# Direct Pattern-based Simulation of Non-Stationary Geostatistical Models

Mehrdad Honarkhah and Jef Caers
Department of Energy Resources Engineering
Stanford University

## Abstract

Non-stationary models often capture better spatial variation of real world spatial phenomena than stationary ones. However, the construction of such models can be tedious as it requires modeling both statistical trend and stationary stochastic component. Non-stationary models are an important issue in the recent development of multiple-point geostatistical models. This new modeling paradigm, with its reliance on the training image as the source for spatial statistics or patterns, has had considerable practical appeal. However, the role and construction of the training image in the non-stationary case remains a problematic issue from both a modeling and practical point of view. In this paper, we provide an easy to use, computationally efficient methodology for creating non-stationary multiple-point geostatistical models, for both discrete and continuous variables, based on a distance-based modeling and simulation of patterns. In that regard, the paper builds on pattern-based modeling previously published by the authors, whereby a geostatistical realization is created by laying down patterns on the simulation grid, such that the simulated patterns are consistent (in terms of a similarity definition) with any previously simulated ones. In this paper we add the spatial coordinate to the pattern similarity calculation, thereby only borrowing patterns locally from the training image instead of globally. The latter would entail a stationary assumption. Two ways of adding the geographical coordinate are presented, 1) based on a functional that decreases gradually away from the location where the pattern is simulated and 2) based on an automatic segmentation of the training image into stationary regions. Using ample 2D and 3D case studies we study the behavior, in terms of spatial and ensemble uncertainty of the generated realizations.

## 1. Introduction

Most geostatistical methodologies rely on some form of stationarity to spatial model formulation, whether such approach is variogram-based, Boolean or multiple-point. In fact, one could easily argue that such assumption of stationarity is at the very foundation of any statistical modeling. To accommodate the reality that most actual phenomena require some non-stationary modeling element, models are often decomposed into a trend component and a stationary stochastic component. Trends can either be deterministic such as imposed changes in mean (Goovaerts, 1997), non-stationary two-point models (Chiles and Delfiner, 1999), changes in affinity or angular direction variations of variograms (Xu, 1996) or any other feature of the local spatial field. Trend variations can be continuous or a trend map can consist of discrete zones (regions) in which stationary modeling is performed (Caers and Zhang, 2004). Trends can also be made part of the estimation procedure such as in Kriging with trend (Chiles and Delfiner, 1999). Most approaches assume that "trend" is some lower-order or low-variance statistic while the stochastic component is highly varying or of higher-order (such as multiple-point statistics deduced from training images).

Specific to multiple-point geostatistics (MPS), a set of approaches have been developed to model non-stationarity. In MPS, the training image is considered to be a "container" from which stationary higher order statistics (conditional probabilities, Strebelle, 2002; higher-order cumulants, Dimitrakopoulos et al., 2010; patterns, Arpat and Caers, 2007; Honarkhah and Caers, 2011; data events, Mariethoz et al., 2010) are extracted and then combined in various ways to create non-stationary models. An alternative is to make one large training image consisting of zones that are stationary by dividing the training image into smaller subzones (Arpat and Caers, 2007; de Vries et al., 2009; or using feature based methods such as in Boucher, 2009) and then extracting statistics from each subzone. Another approach relies on the definition of auxiliary variables (Chuganova and Hu, 2008); modeling how certain statistics vary spatially. Again such auxiliary variables are typically a low-variance spatial variable. In summary, all these techniques rely, in a broad sense, on a decomposition as follows

*Non-stationary model = low-variance trend model + high variance stationary model*

While such decomposition has proven practical in many cases, it may not apply to all real world spatial phenomena and hence limit modeling capabilities. Often, the trend/region definition is taken as deterministic, possibly understating uncertainty. Moreover, the modeler has the arduous task of modeling both trend and stochastic component, which often hampers the wide-spread application of these techniques. More specifically, in the context of training images, it is often not intuitive to the modeler who is an expert geologist (and not an expert geostatistician) what this stationary training image model is, or, how it should be constructed. We argue that not all trends are in the form of a mean, direction or other simplified spatial feature.

In this paper, we add a new approach to the non-stationary geostatistical modeling methodologies. In particular, we consider the case where a training image is constructed from a real-world phenomenon and hence, requires almost by definition a non-stationary modeling approach. Note that we do not agree with the notion often mentioned of a "non-stationary training image". The assumption of stationarity is a model decision and not a property of an image. The 3D image is whatever is given. However, when extracting patterns, cumulants or statistics by scanning the entire image, an assumption of stationarity is made. We want to reduce relying on this assumption in this paper or at least provide more modeling flexibility.

Our approach is pattern-based and follows, for the stationary modeling approach, the ideas outlined in Arpat and Caers (2007) and Honarkhah and Caers (2011).  In a first section we will therefore review basic concepts in pattern-based stochastic modeling, in particular, the distance-based approaches to such modeling. Then, by adding simply the geographic coordinate to the pattern vector, we present two computationally efficient methods for generating directly non-stationary discrete or continuous stochastic models, without the need for any explicit decomposition. Ample examples are used to study the properties of the methods presented.

## 2. Methodology

### 2.1 Distance-based pattern classification and simulation

Before developing a non-stationary modeling approach, we first review important elements of the stationary pattern modeling and simulation, *dispat*, published in Honarkhah and Caers (2011). The *dispat* (DIStance-based PATtern simulation) builds further on the idea of *simpat* (Arpat and Caers, 2007) where instead of extracting and modeling statistics, one extract patterns from a training image **ti** using a fixed template *T*. *Dispat* solves some important challenges remaining in *simpat* and that is 1) finding the most similar pattern to a data-event in an efficient way and 2) defining the size of an appropriate template *T* (Honarkhah, 2011). The *dispat* algorithm consists of two stages: 1) constructing a searchable pattern database $\mathbf{patdb}_T$ and 2) stochastic simulation of patterns. Stochastic simulation is much the same as other pattern-based or distance-based codes such as *simpat*, *filtersim* (Zhang, 2006) and also direct sampling (Mariethoz et al, 2010). One starts with an empty grid, gradually filling the grid with patterns extracted from the training image. At each grid-location **u**, some previously simulated grid cells may be present (a data-event or $\mathbf{dev}_T(\mathbf{u})$) and therefore a pattern that is similar (in some distance-based sense) with the current data-event must be found and pasted at that location. In *dispat* patterns are organized by means of a distance between any two patterns. The choice of distance depends on the type of patterns available in the training image. Distances allow for 1) dimensionality reduction using multi-dimensional scaling and 2) clustering of patterns into groups and 3) quick searching for a pattern matching the data-event. Such distance in general is expressed as

$$d_{pat}\langle\cdot,\cdot\rangle = d\left\langle \mathbf{dev}_T(\mathbf{u}), \mathbf{pat}_T^k \right\rangle$$

The CPU/RAM performance of *dispat*, its pattern reproduction and data conditioning are some of the attractions of the algorithm (see Honarkhah, 2011 for details).

## 2.2 Non-stationary simulation by spatial similarity

Consider a training image **ti** requiring non-stationary modeling. Instead of just storing all patterns of the training image in the pattern database $\mathbf{patdb}_T$, we also store their spatial grid locations **u**=(*i,j,k*) in another database, called location database denoted by $\mathbf{locdb}_T$. Hence, the joint pattern database is no more a location-independent matrix. Consider the pattern at location **u**. The corresponding pattern that is stored in the *k*-th row of the pattern database, $\mathbf{patdb}_T$, is $\mathbf{pat}_T^k = \mathbf{ti}_T(\mathbf{u})$. In addition to storing the pattern, the *k*-th row of location database**,** $\mathbf{locdb}_T$, is $\mathbf{loc}_T^{\mathrm{k}}$ =**u**.

The simulation proceeds similar to the stationary pattern-based technique outlined above, the difference lies in the distance definition. In order to find a pattern to be pasted on the simulation grid, one will define a distance between the data event and the patterns within the pattern database $\mathbf{patdb}_T$, as well as, a distance between the location $\mathbf{u}$ and all the locations in the location database $\mathbf{locdb}_T$, namely

$$d_{loc}\left\langle \cdot,\cdot \right\rangle = d\left\langle \mathbf{u},\mathbf{loc}_T^k \right\rangle$$

We propose now to search patterns based on the following combined distance

$$d_{nonstat} = (1-\omega)d_{pat} + \omega d_{loc}$$

with ω a weight between [0,1] with ω=0 entailing stationary simulation and ω=1 returning the training image itself (in a way being "perfectly" non-stationary is equal to being perfectly deterministic). The idea of including the spatial location of the patterns into the similarity search causes the patterns to not only be ranked according to their similarity with the data event but also be ranked according to their spatial closeness to the simulated node $\mathbf{u}$. One should note here another important difference with stationary simulation: the training image grid needs to be the same as the simulation grid. Indeed, only an assumption of stationarity allows for such difference in grid sizes.

We propose a method called spatial-similarity method or SSM that assumes a smooth and convex distance weighting of patterns. In other words, all the patterns of the database are included in the similarity search, but they are weighted according to their geographic distances (on the original training image grid) to the location of the visited node (on the realization grid). For a 2D grid, the smooth and convex function of distances that is used to influence the pattern similarities is shown in Figure 1 (top). An example procedure for a simple training image and one grid node location $\mathbf{u}$ is shown in the remainder of Figure 1 : a node location $\mathbf{u}$ on the realization grid is being simulated. The data event is extracted and shown below the realization grid. Two distance vectors are depicted in a gray color bar; one for distances between patterns of the training image and the data event, and the other for the distance between location $\mathbf{u}$ and all locations of the training image. These two distance vectors are combined using ω=0.55. As can be seen in the 2D depiction of the non-stationary distance $d_{nonstat}$, the most appropriate pattern is obtained at the position with the joint minimum distance.

