Multiple-point Geostatistics: A State of the Art

Andre G. Journel

Stanford Center for Reservoir Forecasting

April 23, 2003

Abstract

The advent of computers in the 1960's made possible the development of geostatistics. Forty years later, large RAM's and massive processing power is about to change dramatically the theory and practice of geostatistics. Analytically defined, concise hence parameter-poor, structural models are not any more a must. Large 3D training images in their full complexity can be considered as random function models which provide the multiple point (mp) statistics needed to characterize and utilize data shapes and patterns, a critical information out of the realm of traditional variograms.

Practice has led the development of mp geostatistics, but we have reached a point where a theoretical formalization can come in. A detailed analysis of SCRF v.10.0 snesim stochastic simulation algorithm allows an understanding of its theoretical foundations, constitutive hypothesis, potentials and limitations, opening avenues for new developments.

1 A short historical review

Initiated by the work of Daniel Krige estimating gold grades in S. African mines (Krige, 1951), geostatistics as a body of theory was built up in the 1960's by Georges Matheron and his fellows in Fontainebleau, France (Matheron, 1962-63, 1965). Geostatistics is now widespread and its field of applications has extended well beyond its original mining roots. As many mature disciplines, it has branched out, taking and sharing concepts and algorithms from so many related disciplines that it would be difficult to attempt an up-to-date definition. In its origin geostatistics was developed towards one goal, that of providing locally accurate grade estimates of mining blocks. The novelty with regard to then well established least-square regression techinques (Williams, 1959; Draper and Smith, 1966) was:

1. The inference from actual data and the modeling of an analytical variogram model, as opposed to the more traditional inference of a covariance matrix which required gridded data. Notions of nested structures, nugget effect/white noise and anisotropy were then introduced.

2. The concept of support effect: the volume support of data were typically much smaller than that of the mining block being estimated. Corrections for difference of support volumes were proposed.

3. Possibly most critical, the full accounting for data redundancy through inversion of a kriging matrix, that is the data-to-data covariance matrix. This was as opposed to the then traditional practice of regression with the data considered as independent variables.

Actually, Halmos (1951), Goldberger (1962) and later Luenberger (1969) did develop a general theory for regression and projection into vector space that encompasses much of Matheron's work on kriging (1969). Matheron and his group of mining engineers had the advantage of wide practice.

The wanting of kriging:

Debate about publication antecedence often masks the real advances implemented by practitioners to fix a wanting algorithm and meet deadlines. How many clean, supposedly new, theories were actually developed from such fixes proven efficient? Kriging, and

for that matter all regression algorithms, had two major drawbacks that quickly became apparent to practitioners:

1. Maps of kriging estimates displayed artefacts (data locations are immediately visible) and, generally, did not reflect the prior image one had from prior experience and visual observation. The smoothing effect of kriging rendered kriging maps biased for any selection that involved jointly several blocks: local accuracy of each block estimate does not suffice. Very early, this author (Journel, 1975) suggested to the sneer of the 1^{st} international geostatistics congress that kriging estimates should never be mapped or used as such.

2. The estimation variance, whose minimization defines kriging and more generally all projection-type estimators, is an incomplete measure of estimation accuracy since it is data values-independent. The crossplots of kriging variance vs. ranks of the cross validation errors typically display insignificant correlation; then how useful was it to minimize such estimation variance? One major exception is provided by the multivariate Gaussian model much touted by theoreticians, but practitioners quickly found that their data (earth sciences) were rarely Gaussian, less multivariate Gaussian distributed.

The advent of simulation:

Stochastic simulation was introduced in the early 1970's to correct for the smoothing effect of kriging and provide maps that displayed the spatial variance predicted from the variogram model (Matheron, 1973; Journel, 1974). The initial turning band algorithm was adapted from an idea suggested much earlier by Matern (1960), a much unsung founding father of geostatistics. The turning band algorithm reduced the problem of generating large 3D (theoretically n D) realizations with a specific 3D isotropic covariance into that, simpler, of generating a series of independent 1D realizations with a related 1D covariance; the latter simulation could be achieved with established spectral techniques.

Because many different, yet equiprobable, realizations could be generated, all conditional to the same set of data, the idea came to using these different realizations as a model of spatial uncertainty (Journel & Huijbregts, 1978). The second major phase in the development of geostatistics, simulation instead of kriging, assessing the impact of spatial uncertainty, had started.

The turning band algorithm was quickly superseded by faster and more flexible algorithms that allowed handling of complex anisotropies and conditioning to a variety of data, hard and soft. Most notable, the sequential algorithm brought back into focus the foundation of any stochastic simulation which is the derivation of conditional probability distributions (cpdf's) from which simulated values can be drawn, e.g. by Monte Carlo sampling. Sequential simulation algorithms are presently at the core of geostatistical theory and practice (Journel, 1983; Isaaks, 1990; Srivastava, 1992; Goovaerts, 1997; Chiles and Delfiner, 1999).

Simulated maps did correct for the smoothing effect and artefacts of kriging, they proved useful in reproducing amorphous, high entropy, structures that could be summarized by a variogram, an example being the spatial distribution of grades or petrophysical properties within an homogeneous mineralization zone or facies. But variogram-based simulation algorithms fail to reflect crisp geometries or specific (as opposed to amorphous) patterns of spatial variability, such as displayed by the distribution in space of facies indicators. Again, a solution was initiated by practioners, this time from the oil industry, more specifically from Norway. Stochastic simulation took its first major turn away from variogram and kriging.

Boolean object-based algorithms were introduced in the late 1980's to simulate random geometry (Stoyan, Kendall and Mecke, 1987; Haldorsen and Damsleth, 1990). Parametric shapes, such as sinusoidal channels or ellipsoidal lenses, are dropped onto the volume to be simulated, then are displaced or removed, their shapes changed, to fit conditioning statistics and local data through an iterative process. At long last, the simulated objects did look as expected from geologist drawings or photographs of present-day depositions. After a period of enthusiasm, the limitations of object-based simulation algorithms became apparent: the iterative, perturbation-type, algorithm for data conditioning did not converge in presence of dense data or could not account for diverse data types (e.g. soft seismic data that are at places locally accurate). The situation of exact hard data conditioning and flexibility in soft data conditioning provided by traditional sequential simulation algorithms was sorely missed.

