further away. Ties in the minimum distance to conditioning data are broken by small random numbers. Conditioning data are not included in the spiral path, ensuring exact reproduction.

To implement the multiple grid option those points which fall on the finer grids have very large values added to their distance. This guarantees that the simulation will visit the locations on the coarsest grids first, spiraling away from the conditioning data on the larger grids before filling in the remaining locations, once again spiraling away from conditioning data.

Simulate Values

Visiting the nodes in the order developed by the spiral path subroutine, sequential indicator cosimulation is performed. A subroutine first finds the existing data (conditioning or previously simulated) that is within the statistical template defined earlier. The number of data found is limited to the parameter specified by the user.

Once the data locations are known relative to the point currently being simulated, the relevant statistics are extracted from the statistics found earlier and stored. If a lag vector to a point is in a negative direction, the offsets are reversed to find the covariances and cross-covariances. The left-hand side kriging matrix is built, and is the same for estimating the probabilities of all facies (as mentioned above for Equation 5); right-hand side matrices are built for each rock code. Simultaneously solving the kriging systems yields the cpdf. These values are then modified to account for local probabilities by using Bayesian Updating (Deutsch, 2002, p.201):

$$I^*(u; k) = I^*(u; k) \cdot \frac{P_k^{LOCAL}}{P_k^{GLOBAL}} \quad (7)$$

If no local pdf model is being used, then the P_k values are the same locally as globally. Once the updated probabilities are found for each indicator, the cpdf is corrected for order relations and the ccdf is created. A simulated value is then drawn, and the process repeats at the next point in the spiral path.

Write to File

After the grid is fully populated, the results are output to the specified file in standard GSLIB format. The X, Y, and Z coordinates are also output. The procedure then repeats for the remaining realizations, from developing the random spiral path onwards.

Examples

Following are several examples produced by the TISIS program with three different training images. Realizations using conditioning data, locally varying proportions, and multiple grids are also shown.

Channel TI, Two Facies

Figure 1 shows a training image that is 256x256x1 blocks in size, has two indicator values, and contains fluvial channel structures. Below the TI are shown realizations produced using TISIS, with varying numbers of retained data in the kriging equations (2, 4, 8, and 12). It is interesting to note that the realizations do not appear to significantly improve as more previous nodes are used. In fact, as more data are used, the randomness of the simulation field increases.

Channel TI, Three Facies

The same TI as in Figure 1 is again shown in Figure 2; however, in this case there are two distinct channel facies. TISIS realizations which used varying numbers of previous nodes are shown below. In this example it is quite clear that retaining more data makes the simulation noticeably worse, with much more randomness and noise.

Complex TI, Four Facies

Shown in Figure 3 is a training image with four different indicators, which have very complex relations; below the TI are TISIS realizations which once again retain 2, 4, 8, and 12 previous data at each point. As before the simulated field gets progressively worse as more data are retained; with 12 data the main structures of the TI are just barely visible. Looking at the cases with fewer retained data, the directions of continuity and relations between facies are reproduced reasonably well for such a simple algorithm.

Channel TI, Three Facies, Conditioning Data

The training image in Figure 4 is the same as in Figure 2. The TISIS realizations in Figure 4 were created using conditioning data. This conditioning was in the form of four vertical strings of data simulating drillholes or well logs, and is shown next to the TI. The realizations appear to be fairly good near the conditioning data, but very near the midpoint between strings of data there are discontinuities and/or randomness. This is caused by the spiral path, which starts near the conditioning data; close to the data, the simulation reproduces the features well. However, after so many nodes have been informed the paths from each of the strings of conditioning data begin to interact; at this point halfway between the data the artifacts occur.

The ideal number of retained data appears to be different when using conditioning data. Recall the earlier example shown in Figure 2, in which the realizations using only 2 and 4 retained data were visually much better than those retaining 8 or 12. With conditioning data, the spiral path is much more continuous; as such, the spatial structure from the training image is reproduced near the data, and more data need to be retained to ensure the conditional pdf is fully informed. In Figure 4 the “best” number of previous nodes to retain for simulation appears to be in the range of 8-12; the discontinuities are least pronounced at this level.

Channel TI, Three Facies, Locally Varying Proportions

Shown in Figure 5 are plots of local proportions for each of the three facies in the same channel-based TI as above. This local pdf model was created by performing a local search of the TI to find the proportions of the different indicator codes. In general, the background facies occurs with the highest proportions nearly everywhere; the black channel facies has its highest probabilities of occurring in the middle area of the field; and the blue facies type has two distinct zones of higher proportions, one near the top of the field and one about a third of the way up from the bottom.