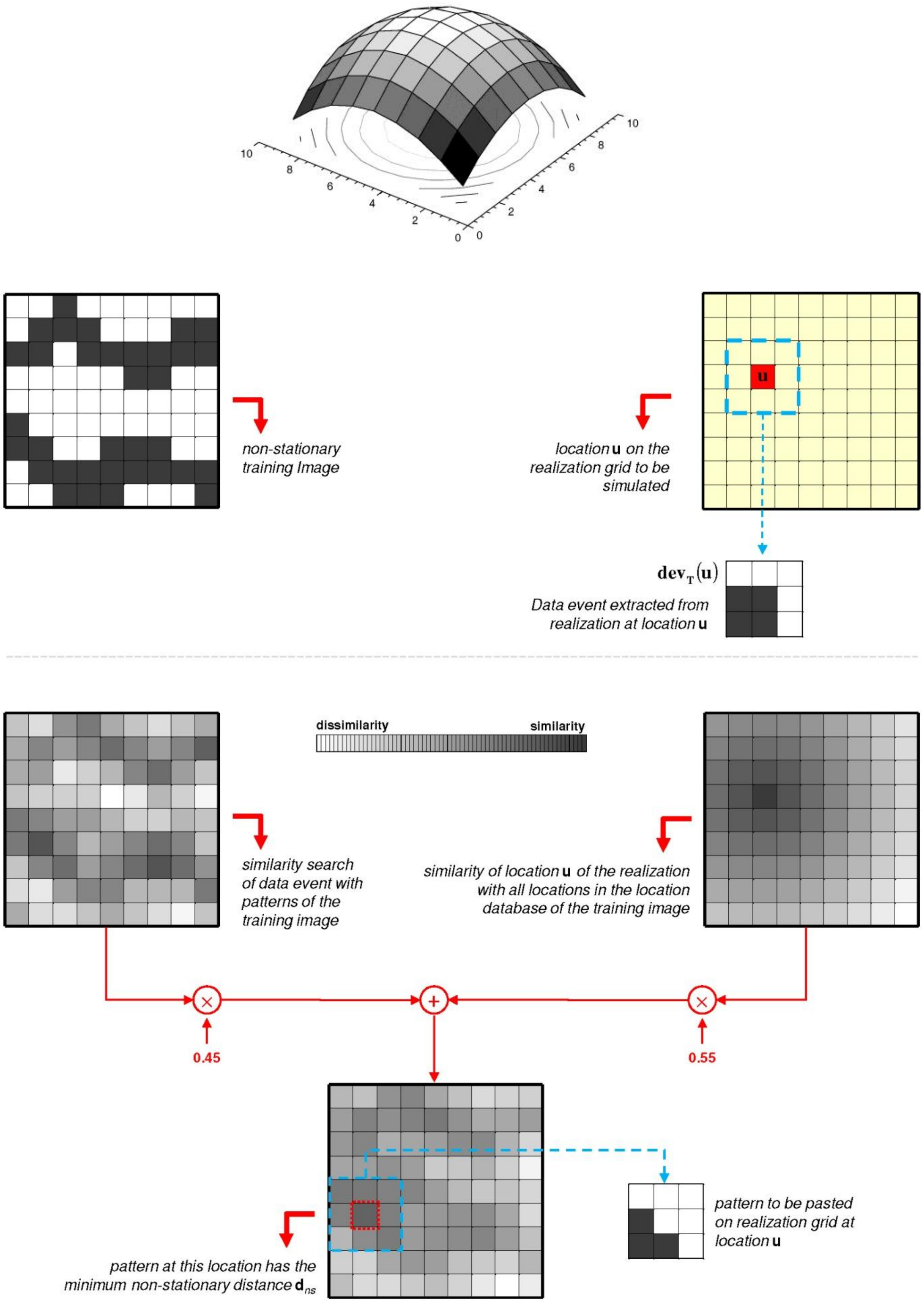

10
8
6
4
2
0
0
2
4
6
8
10
non-stationary training Image
u
location u on the realization grid to be simulated
dev_T(u)
Data event extracted from realization at location u
dissimilarity
similarity
similarity search of data event with patterns of the training image
similarity of location u of the realization with all locations in the location database of the training image
0.45
0.55
pattern at this location has the minimum non-stationary distance d_ns
pattern to be pasted on realization grid at location u

Figure 1: Example application of spatial-similarity method (SSM) is provided. A training image and a corresponding node location $\mathbf{u}$ on the realization grid, that is to be simulated, are shown. The two distances for pattern similarity and location similarity are visually illustrated on the original grid. The most optimal pattern, to be pasted on the realization, is obtained by searching with $\mathbf{d}_{nonstat}$.

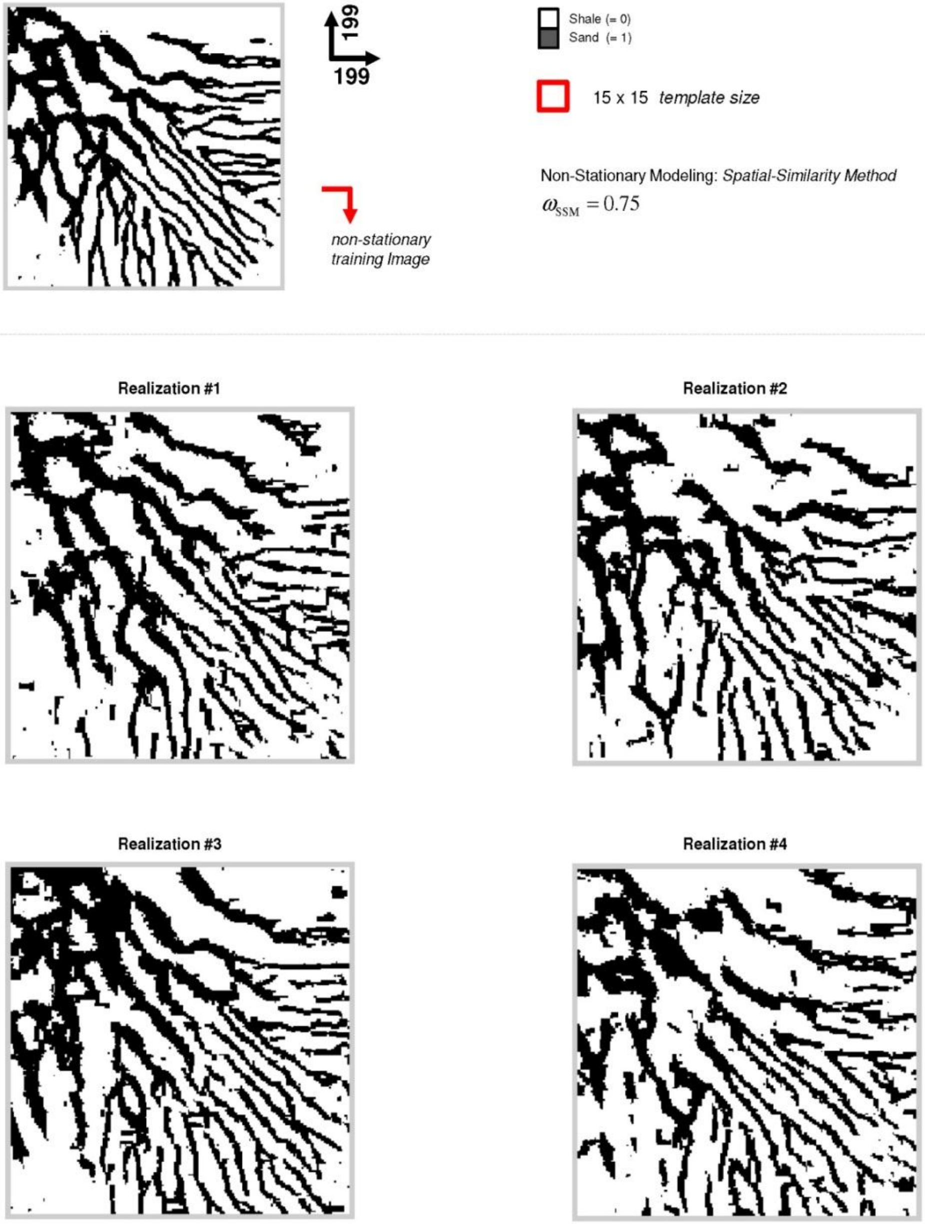
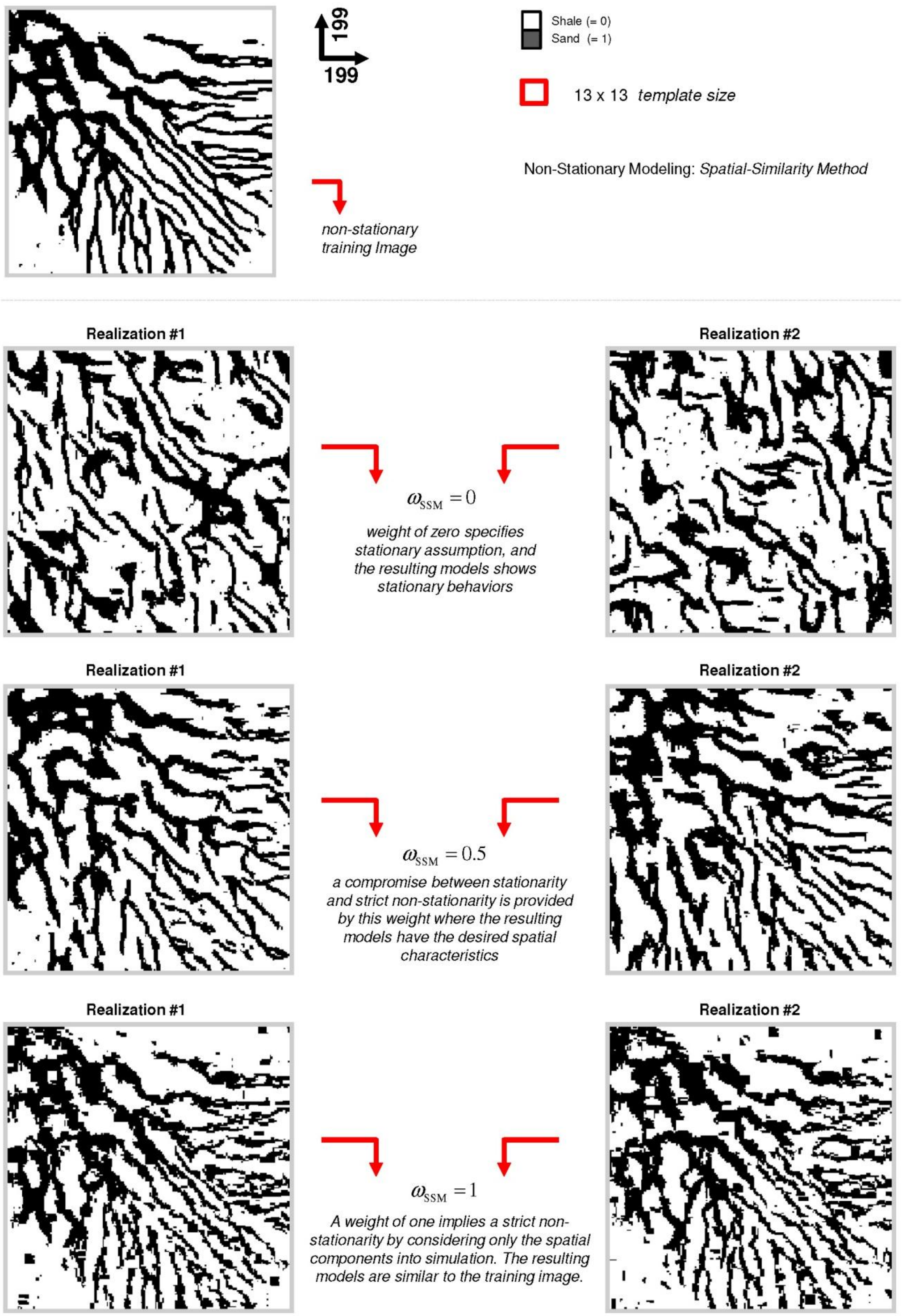