The wanting of variograms:

The development of geostatistics has been held down by the two basic concepts that made its initial success: kriging as discussed above and the variogram.

The pre-eminence of the variogram/covariance model was never questioned, until recently. Perhaps, the process of inferring the variogram model from actual data gave it a mythical aura of objectivity? Yet, and again, practitioners quickly realized that inference of a variogram, or for that matter any other statistics starting with the histogram, was more an art than an objective science. In most applications, there are rarely enough actual subsurface data to get an interpretable directional variogram, let alone an anisotropic 3D variogram. Whenever the data were abundant, the prior decision of stationarity (i.e. to pool data together) was revisited, leading to pool limitation and consequent renewed data sparsity. It came to the point that variograms are not anymore inferred from actual data, they are chosen from a "mysterious" list of authorized models or, worse, set to the default values of software packages.

At the same time that variograms lost their aura of objectivity, their limitations as measures of spatial variability because evident, again to practitioners. Many different patterns of spatial variability, see Figure 1, may share the same variogram. Thus what controls the output of traditional simulation algorithms is not so much the user's input (the variogram model) but the algorithm retained for simulation, something the user has little control on. Gaussian algorithms and to a lesser extent indicator kriging-based algorithms tend to maximize entropy (i.e. minimize structures) beyond the input variogram: the resulting simulated maps are essentially amorphous. Such modeling decision of maximum entropy is a justifiable starting point in message decoding or in thermodynamics where high entropy is synonym of stable systems, it is a sufficient decision for modelling amorphous distributions of grades or petrophysical properties within an homogeneous mineralization zone or facies. Maximum entropy is **never** a good choice for modelling random geometries particularly those for which we have prior knowledge of shapes and patterns, such prior knowledge being synonym of organization hence low entropy. The uncertainty imparted by organized facies distributions typically overwhelms that of the homogenous, amorphous, distribution of petrophysical properties within those facies.

As to borrow a variogram from an outcrop or a geologist rendering of expected structures, why not borrow much more, borrow statistics that would enable reproducing those structures displayed by the proxy image and that are known to exist in the field being simulated? Not using a training image, because one is uncertain about it, amounts to use the implicit training image of the estimation or simulation software, a training image possibly totally wrong or inappropriate for the types of structures under study. A misplaced desire of objectivity, e.g. accept only data-based variogram, will almost certainly lead to ac-

cept blindly the convenient but arbitrary least-structure option of the simulation algorithm used.

Haldorsen and Damsleth (1990) and the proponents of object-based modelling were forerunners, dropping the variogram and kriging all together, instead borrowing whole shapes from prior information. They were successful inasmuch as that prior information was valuable information that would have been ignored by variograms. Unfortunately, in their boldness they also drop the pixel-based approach whereby the simulated field, its shapes and patterns, are constructed one point or one grid node at-a-time, a feature particularly convenient for conditioning to data of various volume supports and resolutions: it suffices to freeze the hard data at their locations then build around them.

Borrowing directly conditional probabilities:

The variogram function serves as a distance model between any 2 points (but only 2 points); that distance is used to infer the probability distribution of any unsampled value weighting the influence of each datum, taken one at-a-time, by its distance to the unsampled location: that weighting process is none other than kriging. Variogram and kriging are but tools to get the critical ingredient to any simulation and to any uncertainty assessment: the probability distribution of the unsampled value given its specific conditioning data event. Hence the idea of borrowing directly that probability distribution from experimental replicates of the conditioning data event.

Consider available a training image that would display jointly data patterns and patterns of the attribute being simulated, that training image would be scanned for replicates of the experimental conditional data event. Consider that L such replicates are found, the histogram of the L corresponding central attribute values can be taken as a model for the probability distribution of the unsampled value given that data event, see Figure 2. What is borrowed from the training image are not 2-point statistics, variograms or 2-point correlations, from which to stitch back haphazardly a conditional distribution, that is a $(n+1)$-point statistics if the conditioning data event comprises n data locations. What is borrowed is directly that $(n+1)$-point or multiple-point statistics, that is the required conditional distribution, Srivastava (1992), Caers (1998), Strebelle (2000). All structural statistics, including all variograms, come from the training image, hence are controlled by the user through his choice of that training image; none are coming from the simulation algorithm which is outside the user's control. The training image plays the role of a non-analytical, yet fully explicit, random function model which provides a number (ideally

sufficiently large) of multiple-point (mp) statistics. Of course, the proviso is availability of such rich training image, see later discussion in section 3.

Note that the conditioning data event could be absolutely anything, n can be large, the n data need not relate to the attribute being simulated, the n data configuration can be anything, etc $\cdots$. Of course, the training image should be equally diverse, be multiple attributes, displaying enough replicates of the varied data events.

2 The concept of random variable

The modeling of uncertainty which is at the roots of probability theory, geostatistics and stochastic simulation, relies on the concept of a random variable. There is no need, actually it is unhelpful, to dwell into axiomatic definitions (Yaglom, 1962); it suffices to retain the definition which made the concept of random variable useful for practical applications.

A random variable (RV), denoted by the capital letter Z, can be seen as a collection of outcomes $\{z^{(l)}, l = 1, \cdots, L, \text{ or } l \in (L)\}$ which the variable Z can take; each outcome or class thereof is attached with a probability $p^{(l)} \in [0, 1]$, then $\sum_{(l)} p^{(l)} = 1$. The probability distribution $\{p^{(l)},\ z^{(l)},\ l \in (L)\}$ fully characterizes the RV Z and provides a model for the uncertainty about the actual outcome, typically unknown, of Z. The building of that uncertainty model thus requires: (1) a census of all possible outcomes $\{L\}$ or classes thereof and, (2) attaching a probability to each such outcome.

Remarks:

- The most critical point about modeling uncertainty is that there is **no** unique, best or true model. There can be several alternative probability distributions for the RV Z, depending on various subjective decisions about what should be considered as relevant information. Playing with these decisions, uncertainty can be modeled as small or as large as one wishes, and it is certainly naive to believe that there could be an "objective" assessment of uncertainty.