Realizations produced by TISIS using these locally varying proportions are shown in Figure 6. The local pdf obviously has a significant impact on the results: the simulated “channels” present themselves primarily in the areas where they have the highest pdf values. While the actual local mean model used may not be ideal for a real case, as an example of how the program reproduces the local model the results are quite good.

The number of data retained in the simulation appears to have less impact using a local pdf model. While more points once again results in more noise (as was evident in the earlier examples), overall there is less difference in the realizations based on the number of points retained.

Channel TI, Two Facies, Multiple Grids

Figure 7 shows the same training image with two facies as the first example in Figure 1. Below the TI are four TISIS realizations, each made with a different number of multiple grids as specified in Line 6 of the parameter file. All of the realizations used six retained data for building the ccdfs. While the results are not visually striking, there is more east-west continuity in the realizations which used multiple grids. As the time for simulation does not increase noticeably when this option is being used, even such a subtle improvement is worth effort.

Discussion

In the examples several interesting features may be seen. The use of so few previously simulated nodes for the best reproduction of features in the first few examples is very surprising. This could be caused by the large size of the kriging matrices; retaining eight data with three facies results in a 24x24 left-hand side matrix. The size of this matrix, and the number of terms in the linear estimate, then in turn results in more reliance on the P_k values and hence more randomness. This could also explain why realizations retaining more data begin to go towards randomness more quickly as more facies are added.

The ease with which conditioning data, local proportions, and multiple grids may be implemented is a product of the two-point simulation method. Kriging and sequential simulation have been well refined, and as such this methodology is accepted and widely used. The goal of the implementation in TISIS, which uses all covariances and cross-covariances, is to take two-point

indicator simulation as far as it may possibly go: no further steps may be taken to use more information without moving to a multiple-point statistical method.

The workflow implemented here to perform full indicator cosimulation is meant to be used as a complement to multiple-point simulation. All relations between the facies should be properly reproduced, producing ideal images for MPS post-processing, comparison to MPS simulations, or standalone realizations using an alternative to strict MPS.

References

- J.P. Chiles and P. Delfiner. *Geostatistics: Modeling Spatial Uncertainty*. John Wiley & Sons, New York, 1999.
- Deutsch, C.V. and Journel, A.G., 1998. *GSLIB: Geostatistical Software Library: and User's Guide*. Oxford University Press, New York, 2nd Ed.
- C.V. Deutsch. *Geostatistical Reservoir Modeling*. Oxford University Press, New York, 2002.
- A.G. Journel, Ch.J. Huijbregts. *Mining Geostatistics*. The Blackburn Press, Caldwell, New Jersey, 2003; Original printing 1978.