199
199
Shale (= 0)
Sand (= 1)
15 x 15 template size
Non-Stationary Modeling: Spatial-Similarity Method
$\omega_{\text{SSM}} = 0.75$
non-stationary training Image
Realization #1
Realization #2
Realization #3
Realization #4

Figure 2: Application of Spatial-Similarity Method (SSM) on fluvial fan-deposit training image of size $199 \times 199$. Template size is $15 \times 15$ with a multi-grid level of 3. The stationarity level is set to $\omega = 0.75$. Four realizations are shown to demonstrated non-stationarity and stochasticity of the proposed algorithm.



*weight of zero specifies stationary assumption, and the resulting models shows stationary behaviors*

*a compromise between stationarity and strict non-stationarity is provided by this weight where the resulting models have the desired spatial characteristics*

*A weight of one implies a strict non-stationarity by considering only the spatial components into simulation. The resulting models are similar to the training image.*

Figure 3: Sensitivity analysis of the non-stationarity level ω on the realizations generated by the application of Spatial-Similarity Method (SSM) on fluvial fan-deposit training image of size $\mathbf{199 \times 199}$.

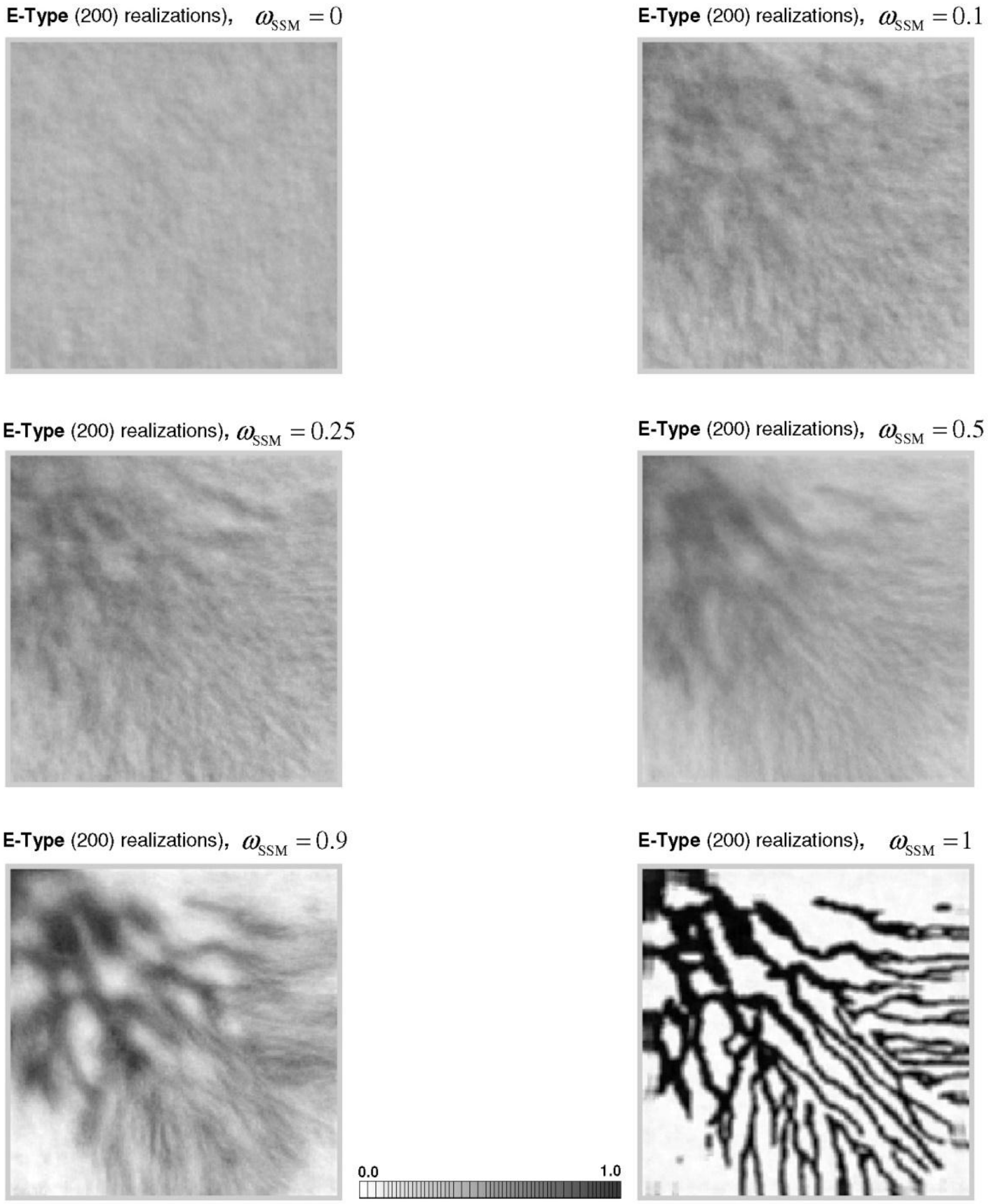


Figure 4: Sensitivity analysis on the E-types with respect to the non-stationary weight $\omega$. The training image is the same fluvial fan-deposit shown in Figure 3. Larger weight lead to increased non-stationarity models.

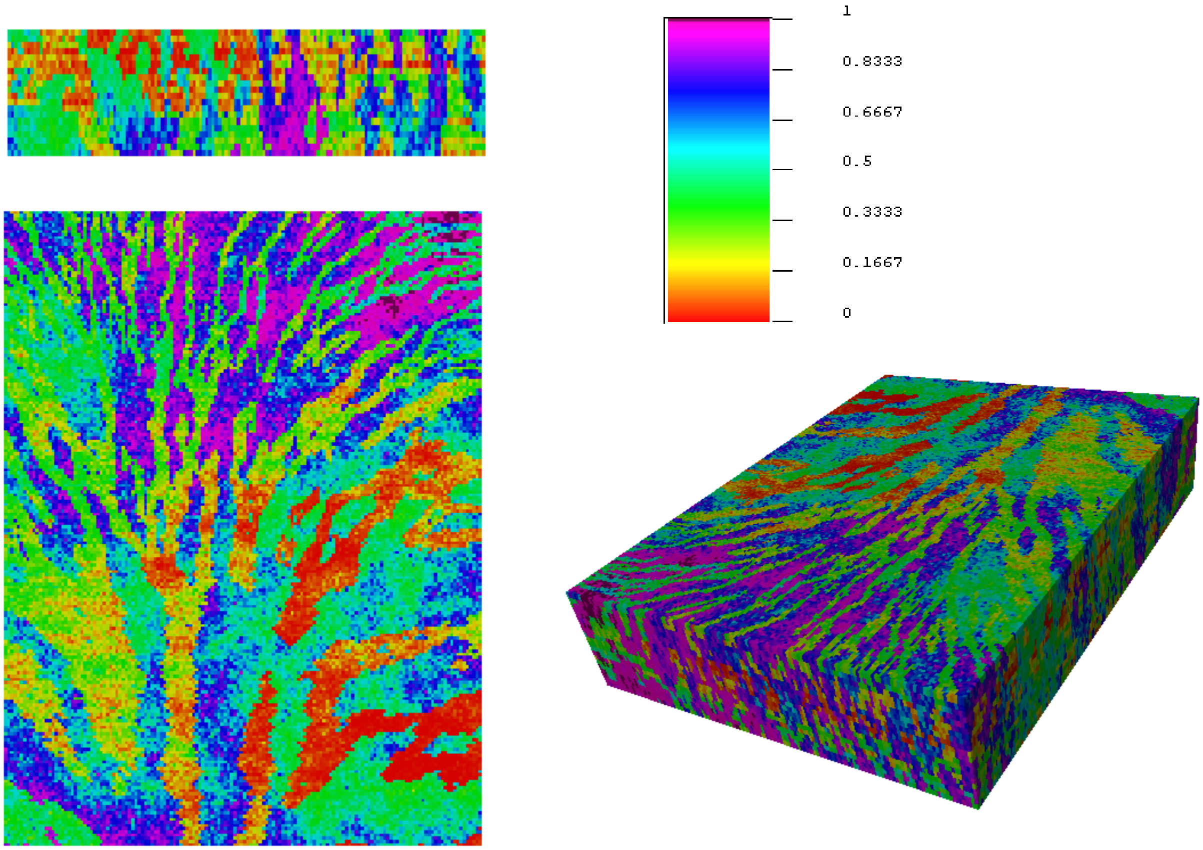


Figure 5: An example of a complex 3D non-stationary training images .