- The random variable concept can be extended to cover the joint distribution of several variables not all necessarily related to the same attribute or the same location in space. Consider K such RV's Z_k, $k = 1, \cdots, K$ and the random vector $\mathbf{Z} = \{Z_k,\ k = 1, \cdots, K\}$. The probability distribution characterizing that random

vector $\mathbf{Z}$ is considerably more complex than for the previous case $K = 1$. It requires (1) a census of all joint outcomes $\{z_k^{(l_k)},\ l_k \in (L_k)\},\ k = 1, \cdots, K$ and, (2) the derivation of the corresponding joint probabilities such as:

$$Prob\{Z_k = z_k^{(l_k)},\ k = 1, \cdots, K\},\ \forall\ l_k \in (L_k),\ k = 1, \cdots, K. \tag{1}$$

Note that knowledge of all probabilities of type (1) allows calculating any conditional probability, such as:

$$\begin{aligned} &Prob\{Z_{k_0} = z_0^{(l_0)} \mid Z_{k'} = z_{k'}^{(l_{k'})},\ k' \in (K') \subset (K)\} \\ &= \frac{Prob\{Z_{k_0} = z_0^{(l_0)},\ Z_{k'} = z_{k'}^{(l_{k'})},\ k' \in (K') \subset (K)\}}{Prob\{Z_{k'} = z_{k'}^{(l_{k'})},\ k' \in (K') \subset (K)\}} \end{aligned} \tag{2}$$

where, e.g., Z_{k_0} would model the variable whose uncertainty is to be assessed, $Z_{k'}$ are the RV's data providing information related to Z_{k_0}, the K' corresponding data values are the $z_{k'}^{(l_{k'})}$, $k' \in (K')$. The decision to retain K' specific variables as data among the K possible variables is not necessarily unique, nor is it undisputable; so is the decision about what the initial pool of K variables should be. Then, there is also the issue of inference or modeling of the set of joint probabilities (1). As we will see that inference process is far from being objective.

Random function:

In the case where all K variables defining the random vector $\mathbf{Z} = \{Z_k, k = 1, \cdots, K\}$ relate to the same attribute but at different locations in space, the term random function (RF) is used and the notation is:

$$Z(\mathbf{u})\ :\ \{Z(\mathbf{u}), \mathbf{u} \in \text{domain } D\} \tag{3}$$

where $Z(\mathbf{u})$ denotes a single RV at a location of coordinates vector $\mathbf{u}$.

This author regrets the terminology "function", implying some functional, analytical, relation of the RV Z with its location in space; there is usually none. The major advantage of the RF concept is its concise notation: the RF $Z(\mathbf{u})$ is fully characterized by its spatial law defined as the set of all joint probabilities of type (1) rewritten as:

$$Prob\{Z(\mathbf{u}_1) = z_1,\ Z(\mathbf{u}_2) = z_2,\ \cdots,\ Z(\mathbf{u}_K) = z_K,\ \} \tag{4}$$

$$\forall\ \mathbf{u}_k \in \text{domain } D,\ \forall\ z_k,\ \forall\ K$$

Again knowledge of the spatial law (4) allows calculating the conditional probability of any single $RV\ Z(\mathbf{u}_0)$ given any number K' of data related to neighboring RV's $Z(\mathbf{u}_{k'})$.

In practice the RF concept may have to be extended to a vectorial RF involving several distinct yet (cor)related attributes, each distributed in space:

$$\mathbf{Z}(\mathbf{u}) = \{Z_k(u),\ k = 1, \cdots, K;\ \mathbf{u} \in \text{Domain } D\}$$

Think about $Z_1(\mathbf{u})$ being porosity; data can be either porosity data $z_1(\mathbf{u}_\alpha)$ and/or related sonic log values $z_2(\mathbf{u}_\beta)$.

Representation of the spatial law:

The random vector or random function concept is useful inasmuch as its spatial law (1) or (4) can be determined yielding the required conditional probabilities (2).

In a pre-digital computer era, there was no alternative but to restrict the extraordinary information wealth of the spatial law to a parametric analytical model with as few parameters as possible, for these parameters had to be evaluated. Many decisions can be made to restrict the wealth of the spatial law (1) and thus make its inference easier. Unfortunately, these decisions all come at the cost of loss of information related to the variable Z_{k_0} whose uncertainty is being assessed:

- reduce the number of related variables, the $Z_{k'}$'s in expression (2)
- assume stationarity, i.e. invariance of the spatial law by translation in space
- limit the complex joint dependence of any set of K variables Z_k, $k = 1, \cdots, K$ to the $K \times K$ correlation matrix of any pair of variables
- assume an analytical spatial law that would depend only on such correlation matrix or covariance function, typically a Gaussian-related random function or vector model. Gaussian models are indeed congenial and parameter-poor, but their consequence in terms of estimation and under-assessment of uncertainty were not always well understood (Anderson, 1958; Journel, 1996).

Many of the previous limiting decisions, in particular the two latter which limit statistics to the covariance, need not be any more now that large digital memory and processing power allows a non-analytical representation of the spatial law. Instead of characterizing

the RF $Z(\mathbf{u})$ by its spatial law (4), one could characterize it by a (very) large number L of its realizations over the domain D:

$$\{z^{(l)}(\mathbf{u}), \mathbf{u} \in \text{Domain } D\}, \quad l = 1, \cdots, L \tag{5}$$

Anyone of the joint probabilities defining the spatial law (4) can be identified to the corresponding proportion found within the L realizations, e.g. for $K = 3$:

$$\begin{aligned} &Prob\{Z(\mathbf{u}_1) = z_1,\ Z(\mathbf{u}_2) = z_2,\ Z(\mathbf{u}_3) = z_3\} \\ &\quad = \text{Proportion of realizations among the } L \text{ available such that:} \\ &\qquad z^{(l)}(\mathbf{u}_1) = z_1,\ z^{(l)}(\mathbf{u}_2) = z_2,\ z^{(l)}(\mathbf{u}_3) = z_3 \ \text{ simultaneously.} \end{aligned} \tag{6}$$

Similarly, for the conditional probability of having $Z(\mathbf{u}_1) = z_1$ given the data event $\{Z(\mathbf{u}_2)\} = z_2,\ Z(\mathbf{u}_3) = z_3\}$:

$$\begin{aligned} &Prob\{Z(\mathbf{u}_1) = z_1 \,|\, Z(\mathbf{u}_2) = z_2,\ Z(\mathbf{u}_3) = z_3\} \\ &\quad = \frac{\text{\# of realizations s.t. } z^{(l)}(\mathbf{u}_1) = z_1,\ z^{(l)}(\mathbf{u}_2) = z_2,\ z^{(l)}(\mathbf{u}_3) = z_3}{\text{\# of realizations s.t. } z^{(l)}(\mathbf{u}_2) = z_2,\ z^{(l)}(\mathbf{u}_3) = z_3} \end{aligned} \tag{7}$$

The theoretical concept of a RF defined by its spatial law (4) is replaced by an explicit set of realizations. Defining a RF amounts to pooling together the L realizations (5); adopting the RF model (3) to assess uncertainty about a particular unsampled value $z(\mathbf{u}_1)$ amounts to decide that this unsampled value and its related data event $\{z(\mathbf{u}_2) = z_2,\ z(\mathbf{u}_3) = z_3\}$ belongs to the specific family of realizations (5).