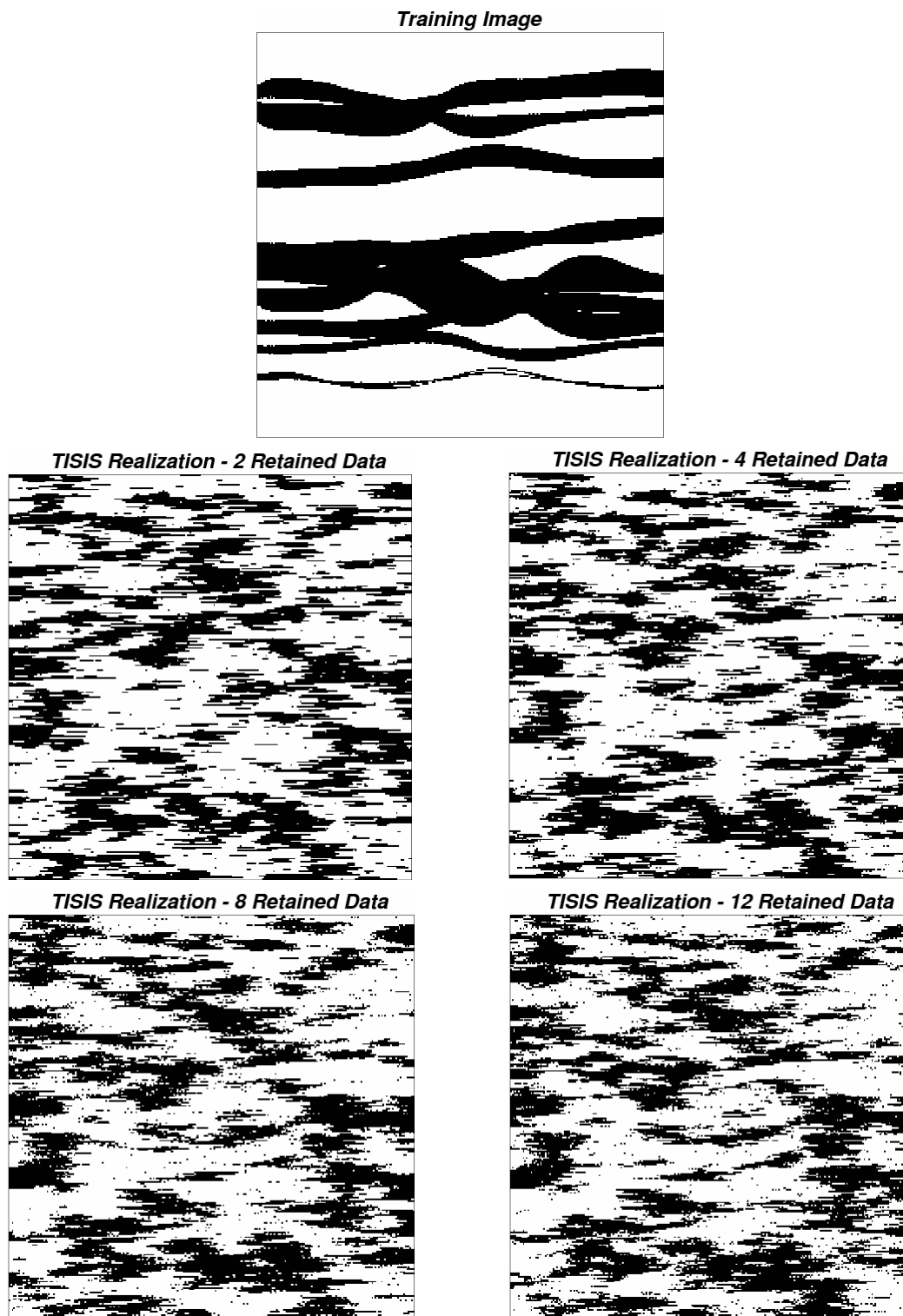


Figure 1: 256x256x1 training image with two facies (top); TISIS realizations using different numbers of retained data (lower).