## 2.3 Examples

An example application of non-stationary modeling with the proposed SSM approach is depicted in Figure 2. The template size is 15×15, and ω=0.75. Four different realizations are shown. It can be observed that the realizations are indeed non-stationary, and there is variability between them. For example, the thick channels originating from the top left corner of the realizations have varying spatial continuities. Now, the effect of the non-stationarity weight ω will be investigated. Three values for the non-stationary weight are explored: ω ={0,0.5,1}. The realizations for each ω are shown in Figure 3.

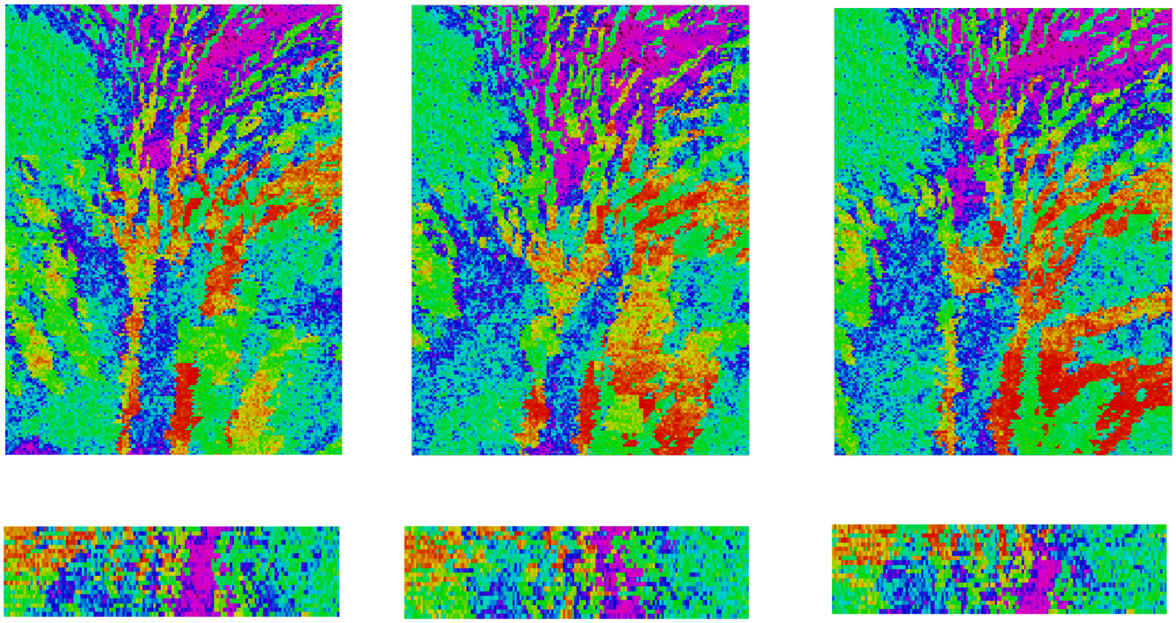


Figure 6: Three non-stationary realizations generated from the training image shown in Figure 5 using the SSM method.

Similarly, one can study the effects of ω on the E-types (Ensemble averages). 200 realizations are generated for each case, and the E-type is computed. Figure 4 shows the ensemble averages of the realizations. The E-type corresponding to the weight of zero shows no structure. However, as the weight increases to 0.1, some structures become apparent in the E-type. Further increasing the weight for spatial components reveals more of the non-stationarity of the training image. Finally, a value of one for ω leads to an E-type equal to the training image itself.

Consider in Figure 5 a 3D case example that illustrates the versatility of the approach in a real-world setting. Consider that considerable effort has been made in constructing a 3D training image such as show in Figure 5. In current MPS approaches, one would need to decompose this 3D image into a trend (either as auxiliary variable or regions) and a stationary training image. It would not be immediately clear what such training stationary image would be. Additionally, most current MPS approaches only work for discrete variables. With the proposed method, we simply take this 3D image as a non-stationary model and generate directly realizations. Only two parameters needs to be specified: ω=0.7 and the template size = 21x21x3 (three multiple grid simulation). The variable being simulated here has a uniform distribution: hence mean = 0.5 and variance = 0.083.

Figure 6 shows three realizations, each displaying similar features as the training image. To assess the variability produced, we plot the ensemble average and variance in Figure 7. In areas with high channel variability, such as in the middle of the 3D training image, we observe considerable variance between realizations, while in areas with low pattern variability, such as in the top left part of the 3D training image, we observe low variance. It appears that a property of this approach is that pattern variability is translated into between-realizations variability. In other words, given a training image with little variance over the domain will result in realizations with low ensemble variance. For this large case the CPU time to generate one realization is around 30sec for a grid size of 150x200x15 (600.000 cells), the RAM used about 50Mb, hence, from a practical computational point of view, this method, has considerable appeal.

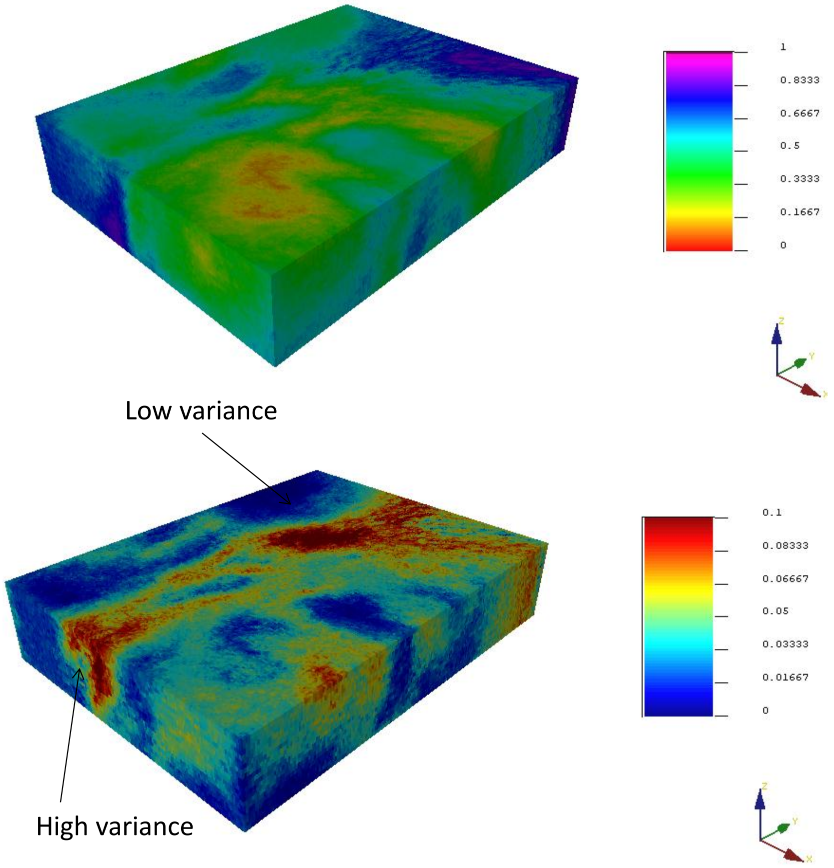


Figure 7: The ensemble average (shown in top), and the variance (shown in bottom) of a set of realizations generated using SSM method on a complex non-stationary 3D training image.