Remarks:

- Inasmuch as each of the L defining realizations $z^{(l)}(\mathbf{u}),\ \mathbf{u} \in D$, exists and is considered a relevant member of the pool, **all** statistics of type (6) or (7) are perfectly licit and consistent with each other. In particular, were a 2-point covariance matrix inferred from these L realizations, that covariance matrix is necessarily positive definite without any need for any correction (Journel, 1996).

- The critical problem is how large should be the number L of realizations to ensure that all unknown-to-data event of the type $z(\mathbf{u}_1) = z_1,\ z(\mathbf{u}_2) = z_2,\ z(\mathbf{u}_3) = z_3$ are represented in the pool with enough replicates to allow "reliable" derivation of conditional proportions of the type (7).

Note that if a particular data event is not found in the pool (5) or is found with too few replicates, this only means that the pool is not "rich" enough to permit calibration of that particular data event. The solution is then to either define a larger pool, or to reduce or approximate the data event to allow finding enough replicates of it in the initial pool; the latter solution is discussed further in the next section 3.

- A mere combinatorial calculation for the number L of realizations necessary would be incorrectly scary. Indeed if the data search neighborhood consists of 1000 possible locations: $K = 1000$, and each variable $Z(\mathbf{u}_k)$ is discretized into 10 classes, the total number of different possible realizations is formidable: 10^{1000}, then if one wishes to have an average of 10 replicates for each realization, then $L = 10^{1001}$! However, (1) the vast majority of these 10^{1000} realizations will never be encountered in practice, hence there is no need for the pool (5) to include them; (2) a particular data event not found in the pool can be either reduced (Strebelle, 2002) or it can be related to different but similar data events present in the pool (Arpat, 2003). A similarity or distance measure need to be defined for the latter solution.

- **Local stationarity**: Instead of a large number L of realizations of type (5) over a domain D, one could equally consider one single realization $L = 1$ over a much larger domain DD, say L times the size of D:

$$\{z^{(1)}(\mathbf{u}),\ \mathbf{u} \in \text{domain } DD\}, \quad \text{with } |DD| >> |D|. \tag{8}$$

  That larger domain DD should not be scanned for any data event extending over dimensions larger than D. Pooling replicates over this larger domain DD amounts to a decision of local stationarity (invariance by translation) of all statistics of type (6) or (7) defined over the smaller extent D.

- **Sequential approach**: The explicit representation of a spatial law by a set of realizations of type (5) does also provide probabilities of multiple-point events conditional to multiple-point data events. For example:

$$\begin{aligned} &Prob\{Z(\mathbf{u}_1) = z_1,\ Z(\mathbf{u}_2) = z_2,\ |\ Z(\mathbf{u}'_\alpha) = z'_\alpha,\ \alpha = 1, \cdots, n\} \\ &\quad = \frac{\text{\# of realiz. s.t. } z^{(l)}(\mathbf{u}_1) = z_1,\ z^{(l)}(\mathbf{u}_2) = z_2,\ z^{(l)}(\mathbf{u}'_\alpha) = z'_\alpha,\ \alpha = 1, \cdots, n}{\text{\# of realiz. s.t. } z^{(l)}(\mathbf{u}'_\alpha) = z(\mathbf{u}'_\alpha),\ \alpha = 1, \cdots, n} \end{aligned} \tag{9}$$

  However, it may be more efficient to proceed sequentially, building on the exact probability decomposition:

$$Prob\{Z(\mathbf{u}_1) = z_1,\, Z(\mathbf{u}_2) = z_2,\, |\,(n)\} \tag{10}$$
$$= Prob\{Z(\mathbf{u}_1) = z_1,\, |\, Z(\mathbf{u}_2) = z_2,\, (n)\} \times Prob\{Z(\mathbf{u}_2) = z_2,\, |\,(n)\}$$

where $|(n)$ is short notation for conditioning by the n-data event

$$Z(\mathbf{u}'_\alpha) = z(\mathbf{u}'_\alpha),\ \alpha = 1, \cdots, n$$

If approximations are accepted for the identification of conditional probabilities to proportions as in relation (7), then evaluating separately the two single-point probabilities of the product expression (10) provides more flexibility.

The objectivity issue:

Which of a concise analytical representation of the spatial law or an explicit numerical representation through training images of the types (5) or (8) is better?

A few years ago, the answer was clear: the analytical representation, because we did not have the computers to generate, store and retrieve the large amount of data involved in a training image. However, those analytical representations could not reflect prior information about complex shapes and geometrical patterns, hence the resulting maps were poorly conditioned in that this prior information, however critical, was forfeited. Object-based simulation algorithms fall into the category of training image-based algorithms, in that after conditioning to local data there is not anymore any analytical representation of the corresponding spatial law.

There is unfortunately an aura of objectivity to analytical representations, perhaps due to that ever lingering complex of geologists towards mathematicians and statisticians, their belief that an equation yields objectivity. An equation models concisely prior information, it is no better than the prior information it captures, then, precisely because of that conciseness, equations and analytical representations can only capture very simple, elementary, possibly simplistic, information. The real question is: " what is better, a numerical model (e.g. a training image) that captures more of the prior information deemed critical, or a concise "clean" analytical model? " What does conciseness brings us in an era of ever increasing computing power?