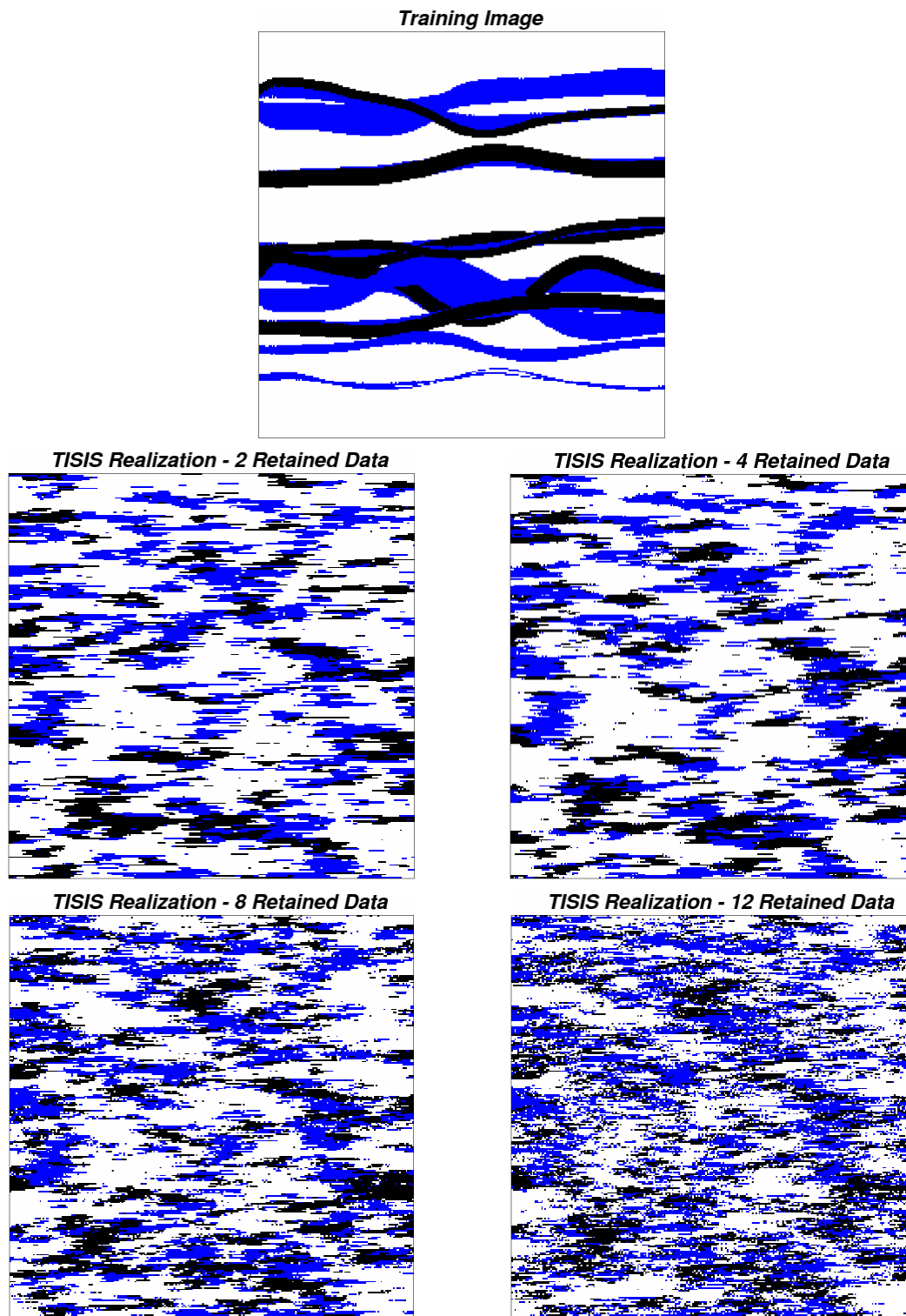


Figure 2: 256x256x1 training image with three facies (top); TISIS realizations using different numbers of retained data (lower).