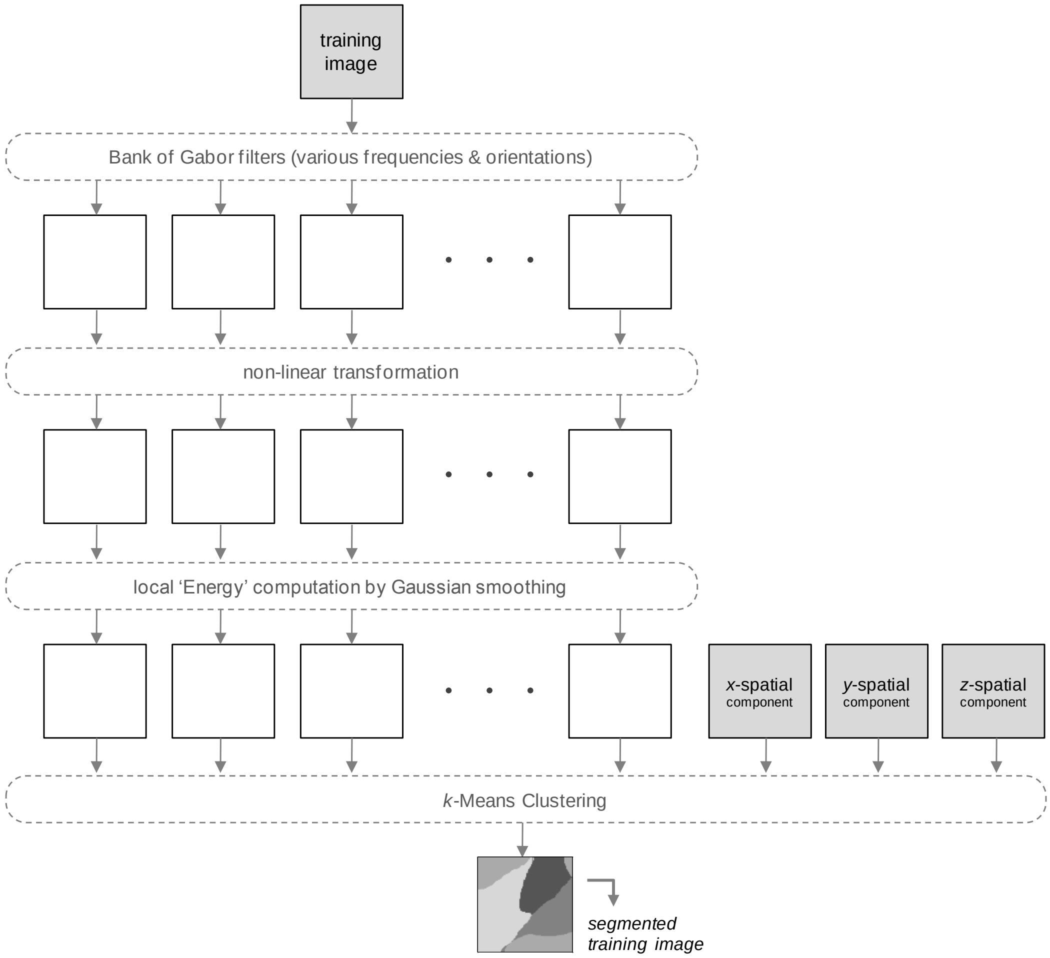

training image
Bank of Gabor filters (various frequencies & orientations)
non-linear transformation
local 'Energy' computation by Gaussian smoothing
x-spatial component
y-spatial component
z-spatial component
k-Means Clustering
segmented training image

Figure 8: Flowchart of Automatic training image segmentation using a bank of Gabor filters, with the corresponding non-linearity filters and the Gaussian smoothing. The k-means clustering algorithm is used for defining the stationary regions.

## 2.4 Non-stationary simulation by region similarity

### 2.4.1 Introduction

SSM simulation provides a fairly general approach to non-stationary simulation. In a second part to this paper we provide an approach that is more specific to the case where non-stationary modeling is required due to need to model different patterns in different regions of the model. So, in fact, we model a form of quasi-stationarity. In the more traditional approaches (Caers and Zhang, 2004; deVries et al., 2009) one first creates fixed regions (trend), then, assigns a different training image to each region (higher-order stochastic component). In this section we propose an Automatic Segmentation Method (ASM) to non-stationary modeling that takes as input a training image and automatically creates regions based on that training image, then, simulates patterns within that region using the segmented training image. We will use Gabor filters for segmenting a training image into stationary zones. The overal procedure consists of four main steps (see Figure 8), namely (1) applying Gabor filter banks, (2) energy (feature) extraction from filter results, (3) clustering to create compact regions, and (4) non-stationary simulation.

### 2.4.2 Gabor filtering

Gabor filters have been extensively used in various image and vision processing applications; such as, handwritten character recognition (Hamamoto et al. 1998), face recognition (Ayinde and Yang 2002), image segmentation (Jain and Farrokhnia 1991; Dunn and Higgins 1995), feature detection (Manjunath et al. 1996), edge detection (Mehrotra et al. 1992). Besides the advantages of Gabor filters in signal processing, there is considerable evidence on their physiological similarities with the visual processing mechanisms of several mammalians (Marcelja 1980; Daugman, 1984; Daugman 1993), hence their appeal in the spatial domain of pattern processing.

At the foundation of this pattern processing method lies the Gabor function which for a time signal is expressed as

$$g(t) = \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right)\cdot\exp(iwt)$$

where $\sigma$ is the resolution of the analysis, or simply, the spread of the Gaussian function whose center is located at $t$, and $w$ is the position of the signal in the frequency domain. Two dimensional signals (images) can be represented by a bank of Gabor functions (Gabor filters). A 2D Gabor filter in the spatial domain *(x,y)* is defined as,

$$g(x,y) = \exp\left(-\frac{1}{2}\left[\frac{x'^2}{\sigma_x^2}+\frac{y'^2}{\sigma_y^2}\right]\right)\cdot\exp(i2\pi fx')$$

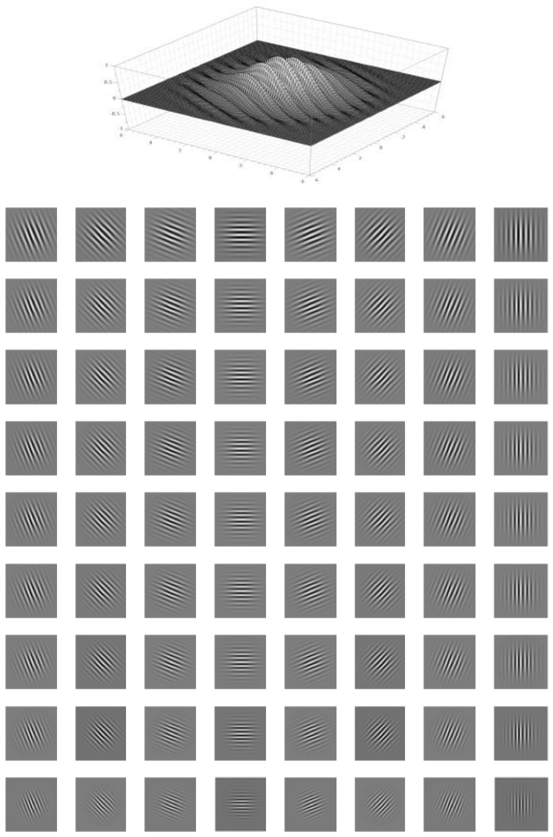
where $\sigma_x$ and $\sigma_y$ represent the spatial variances of the Gaussian kernel, determining the size of the support of Gabor filters. $f$ denotes the spatial frequency of the sinusoidal function. The parameters $x'$ and $y'$ represent the rotation of the Gabor filter as follows:

$$\begin{aligned} x' &= x\cos(\theta) + y\sin(\theta) \\ y' &= -x\sin(\theta) + y\cos(\theta) \end{aligned}$$

where $\theta$ is the orientation of the Gabor filter.

Figure 9 illustrates an example of a Gabor filter in a two-dimensional spatial space.

Figure 9: The real component of a 2D Gabor function, with parameters $\boldsymbol{\sigma_x^2 = \sigma_y^2 = 20}$, $\boldsymbol{\theta = \frac{\pi}{6}}$, and $\boldsymbol{f = 1}$, plotted for $\boldsymbol{x, y} \in [\boldsymbol{-6, +6}]$ (shown in top), and in bottom, A bank of real-valued even-symmetric Gabor filters is shown. Eight different orientations with angular separation of 22.5 ° are shown horizontally, from left to right. Furthermore, nine different frequency values are shown in a descending order from top to bottom.



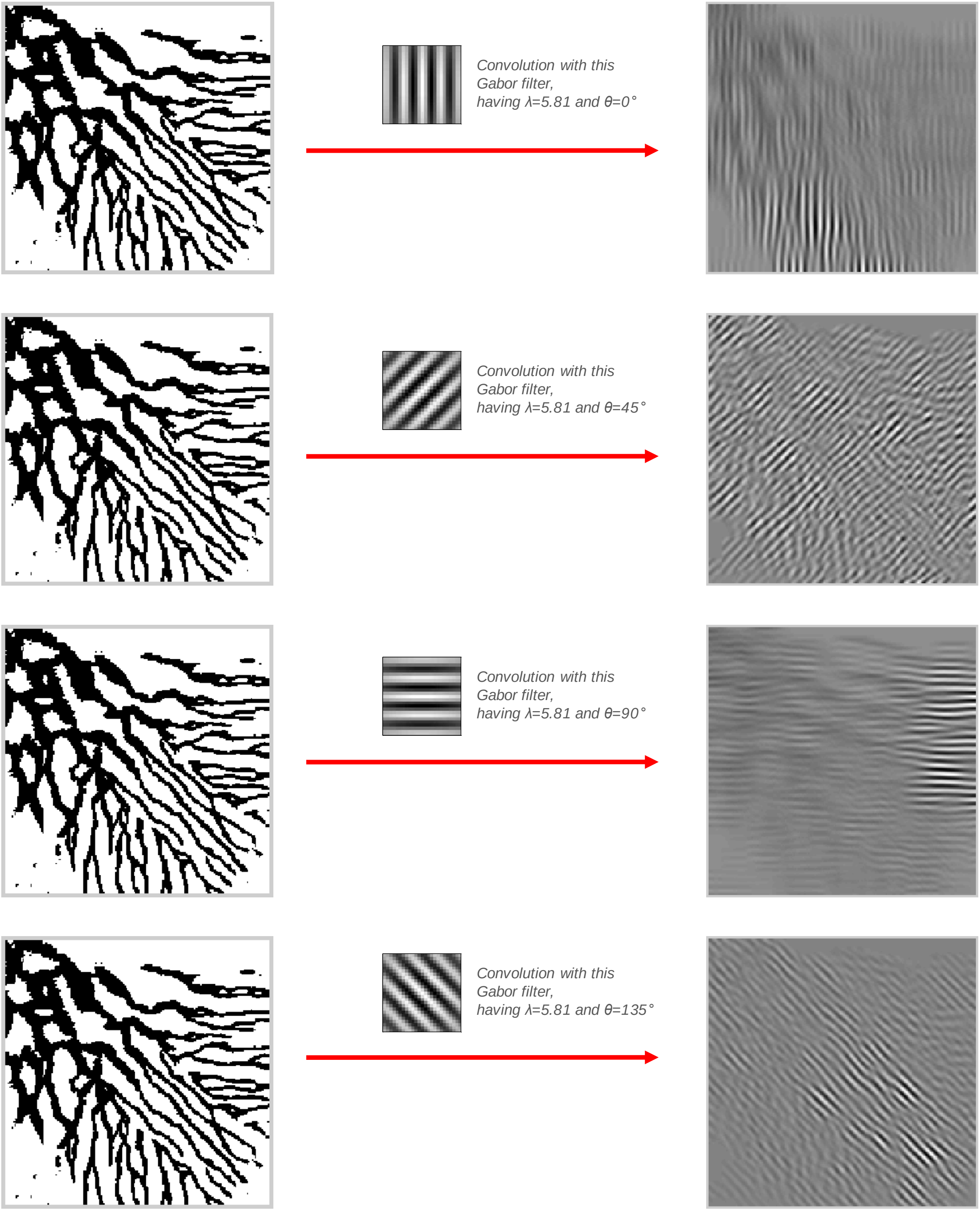


Figure 10: Application of four Gabor filters with orientation angles of $\{\mathbf{0}^{\circ}, \mathbf{45}^{\circ}, \mathbf{90}^{\circ}, \mathbf{135}^{\circ}\}$, and a constant small frequency of $\boldsymbol{f} = \mathbf{0.17}$. Each filter is distinguishing a different set of features in the training image.