Training images force the geologist/naturalist to express his prior information. If that prior information is uncertain, then he should provide alternative training images bracketing the range of uncertainty, rather than give up and let a "clean" and arbitrary analytical

model imposes a prior information that, if expressed as a training image, he would likely reject as wrong. Now, if that analytical model does yield non data-conditioned realizations/maps that are acceptable, then for sure the concise model should be preferred.

Objectivity does not mean subservience to default models. Better several imperfect training images than a wrong and biased default model. Remember that maximum entropy or minimum structuration as in Gaussian-related models most often yield too optimistic results; uncertainty and risk arise from the presence of structures or patterns that we know exist but could not locate accurately in space (Journel, 1996).

3 Training image considerations

In this section, we will consider the training images as a numerical representation of the spatial law of the random function model. Denote such training image by:

$$\{z^{Ti}(\mathbf{u}),\ \mathbf{u} \in Ti\} \tag{11}$$

where Ti stands for training image, $z^{Ti}(\mathbf{u})$ is the training value at location $\mathbf{u}$.

Training statistics are limited to data configurations that fit into a template T of size much smaller than the training image size, i.e.:

$$\text{Template } |T| << |Ti|$$

The training image can be seen as the set of all possible template data that can be extracted from it, if they are L such templates (overlapping each other) then the Ti representation (11) is actually equivalent to the representation (5) distinguishing all L training templates. This remarks indicates that, instead of one very large Ti, one could consider many Ti's each of lesser size.

Both the training image and the template are discretized by the same rectangular grid to allow scanning the Ti by sliding over it the template T, see Figure 3.

Inference: Consider an actual reservoir discretized with the same grid used by the Ti and the template T, then an unsampled value at location $\mathbf{u}$ denoted $z(\mathbf{u})$. The attribute z being estimated is of the same type as the Ti attribute z^{Ti}. Notice the notation difference between z-value (actual reservoir) and z^{Ti}-values (training image).

The unsampled value $z(\mathbf{u})$ is informed by a multiple-point (mp) "data event" (DEV) constituted by n data values $z(\mathbf{u}+\mathbf{h}_\alpha),\ \alpha=1,\cdots,n$, at n locations $\mathbf{u}_\alpha=\mathbf{u}+\mathbf{h}_\alpha$ all within the template T centred at $\mathbf{u}$. The DEV $\{z(\mathbf{u}_\alpha),\ \alpha=1,\cdots,n\}$ need not fill-in all locations of the template T, i.e. : $n\leq|T|$.

Uncertainty about the unsampled value is modeled by the histogram of the R central values of training templates which replicate the DEV $\{z(\mathbf{u}+\mathbf{h}_\alpha),\ \alpha=1,\cdots,n\}$. Denote these R central values by: $z^{Ti}(\mathbf{u}^{(r)}),\ r=1,\cdots,R$; they are such that in their T-neighborhood:

$$\{z^{Ti}(\mathbf{u}^{(r)}+\mathbf{h}_\alpha)=z(\mathbf{u}_\alpha),\ \alpha=1,\cdots,n\},\ \ \forall\, r \tag{12}$$

The histogram of the R central values is taken as the conditional probability distribution of the random variable $Z(\mathbf{u})$, see relation (7).

The conditioning data event comes from the actual field, the statistics and the probability values comes from the RF model, e.g. from its training image representation. It is critical to understand this dichotomy: local data must be taken from the actual reservoir, information about how these data relate to the unknown comes from the training image model. Similarly, in traditional geostatistics, how local data relate to the unknown is given by some kriging process built on the variogram model. The huge difference is that a Ti provides mp statistics that allow considering the DEV as a whole, as opposed to that DEV being considered piecewise, one datum at a time, because of the 2-point statistics limitation of the variogram.

Number of replicates: An integral part of the random function model, as delivered by a training image, is the minimum number R_{min} of replicates of the DEV to be found in that Ti before the corresponding histogram of the training central values can be taken as a model of the conditional probability, recall definition (12).

- If R_{min} is too small, those few replicates may be too specific to a few localities of the Ti and deemed not representative of the Ti as a whole.

- If the minimum number R_{min} is set too large, the Ti (or sets of Ti's) may not be large enough to provide enough replicates, in which case approximations have to be made which all amount to tamper with the conditioning DEV.

The case of no replicate ($R=0$) is instructive. This case can have two very different causes:

1. the Ti is not rich or large enough to display that DEV. The solution is then to accept replicates that only approximate instead of reproducing exactly the DEV, or to reduce the DEV, e.g. by dropping out the least "relevant" datum value.

2. the Ti adopted is not the correct one. This would be the case if too many experimental DEV's had no replicates in the Ti, even after reduction of the DEV size. In other words, the actual reservoir data contradict the Ti retained. The solution is then to change the Ti.

One main reason for too few DEV replicates is the requirement for exact replicates. Allowing a tolerance for "approximate" replicates would multiply their number, see hereafter the section on classification of DEV's.

Template size and discretization:

No statistics beyond the template size is taken from the training image, and those statistics must be based on a minimum number of replicates. Template size, minimum number of replicates and the training image define the RF underlying the multiple-point geostatistics approach.

The template geometry and size is the equivalent of the data search neighborhood of traditional geostatistics and, for that matter, of any spatial interpolation technique. You do not wish to retain data that are too far away because: (1) you are not sure they belong to the same population than the unsampled value to be estimated, this restriction is related to the decision of stationarity defining the data pool, (2) you are uncertain about the structural model relating such far away data to the unsampled value, there may not be enough training replicates of the corresponding large size data event.

Yet far away data provide information about large scale spatial structures. Retaining a too small template size amounts to forfeiting reproduction of those structures. If such large scale structures are deemed to exist and must be reproduced, then a corresponding large scale Ti must be available and would have to be scanned with a large template size. A large template size need not contain too many nodes, if it is discretized with a coarse grid. Recall that the more nodes a data event is comprised of, the more specific it is, hence the fewer replicates of it will be found in the Ti. The traditional solution of multiple grids simulation (Tran, 1994) can be adopted, see Figure 3:

- the final high resolution fine grid over which the study field is to be simulated is split into a series of nested grids from coarse to finer. If 3 nested 3D grids are

considered, the coarsest one includes only every $8^2 = 64^{th}$ node of the finest grid, the second coarsest includes every 8^{th} node of that finest grid.