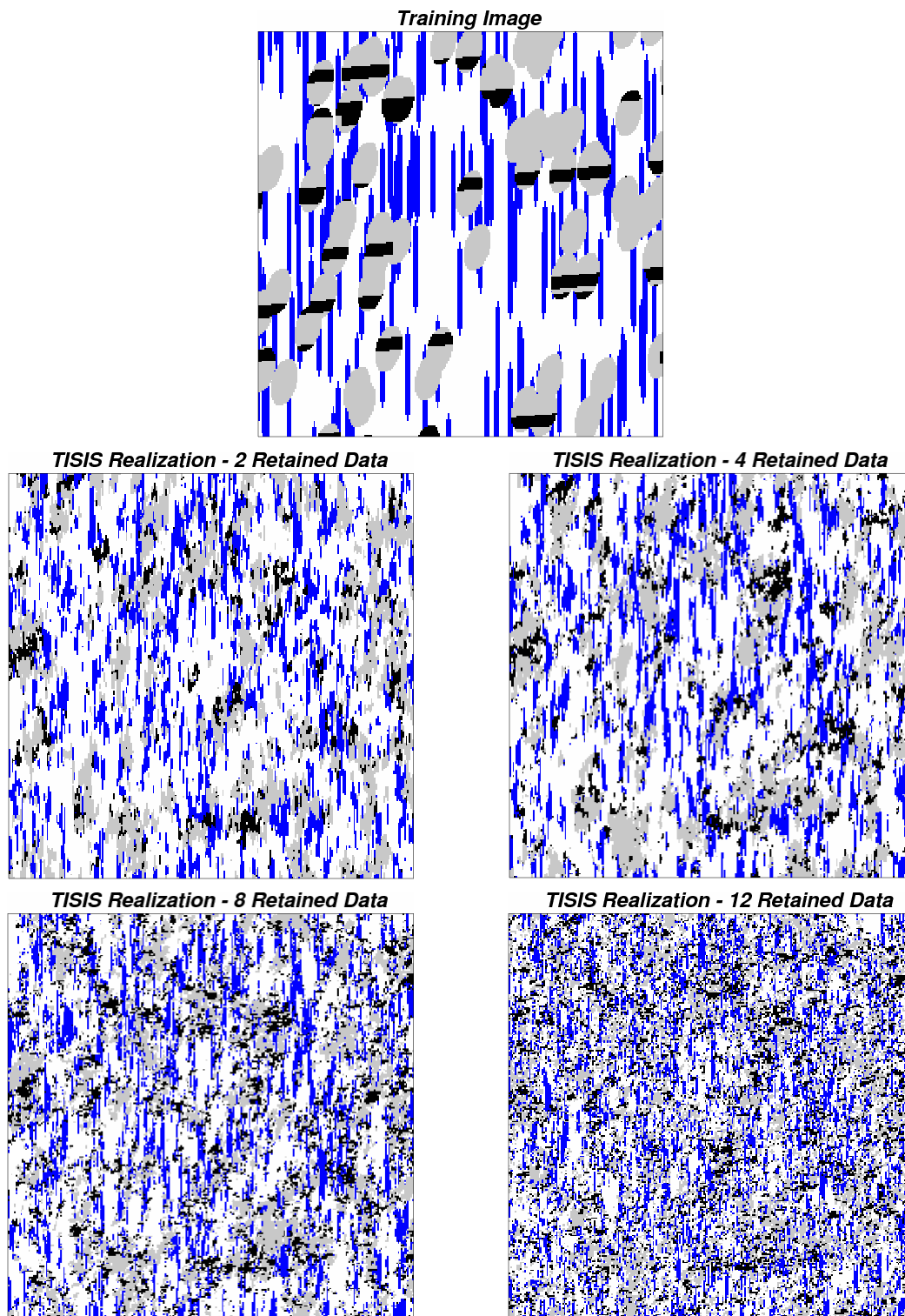


Figure 3: 256x256x1 training image with four facies (top); TISIS realizations using different numbers of retained data (lower).