In order to use Gabor filters for partitioning a training image into stationary zones, one needs to suitably choose the corresponding parameters, such as frequency and orientation. Since these parameters are hard to define a-priori, we will use a bank of Gabor filters, similar to Malik and Perona (1990) and Jain and Farrokhnia (1991). A filter bank generally consists of filters with different frequencies and orientations. For simplicity, in the proposed ASM method, only a set of real-valued even-symmetric filters will be considered. Additionally, the *x* and *y* scale of the Gaussian envelope are set to be equal. The choice on the orientations in 2D is trivial; the more, the better. We found, by experimentation, that the following set of angle is sufficient

$$\theta = \{0^\circ, 22.5^\circ, 45^\circ, 67.5^\circ, 90^\circ, 112.5^\circ, 135^\circ, 157.5^\circ\}$$

For frequency values we use the recommendation of Zhang et al. (2002), leading to the bank of 72 2D Gabor filters provided in

Figure 9. Convolving each filter with the training image can be achieved by a multiplication in the Fourier domain. However, the only concern in convolution is how to process pixels at the boundary of the image. In the ASM methodology, we will reflect the training image at the boundaries to obtain a larger sized image. This allows obtaining the filter responses at all locations on the training image grid, irrespective to the size of the filters themselves.

An example application of Gabor filter banks is performed on the training image of Figure 3. Only a limited set of filters and their responses are shown in Figure 10, namely four different orientations with 45 ° separation angles and a constant frequency of *f*=0.17. The observed ripple structure is due to the sinusoidal component of the filter with positive and negative values. Due to this ripple-like structure, we apply a non-linear transformation on each filtered image to an energy component, as follows:

$$\beta(t) = \tanh(\alpha t) = \frac{1 - e^{-2\alpha \mathrm{t}}}{1 + e^{-2\alpha t}}$$

where $t$ is the value of the filtered training image. This sigmoidal function rapidly saturates the filtered responses, and can be interpreted as blob (i.e. rectangles, ellipses, lines) detectors (Jain and Farrokhnia 1991). In this analysis, we set the value of $\alpha$ to 0.25. After non-linear transformation of the filtered images, we apply a Gaussian smoothing filter in order to extract the local energy for different features. The application of the energy extraction on some of the filtered images is shown in Figure 11. Each of these local energy images, denoted as $\mathbf{o}_i$, $i=1,\ldots m$ defines a characteristic of the non-stationary training image, such as, the orientation of the channels, or the width of the objects. Specifically, different frequencies correspond to various details observed in the training image.

### 2.4.3 Creating compact stationary regions

To define stationary regions we apply *k*-means clustering in the following vector consisting of the filter results combined with the spatial location

$$ti_{feature}(x,y,z):\left\{o_1(x,y,z),o_2(x,y,z),\ldots,o_m(x,y,z),4\frac{x}{n_x},4\frac{y}{n_y},4\frac{z}{n_z}\right\}$$

where, $n_x$, $n_y$ and $n_z$ denote the sizes of the training image in *x, y* and *z* directions. A weight of 4 is applied for the spatial component due to the differences between the number of filters in a training image and the number of spatial dimensions. This weighting will ensure that spatial adjacency is not lost during clustering. The combination of filter results and spatial coordinate provides regions that (1) consist of similar features, and (2) are not overly fragmented. The result for a choice of six stationary regions is shown in

Figure 12. However, due to random initializations in the *k*-means clustering algorithm, each run leads to a possibly different clustering result. Although the boundaries of each region may be different, however, they all have partitioned the training image into stationary regions. These subtle differences in the stationary regions, obtained because of the *k*-means convergence issues, are actually desirable, since they can introduce uncertainty into the boundary identification of the stationary regions.

### 2.4.5 Non-stationary simulation

Given the image segmentation, we now construct, from the training image, the pattern database **patdb**$_T$, and the region database, **regdb**. The region database consists of the label defining the stationary subregions. If $k$ subregions $r_k$ are modeled, then, the following relationship defines the region database:

$$\text{regdb}(\mathbf{u})=i\overset{\text{iff}}{\Leftrightarrow}\mathbf{u}\in r_i$$

Having defined these two databases, the simulation starts by randomly visiting the realization grid nodes, and extracting data events. For each data event, a pattern is selected using the following distance, which incorporates the spatial components (regions) of the training image.

$$d_{ns}(t) = d_{pat}(t) \times \mathbf{I}^{t}(\mathbf{u})$$

with

$$d_{pat}\langle \cdot, \cdot \rangle = d\left\langle \mathbf{dev}_{T}(\mathbf{u}), \mathbf{pat}_{T}^{k} \right\rangle$$

and the indicator variable $\mathbf{I}^{t}(\mathbf{u})$ is defined as follows:

$$\mathbf{I}^{t}(\mathbf{u}) = \begin{cases} 1, & if \quad \text{regdb}(\mathbf{u}_{t}) = \text{regdb}(\mathbf{u}) \\ \infty, & otherwise. \end{cases}$$

where $\mathbf{u}_t$ represents the location of the $t$-th pattern.

### 2.4.5 Examples

An application of non-stationary modeling using the ASM method is depicted in Figure 12. Six stationary subregions have been determined using the Gabor filters. Four realizations are shown. While the ASM methods works for this training image, it is more suited to those training images that can be categorized as quasi-stationary, or in other words, those that contain clearly distinct stationary regions.

The application of non-stationary modeling is applied to a fracture network. The corresponding training image is shown in Figure 13 (borrowed from Boucher 2009). The non-stationarity emerges from the three different fracture orientations. The number of stationary regions is set to three. The final segmentation is shown in the top-right corner in Figure 13. Four different realizations are generated. It can be observed that the realizations exhibit a constant orientation in each subregion. There is also a good pattern reproductivity for each orientation in the fracture network.

In Figure 14, a shallow-water tidal-dominated system will be simulated. The corresponding training image is borrowed from Harding et al. (2004). It consists of a shoreface, a lagoon, and redbed. It is a categorical training image with seven facies types. The ASM method is performed by selecting five distinct subregions. The resulting realizations are shown in Figure 14. One observes the variability between the realizations in terms of the boundaries, and the feature geometries and locations in the provided realizations.