- simulation proceeds first on the coarsest grid with an equally coarse template which thus can be of large size. A separate very large Ti but coarsely gridded could be considered to display the large scale structure to be reproduced.

- in a sequential simulation mode, every node simulated on a coarse grid is considered as a conditioning datum for simulation of all other grid nodes. The Ti'(s) used for the finer grids need not be the same as the coarse large Ti. These fine scale Ti's are scanned by correspondingly finely gridded templates but of smaller extent. The fine scale Ti's need not display the large scale structures, they should however display the fine scale structures to be reproduced.

Data relocation:

Because of the need to scan the Ti to find replicates of the conditioning DEV, both the data and the field to be simulated should be gridded with a rectangular grid. If multiple nested grids are used, then at any one of these grids the original sample data may have to be relocated to its nearest grid node. This relocation not only causes data location inaccuracy particularly at coarse grids, but also data location inconsistency if the same datum is relocated at different locations when used over different nested grids.

Ideally, hard sample data should not be relocated (no tampering) yet the scanning process requires a rectangular grid. One solution consists of replacing the original data with estimated, or better simulated, values at the data neighbouring nodes, see proposal in section 5.

Modeling the training image:

No matter how large the Ti, when scanned with a template T of fair size it will never yield all the DEV's that will be encountered in the course of simulating the field under study. As mentioned before, there are two courses of action: (1) reduce the data event which amounts to some loss of conditioning information, (2) interpolate that DEV from "similar" DEV's actually found in the training image. In the first option, the only statistics taken come from the Ti as is, there is no adding to it; if necessary, data are dropped: this extreme fidelity to the Ti is debatable in that one is never sure that the Ti is fully

representative of the experimental data. This first option is that adopted by the search tree approach of Strebelle (2000, 2002).

The 2^{nd} option calls for modeling the Ti, this can be done in many ways:

- **Classification**: For a given template size T, all training DEV's are retrieved and classified (unsupervised classification) into K classes. The histogram of the training central values of all DEV's falling into a class is taken as the conditional probability given that class of DEV's. During the simulation process, any experimental DEV, whether its template is fully informed or not, is attributed to one of the previous training classes, and that class conditional probability is retrieved. There are many alternative ways to do the prior classification, the avenues that are presently being investigated at SCRF are:

  1. defines a distance between any two training DEV's, then group these DEV's into a determined number of classes. This is the approach of Arpat (2003, in this SCRF report). A similar approach consists of reducing the dimension of each training DEV by retaining only the first few principal components (PC's) of its template data. Some form of cluster analysis then allows to define classes of DEV's based on their PC's. This is the approach used by Caers, Strebelle and Parayzan (2003), Zhang (2003), Liu, Harding and Abriel (2003), the two latter references are in this SCRF report.

  2. the classification tree approach (Breiman et al., 1984) consists of partitioning the KT-dimensional space of all training DEV's into a much smaller number of regions or classes of DEV's. T is the number of nodes of the data template, K is the number of classes discretizing the range of variability of each nodal value. The partitioning of the initial KT-dimensional space is done by a sequence of binary splits, each split ensuring maximum "homogeneity" of the two resulting classes. This requires defining a measure of homogeneity of any class of training DEV's. This approach is being investigated by Remy (2002).

  Defining a distance or a measure of homogeneity between DEV's brings another input decision into the definition of the random function implicit to the training image; associating a specific experimental DEV to a class or region of similar DEV's amounts to tampering with those experimental data. However, the classification approach offers many potential critical advantages: (1) dimension reduction and

limited memory demand associated to the fewer number of classes retained, (2) filtering of training DEV's through their pooling into class, (3) expanding the Ti model to accept any experimental DEV even if that exact DEV is never found exactly in the Ti.

- **Neural net models**: The set of training conditional proportions could be modeled by explicit analytical probability distribution functions such as:

$$\begin{aligned} &Prob\{Z(\mathbf{u}) \in \text{class } z \,|\, Z(\mathbf{u}+\mathbf{h}_\alpha) = z_\alpha,\ \alpha = 1, \cdots, T\} \\ &\quad = \varphi_\theta(z \,|\, z_\alpha,\ \alpha = 1, \cdots, T) \end{aligned} \tag{13}$$

where $\mathbf{u}$ is the central location of a template fully informed at all its T locations $\mathbf{u}+\mathbf{h}_\alpha$, $\alpha = 1, \cdots, T$. The z_α's are the data values. The function $\varphi_\theta(.)$ gives the probability that the central value $Z(\mathbf{u})$ be in the specific class z. The parameters θ of that probability distribution function are fitted by a neural net from the corresponding observed training proportions. The training phase consists of reducing the Ti to the parametric function $\varphi_\theta(.)$, the simulation phase consists of simulating the unsampled value $Z(\mathbf{u})$ by drawing from that function (Caers, 1998; Caers and Journel, 1999).

The neural net approach carries all the advantages of the classification approach, but can only accommodate "full" DEV's with all their template nodes informed. Thus simulation must start with a fully informed field which is gradually updated one node at-a-time until some criterion of convergence is met. For example, the field can be initiated with nodal values all drawn from the target marginal distribution of the Z-attribute; such initial field would thus present no spatial structure and would be sequentially modified by successive drawings from conditional distributions of type (13) or by an acceptance/rejection Metropolis-type algorithm (Hastings, 1970; Caers, 1998). As with all Markov chain-type algorithms, convergence is an issue: which criterion should be used to stop the iteration, convergence may be slow, and when the iteration is stopped is it unclear what statistics or structure has been actually retained from the original Ti.

The search tree concept:

As opposed to classification, the search tree takes the training image as is, does not add anything to it, does not tamper with it. The training proportions are merely recorded in a

dynamic data structure, called a search tree (Roberts, 1998). Search trees are particularly convenient to store nested information such as numbers of outcomes of DEV's, where the larger DEV includes lesser ones.

If the idea of borrowing directly conditional probabilities from a training image is now more than a decade old (Srivastava, 1993), the breakthrough that allowed wide applications was utilization of a search tree as proposed by Strebelle (2000). Instead of scanning the Ti anew at each node being simulated, that Ti is scanned only once prior to any simulation, and all the training DEV occurences within a given template are recorded in the search tree.

There is one search tree for each data template size and geometry, hence if 3 different multiple grids are used in the sequential simulation process with correspondingly 3 different templates, there will be 3 different search trees even if the same finely gridded Ti is used.