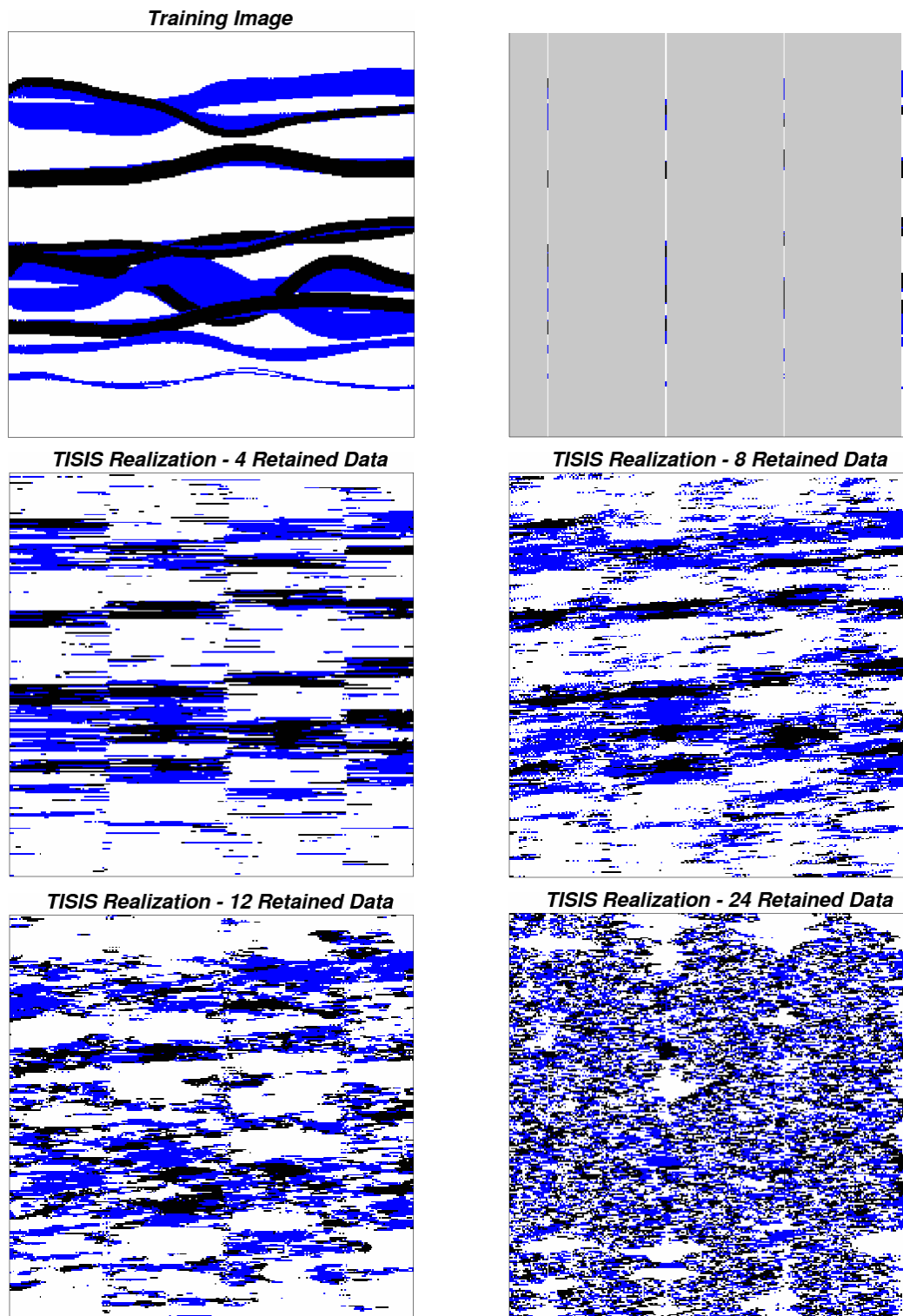


Figure 4: 256x256x1 training image with three facies (top left); conditioning data (top right); conditional TISIS realizations using different numbers of retained data (lower).