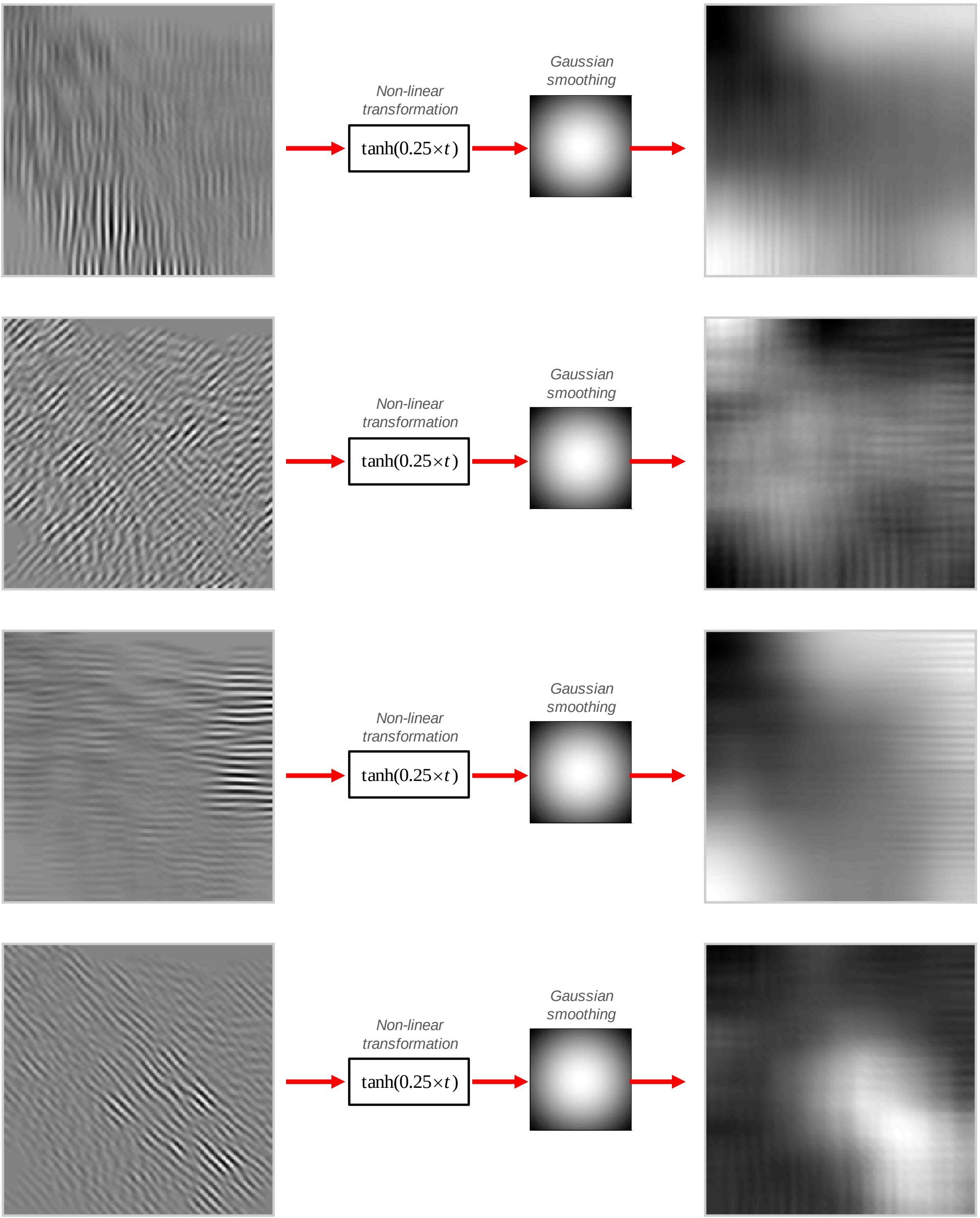


Figure 11: Application of energy extraction on the filtered images shown in Figure 10. It involves a non-linear transformation with sigmoidal function, and a smoothing filter with a Gaussian function that has three times bigger spread than the scale of its corresponding Gabor filter.

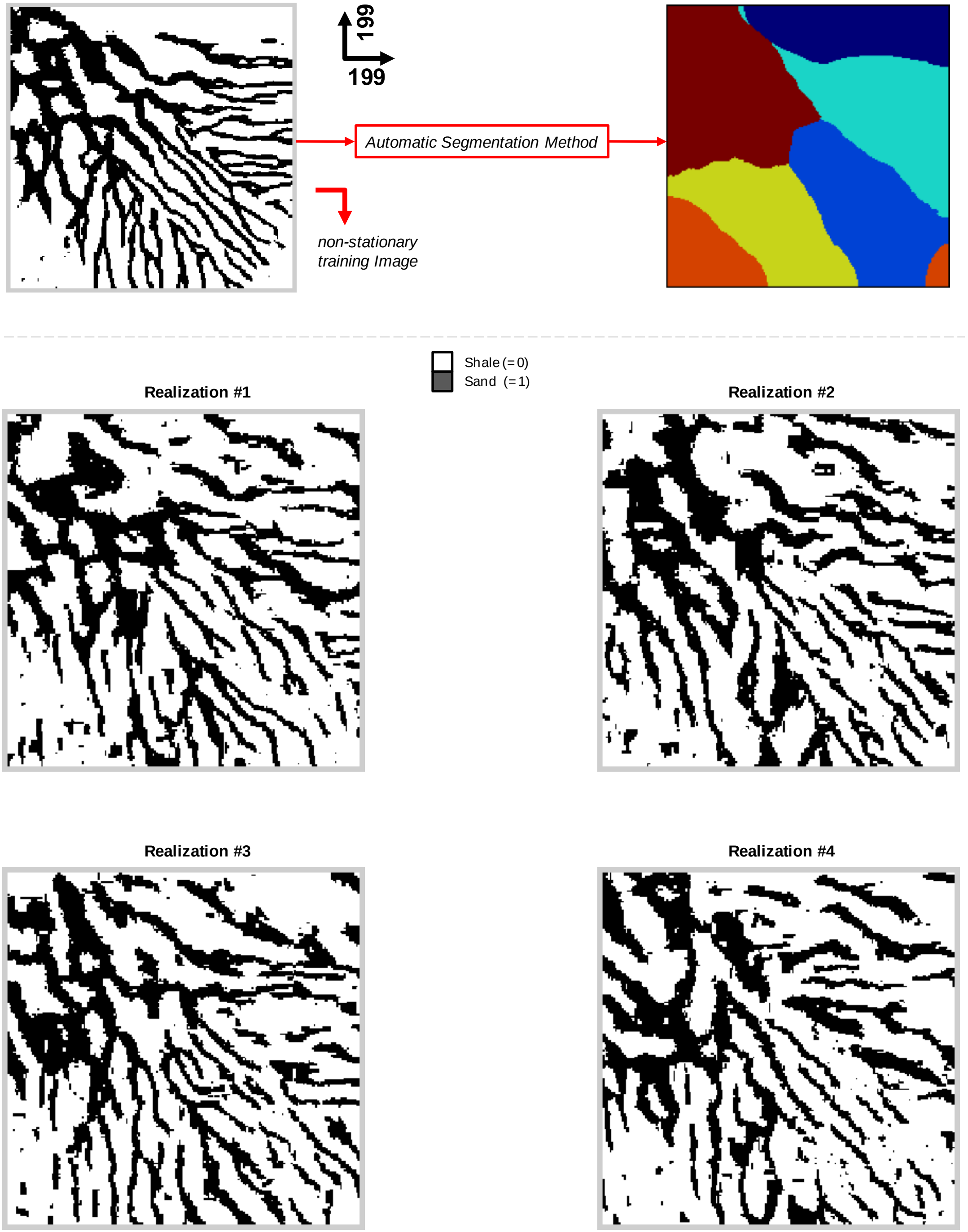


Figure 12: Application of Automatic-Segmentation Method (ASM) on fluvial fan-deposit training image of size $\mathbf{199} \times \mathbf{199}$ is investigated with a template size is $\mathbf{15} \times \mathbf{15}$ with a multi-grid level of three. There are six stationary regions defined for this training image

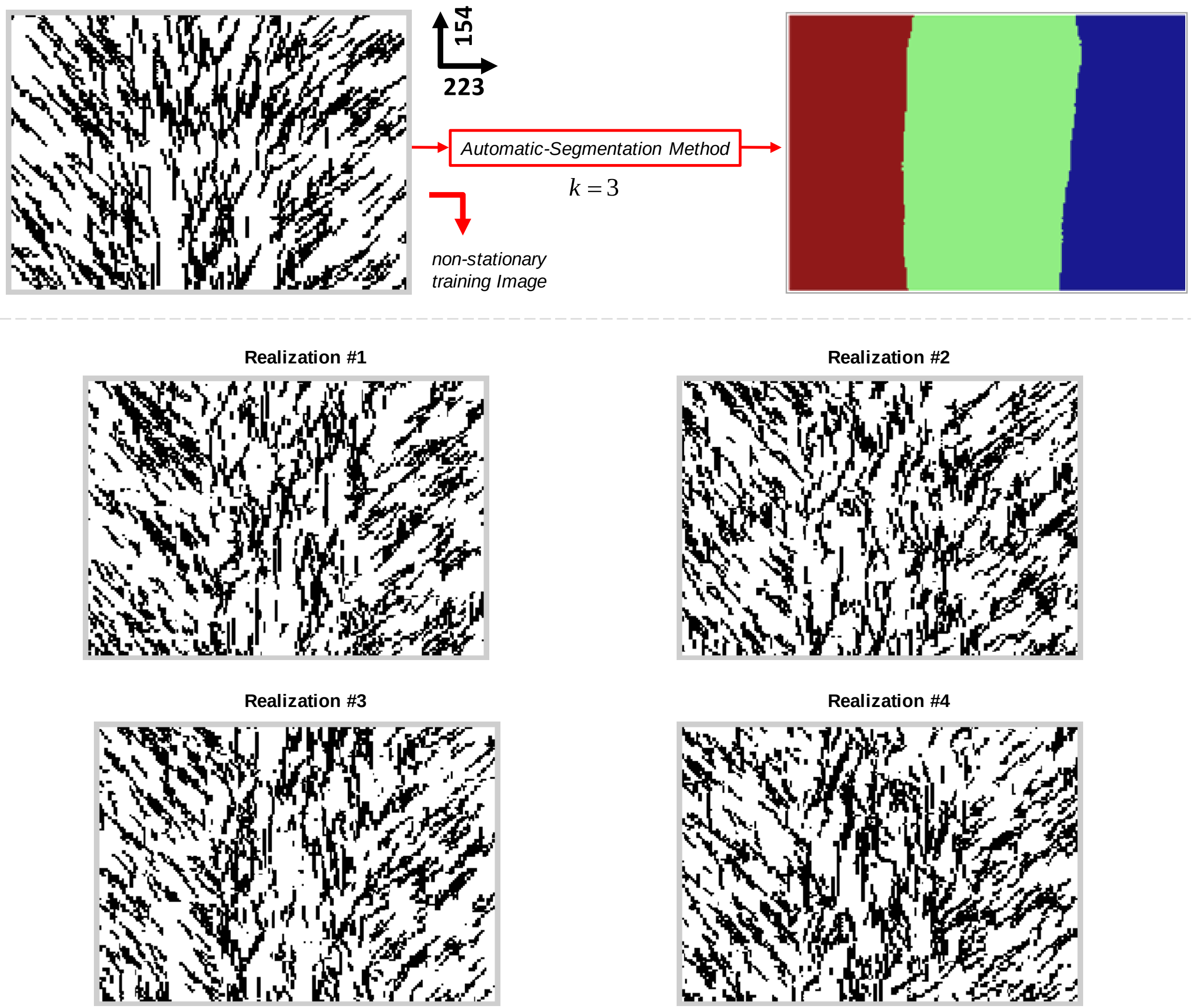


Figure 13: Application of the ASM method for non-stationary modeling of a fractured network.