The grid nodes constituting a template are first ordered, typically with the nodes closest to the central template location given the highest ranks. The root of the search tree gives the training histogram (proportions); discounting border effects this is the marginal histogram of the $z^{Ti}(\mathbf{u})$ values, $\mathbf{u} \in Ti$. If that histogram is discretized into K classes, that search tree root branches out into K daughters corresponding to conditioning by the highest ranked template node value, say $z^{Ti}(\mathbf{u}_1) = z_k$ for the k^{th} class and branch. Each of the K 2^{nd} level nodes of the search tree gives the histogram of the 2^{nd} ranked template nodal value conditional to a particular class value of the 1^{st} ranked nodal value:

$$\text{Proportion}\{z^{Ti}(\mathbf{u}_2) = z_{k'} \,|\, z^{Ti}(\mathbf{u}_1) = z_k\},\ k' = 1, \cdots, K$$

where $\mathbf{u}_1$ is the 1^{st} template node, and $\mathbf{u}_2$ is the second; z_k, $z_{k'}$ are classes discretizing the range of the z-attribute value.

Each of the 2^{nd} level nodes of the search tree branches out into K daughters leading to 3^{rd} level nodes, $\cdots$ and so on until the last ranked template node. For further details, the reader is referred to the remarkable thesis work of Strebelle (2000), or Strebelle (2002).

Actually what is recorded at each node of the search tree is not the proportion but actually the number of training occurences, e.g.:

$$\#\text{ of occurences of } \{z^{Ti}(\mathbf{u}_2) = z_{k'} \,|\, z^{Ti}(\mathbf{u}_1) = z_k\}$$

from which one can check whether the number of DEV replicates is sufficient and retrieve the required conditional proportions.

The size, hence RAM demand, of a search tree is not related to the Ti size, it is related to the number T of nodes of the data template and the discretization level K of the range of the z-attribute value. The size of the search tree is proportional to the number of different DEV's of any size $\leq T$ found in the Ti; that number is typically much lesser than the total of nodes in the Ti, hence much lesser than K^T because not all possible DEV's are found, or need to be found, in the Ti. In typical applications $K \leq 3$, i.e. mp geostatistics is limited to categorical variables. $T \leq (7 \times 7 - 1) = 48$ in 2D, $(5 \times 5 \times 3 - 1) = 74$ nodes in 3D corresponding to a small template discretization; recall however that the template size (extent) can be large if the gridding of that template is coarse.

The advantage and also limitation of the search tree is that it records the Ti DEV's exactly as they are found. Reading from the search tree, an experimental DEV is either exactly matched by a minimum number of Ti replicates, or that DEV needs to be reduced e.g. by dropping out the lowest ranked datum value. The search tree per se does not allow any tolerance for "approximate" replicates; it does not allow for either filtering or interpolation of DEV's. Overcoming this limitation has been an impetus for research, see the previous section on modeling the training image.

Where to get training images:

When interpolating or simulating between data one utilizes two sources of information: (1) the local data on which to anchor (2) the structural model, be it a surface, a covariance function (dual kriging) or a portion of a training image. We insist that the structural model that links the data together should be actual information, not some arbitrary surface. The geologist, more generally the physicist of the data, should have a prior idea of what the final field should look like; that prior information is critical and should not be forfeited for an arbitrary, typically congenial oversmooth and wrong structural model hidden behind the automatic contouring algorithm. In the common case of local data sparsity, the final estimated or simulated field depends more on the structural model used than on the few data on which it is anchored, hence it is wrong not to attempt building a structural model that would match that prior information. A training image provides a vehicle for expressing and utilizing the prior structural information. If the geologist hesitates, then he should be pressed to provide several alternative guesses under the visually explicit form of alternative training images. In most situations, that uncertain geologist would reject categorically an amorphous Ti which is implicit to most Gaussian, variogram-based, structural models. Unfortunately, the structural or random

function model underlying analytical spatial laws are never made visually explicit which prevents them to be outright rejected if deemed inappropriate.

The main contribution of a Ti is to be a vehicle for expressing prior structural information. A Ti is visually explicit, hence is more attuned to the judgement of a geologist. Geologists do no think in terms of histograms or variograms, however they can draw since most of their experience is recorded as figurative patterns.

Because it is dissociated from the field sample data, a training image need not reproduce these data, i.e. it need not be locally accurate. A Ti is a pure conceptural rendering of what shapes and spatial structures should be. Interpreted photographs of outcrop and present-day depositions, hand-drawn sketches of patterns can serve as starting points to build the large 3D Ti's needed for mp geostatistics. They are only starting points, because sketches and photographs are 2D and limited in their extent and resolution. These sketches and more generally the "vision" of the geologist should guide computer runs to generate the full size 3D training images needed for mp geostatistics. Object-based or process-based algorithms and computer-aided design and modeling of 3D volumes are all valid approaches to generating full size 3D Ti's; because these algorithms are freed from the conditioning to local data, they can be fine-tuned to deliver Ti's that match the prior conceptual "vision" of geologists; see Haldorsen & Damsleth (1993), Bratvold et al. (1994), Deutsch (2002) for object-based algorithm; Tetzlaff, D. and Harbaugh, J. (1989), Kolterman and Gorelick (1996), Wen et al. (1998), Srivastava (2002) for process-based algorithms; Mallet (2002) and the gOcad software (Earth Decision Sciences, 2002) for computer-aided modeling of 3D volumes.

Catalog of training images:

There exist many excellent geological books providing a classification of depositional sedimentary systems, see for example Galloway (1996), Miall (1996). One may dream that one day, all major classes of depositional systems would be backed with one or more large training images in digital format. In addition to the traditional qualitative description with typically few supporting statistics, each class will be backed by a high resolution, large extent, numerical and visually explicit, description of how that system may look like in 3D, possibly at various scales calling for various Ti's. Again these Ti's are purely conceptual, they need not locally match any data from any specific deposit. Non-stationarity and, more generally, overly specific aspects of these Ti's would have to be dealt with, see next subsection.

The existence of such catalog would allow sharing prior structural information in a directly usable digital format: any company geologist would page through that catalog and extract those Ti's that express best and bracket his own "vision" of what the deposit under study may look like. Multiple point geostatistics and simulation would then try to match that prior vision to the field data available, both hard (e.g. wells) and soft (e.g. seismic).