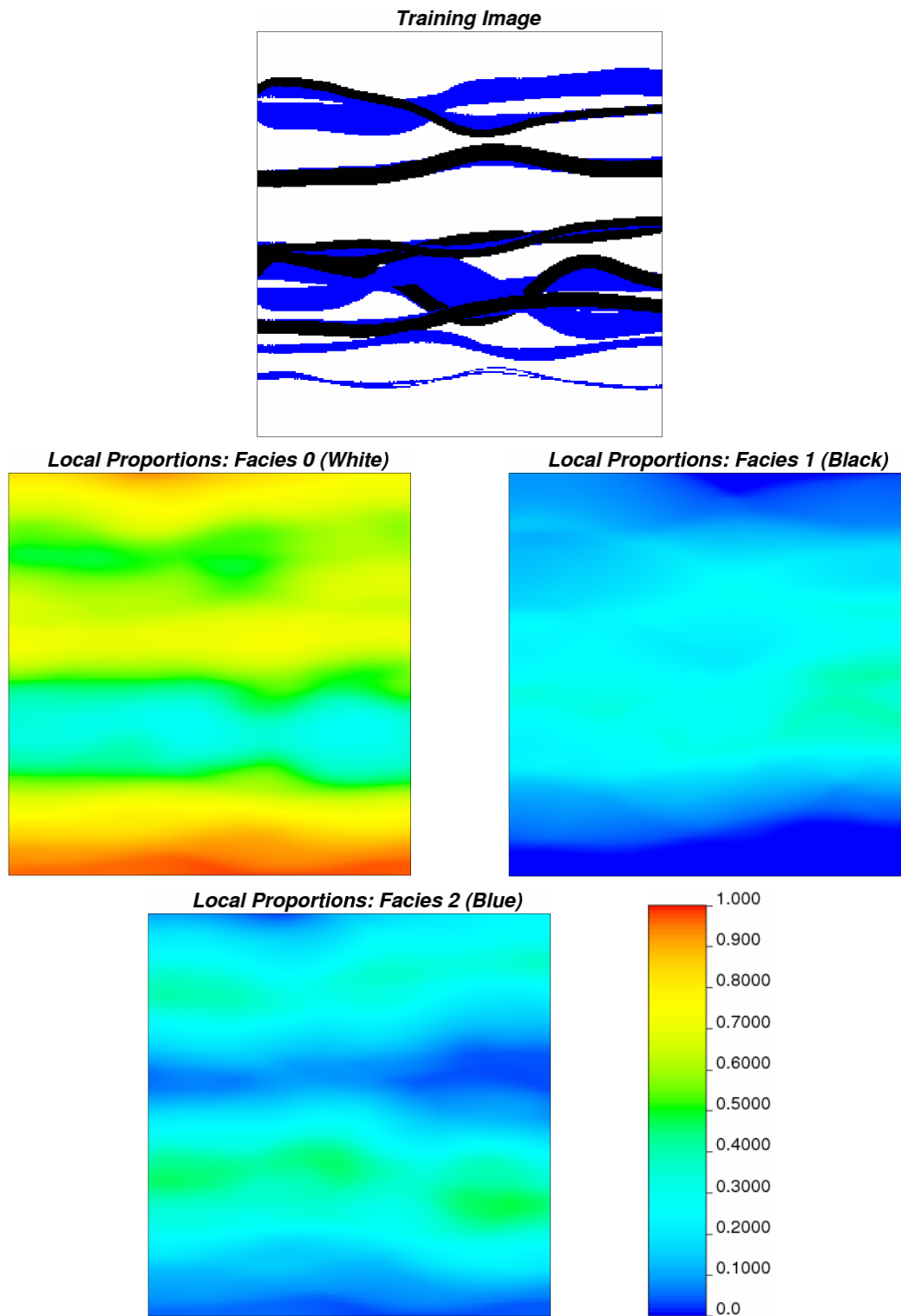


Figure 5: 256x256x1 training image with three facies (top); Locally varying proportions of the different facies in the training image.

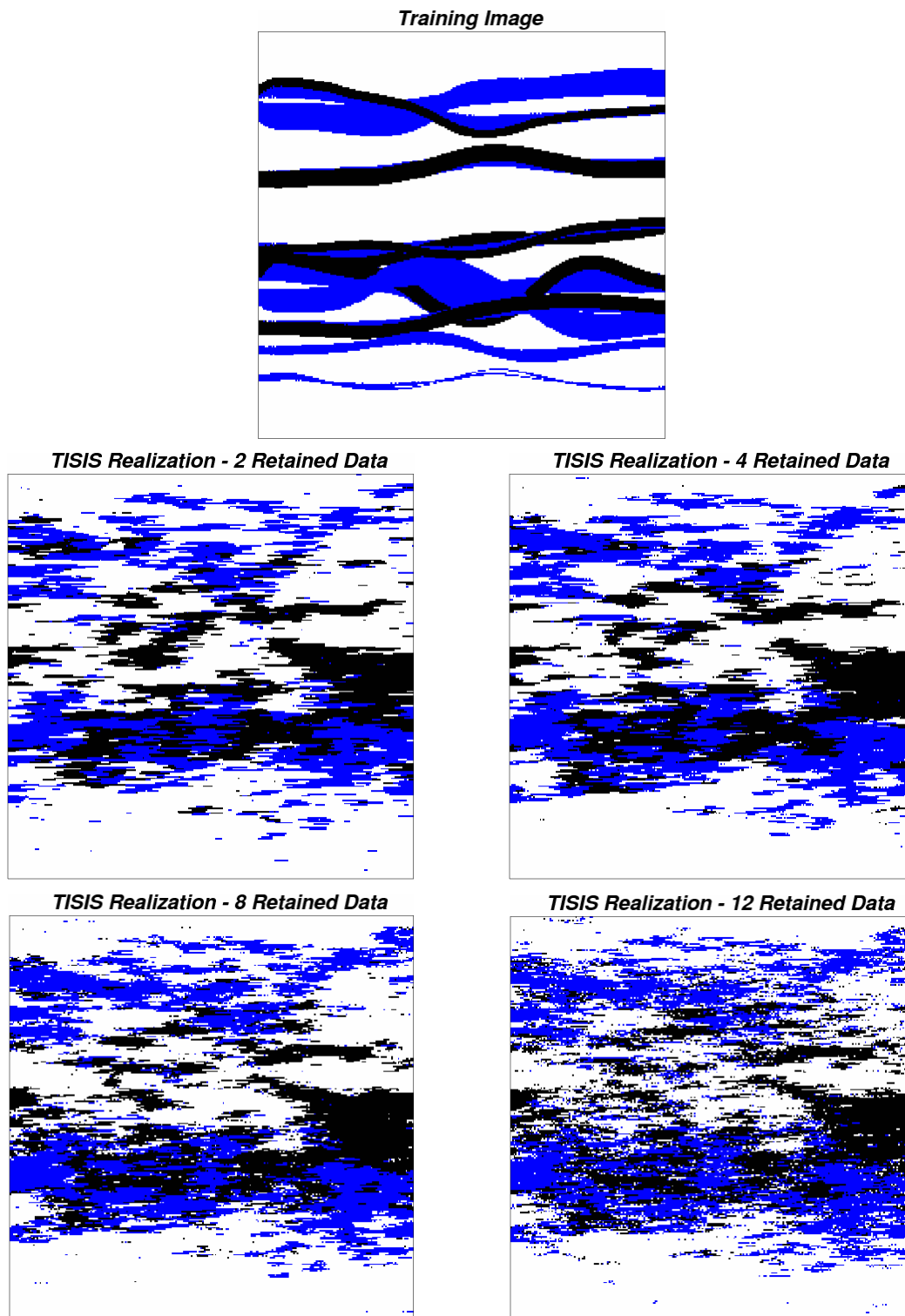


Figure 6: 256x256x1 training image with three facies (top); conditioning data (top right); TISIS realizations using the LVM model shown in Figure 5, retaining different numbers of data (lower).