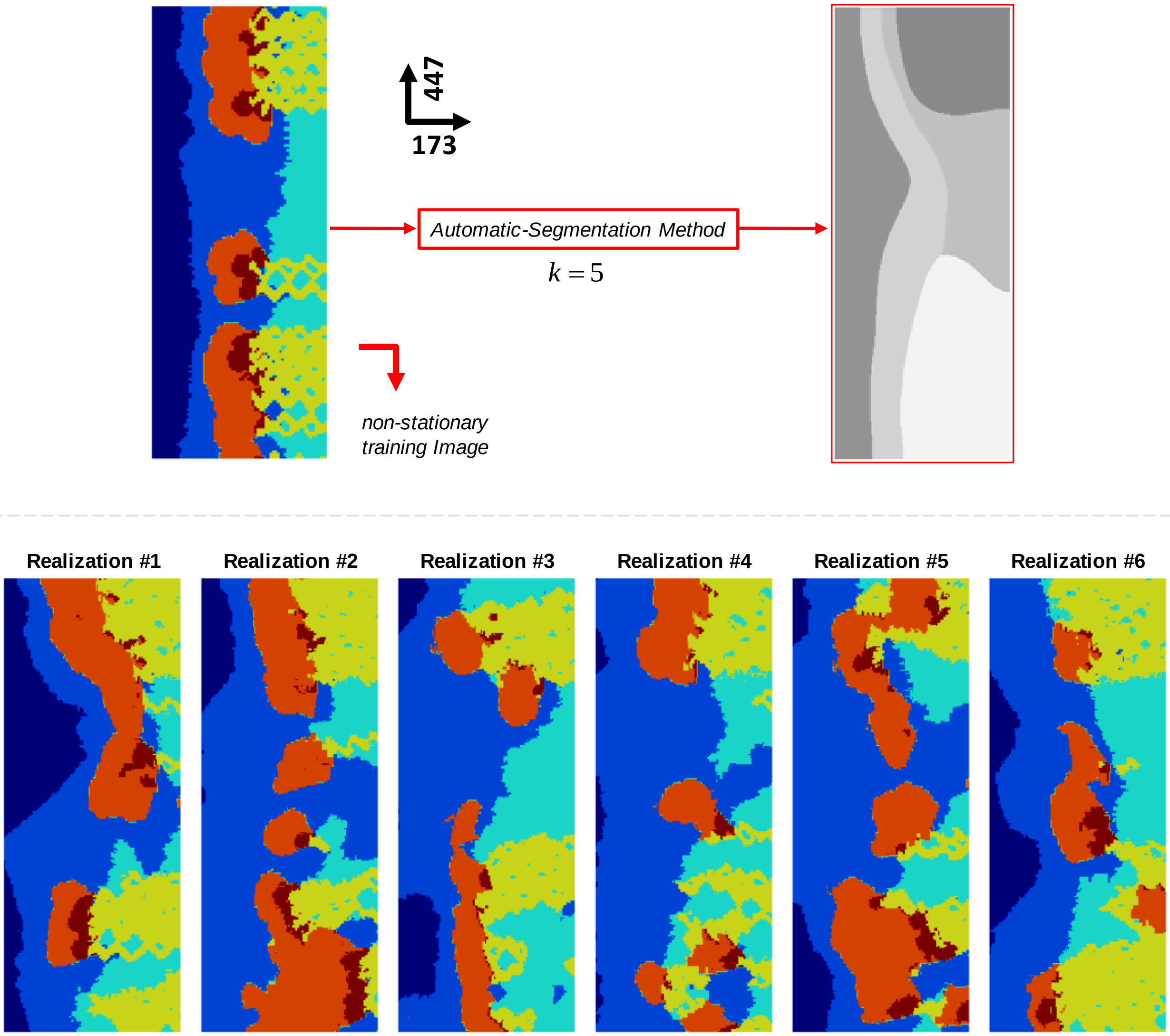


Figure 14: Application of the ASM method for non-stationary modeling of a complex tidal-dominated training image of size $\mathbf{447} \times \mathbf{173}$. The training image is divided into five stationary regions for this simulation.

## 3. Conclusion

In this paper, we addressed the issue of non-stationarity in geostatistical simulation. First of all, we stated that in traditional methods, the introduction of auxiliary variables is seen as instruments in forcing the algorithm to generate the desired characteristics. Second, the stochasticity of the algorithms is affected by the subjectivity of the modeler in choosing the region boundaries, or the probability fields. And finally, the algorithmic adjustments, if possible, that is required for each training image render the current non-stationary modeling a complicated exercise. The simple objective in this paper is to provide a training image to the algorithm and for the algorithm to return realizations, without the need for any additional explicit non-stationary modeling components such as trends, probability maps or auxiliary variables.

By including a geographic component in the pattern they are no longer perceived as location-independent. The Spatial-Similarity Method (SSM) is a general technique that applies to almost any training image. In this algorithm, a parameter is available to the modeler in order to control the non-stationarity modeling assumption. By changing this parameter, the modeler can choose between fully non-stationarity (and hence deterministic) model, or a stationarity model.

An alternative technique, termed Automatic-Segmentation Method (ASM), relies on segmenting the training image is automatically without the need for any auxiliary spatial variable. A non-stationary segmentation variable was incorporated to allow the modeler to select the level of desired non-stationarity for the simulation.

We demonstrated the capabilities of this approach on a variety of non-stationary training images, and showed how this algorithm can be beneficial for each scenario. The results demonstrated the computational and modeling capabilities introduced by the proposed algorithms. The competency of the proposed algorithm not only facilitates the modeling process, but also provides a general framework for simulating any complex spatial model.

## 4. References

Arpat, G. B. and Caers, J, 2007. Stochastic simulation with patterns. Mathematical Geology, v39, no2, 177-203.

Ayinde, O., Yang, Y.-H., 2002. Face recognition approach based on rank correlation of gabor-filtered images. Pattern Recognition 35 (6), 1275-1289.

Boucher, A., 2009. Considering complex training images with search tree partitioning. Computers & Geosciences 35 (6), 1151-1158.

Caers, J and Zhang, T, 2004. Multiple-point geostatistics: a quantitative vehicle for integrating geologic analogs into multiple reservoir models. In: "Integration of outcrop and modern analog data in reservoir models" AAPG memoir 80, p. 383-394.

Chiles , J.P. and Delfiner P., 1999, Geostatistics: Modeling Spatial Uncertainty (Wiley Series in Probability and Statistics).

Chugunova, T., Hu, L., 2008. Multiple-point simulations constrained by continuous auxiliary data. Mathematical Geosciences 40, 133-146.

Daugman, J. G., 1984. Spatial visual channels in the fourier plane. Vision Research 24 (9), 891-910.

Daugman, J. G., 1993. An information-theoretic view of analog representation in striate cortex. MIT Press, Cambridge, MA, USA, pp. 403-423.

de Vries, L., Carrera, J., Falivene, O., Gratac_os, O., Slooten, L., 2009. Application of multiple point geostatistics to non-stationary images. Mathematical Geosciences 41, 29-42.

Dimitrakopoulos, R., Mustapha, H., Gloaguen, E., 2010. High-order statistics of spatial random fields: exploring spatial cumulants for modeling complex non-Gaussian and non-linear phenomena. Math Geosci 42(1):65–99

Dunn, D., Higgins, W., Jul. 1995. Optimal gabor filters for texture segmentation. IEEE Transactions on Image Processing 4 (7), 947-964.

Goovaerts, P., 1997. Geostatistics for Natural Resources Evaluation (Applied Geostatistics Series), Cambridge University Press.

Hamamoto, Y., Uchimura, S., Watanabe, M., Yasuda, T., Mitani, Y., Tomita, S., 1998. A gabor filter-based method for recognizing handwritten numerals. Pattern Recognition 31 (4), 395-400.

Harding, A., Strebelle, S., Levy, M., Thorne, J., Xie, D., Leigh, S., Preece, R., Scamman, R., 2005. Reservoir facies modelling: New advances in mps. In: Leuangthong, O., Deutsch, C. V. (Eds.), Geostatistics Banff 2004. Vol. 14 of Quantitative Geology and Geostatistics. Springer Netherlands, pp. 559-568.

Honarkhah, M. and Caers, J. 2010. Stochastic Simulation of Patterns Using Distance-Based Pattern Modeling, Mathematical Geosciences, 42: 487-517.

Honarkhah, M., 2011, Stochastic simulation of patterns using distance-based pattern modeling PhD dissertation, Stanford University.

Jain, A. K., Farrokhnia, F., 1991. Unsupervised texture segmentation using gabor filters. Pattern Recognition 24 (12), 1167-1186.

Malik, J., Perona, P., 1990. Preattentive texture discrimination with early vision mechanisms. Journal of the Optical Society of America 7, 923-932.

Manjunath, B. S., Shekhar, C., Chellappa, R., 1996. A new approach to image feature detection with applications. Pattern Recognition 29 (4), 627-640.

Marcelja, S., 1980. Mathematical description of the responses of simple cortical cells. Journal of the Optical Society of America 70 (11), 1297-1300.

Gregoire, M., Philippe, R., Julien, S. , 2010. The direct sampling method to perform multiple-point simulations. Water resources Research 46.

Mehrotra, R., Namuduri, K., Ranganathan, N., 1992. Gabor filter-based edge detection. Pattern Recognition 25 (12), 1479-1494.

Strebelle, S., 2002. Conditional simulation of complex geological structures using multiple-point statistics. Mathematical Geology 34 (1), 1-22.

Xu, W, 1996, Conditional curvilinear stochastic simulation using pixel-based algorithms Mathematical Geology, Volume 28, Number 7, 937-949.

Zhang, J., Tan, T., Ma, L., 2002. Invariant texture segmentation via circular gabor filters. In: 16th International Conference on Pattern Recognition. Vol. 2. Quebec City, QC, Canada, pp. 901-904.

Zhang, T., 2002. Multiple-point simulation of multiple reservoir facies. Master's thesis, Stanford University, Stanford, CA.

Zhang, T. 2006. Filter-based training pattern classification for spatial pattern simulation. PhD Dissertation, Stanford University, USA.