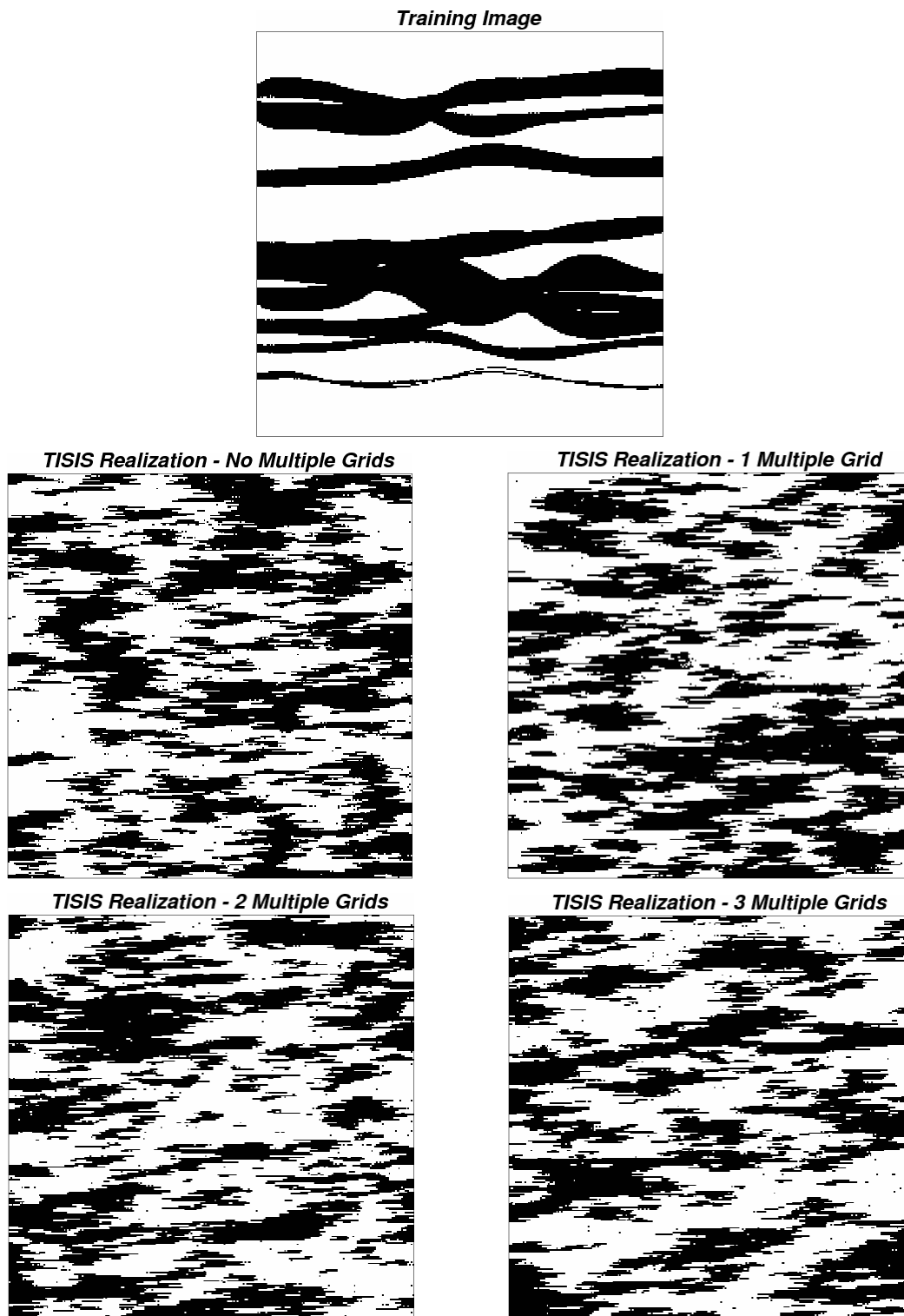


Figure 7: Multiple-grid example. 256x256x1 training image with two facies (top); TISIS realizations using different numbers of multiple grids (lower).