Stationarity of training images:

A training image generated as a non-conditional realization of a stationary and ergodic process will look "stationary", in that our perception of structures and shape will appear invariant under small translation of our eye window or template. For example, an object-based realization of the distribution of ellipses, over a field of dimensions much larger than the size of ellipses or their clusters, is stationary; see Figure 5a. That distribution is not conditioned to any local data hence no area of the Ti appears more particular then another.

The situation is much different with any Ti based on an actual photograph, of a present day deposition for example, see Figure 5b. An actual deposition pattern has necessarily location-specific, non-repetitive, hence non-stationary features; it also typically display multiscale structures, with the larger scale extending over the entire figure, hence these large scale structures are also non-repetitive and non-ergodic. If such pictures of real phenomena are to be used as Ti's, because they are to a certain degree non-stationary and non-ergodic, what "average" structural information can we expect to extract from them, e.g. by scanning the Ti with a fixed size template? Clearly, local non-repetitive features will be averaged out, large scale non-ergodic features would not be seen through the too narrow window. How could those missed-out features or structures be re-introduced in our simulated realizations?

Figure 6a gives a non-stationary image of what could be an interpreted binary map of channels in a delta fan. That map is non-stationary in that the direction of channeling varies in the delta, and it is non-ergodic in that the channel continuity extends across the entire image. Figure 6a was used as Ti to generate through snesim the non-conditional (no local data) simulated realization shown in Figure 6b, see also Zhang (2002). The non-stationary locally varying channel directions of the Ti were averaged out in the simulated realization. One way to re-introduce the large scale fan structure of the Ti is to condition the simulation by local data giving in each area the general direction of the channels, see Figure 7b and later discussion on "Local transforms". Short of such local data about

direction, the prominent but non-stationary fan structure of the Ti of Figure 6a cannot be recovered.

4 Snesim v.10.0 – State of the Art

The snesim code, based on a strict search tree record of the training image, is presently (2003) the most developd mp simulation algorithm. Other algorithms based on some form of classification of the Ti DEV's are still in various stages of development, see Arpat (2003), Zhang (2003), both in this SCRF report.

Snesim stands for single normal equation simulation. That single normal equation expresses the projection of the indicator of presence or absence of a category (variable to be simulated) onto the linear space generated by the single multiple-point data event grouping all relevant data (Journel, 1993; Strebelle, 2002). The basic inference relation (7) identifying the conditional probability to a training proportion is but an expression of that single normal equation.

The snesim version 10.0 is fully flowcharted and described in Zhang and Journel (2003, this SCRF report). A detailed sensitivity analysis of all input parameters is given in Liu (2003, this SCRF report).

Figures 8 and 9, taken from the previous 2 papers, give the flowchart and input parameter file of snesim v.10.0. The following short review intends to highlight the major algorithm decisions of snesim and discuss their shortcomings while suggesting alternatives.

Flowchart:

From Figure 8 it appears immediately that the loop giving different simulated realizations is at the top of the flowchart, embedding all other loops including that constructing the search trees. Hence all search trees will be constructed anew for each new realization, although they are the same since the same training image is used. This major programming decision was taken for 2 reasons:

1. the trend in modern geostatistics is not to assess uncertainty with many realizations of the full field facies and properties. A few realizations (2 or 3) suffice to check that the input parameters are correct and that the simulation algorithm chosen delivers the expected results. Uncertainty is to be assessed earlier regarding critical input

parameters such as target Net to Gross ratio and choice of the Ti, see Bitanov and Journel (2003), Kim and Caers (2003), both in this SCRF report.

2. because of multiple grids and local transforms (see hereafter), the number of search trees needed for any single training image is large. RAM storage becomes an issue if all these search trees (same for all simulated realizations) were to be loaded upfront. Because construction of a search tree is reasonably fast, it was decided to repeat that construction as often as needed, unload from RAM the search tree before constructing the next one; this was deemed faster than reading these search trees from external storage devices.

The other point apparent from the flowchart of Figure 6 is the assignment of sample data to the nearest (sub)-grid nodes, followed by their unassignment then re-assignment when another multiple or sub-grid is considered. This creates data location inaccuracy and, of greater concern, possible inconsistency from one grid to another. The present definition of a data event requires hard data values at the nodes of the data template, hence the location assignment chore. In section 5, we suggest interpolating probabilities at these template nodes which has two major advantages (1) hard data stay where they are, (2) the data template is always full but with probability values.

Multiple grids:

The concept of simulation on successively finer grids, passing simulated values as hard data from one grid to another, is key to reproducing large scale structures (Tran, 1994). Indeed, simulation starts on a coarse grid which allows using a large size data template but coarsely gridded. The number of multiple grids to retain depend on the largest scale of the training structures to be reproduced: the most difficult case is that of structures spanning the entire extent of the field to be simulated. In such case, experience has shown that 4 to 5 nested multiple grids, each with a coarsening factor of 2 in each 2D direction (thus a factor of 8 in 3D), is adequate.

Training image size:

For reasons of ergodicity (number of replicates) the training image extent should be at least twice the dimension (four times would be better) of the largest scale structures to be reproduced. For example, if channels spanning the entire EW length L of a simulated field are expected, the Ti should be large enough to display channels of EW dimension

2L, again 4L would be better. This may call for extremely large Ti's; the solution is again to use large but coarsely gridded Ti's for simulation at coarse grids.

The multiple grid simulation approach calls for multiple different search trees, one per grid specific to the data template retained for that grid. Recall that the size (number of nodes) of a search tree is related to the size of the data template, not to the size of the Ti; the number of replicates is related to the Ti size.

One single large Ti, as finely discretized as the finest simulation grid, could be used to generate all multiple search trees including that corresponding to the coarsest and largest data template. This would be the ideal solution provided that single Ti displays all structures to be reproduced, from smallest scale to largest scale. Such large and finely discretized Ti may be too memory demanding and the building of the corresponding multiple scales search trees may be time-consuming. Remember, however, the concept of a catalog of Ti's and related search trees repositories of DEV's: a search tree should be an upfront, one-time cost.

The other solution is to consider different Ti's at different scales for the different simulation multiple grids. The larger scale Ti's would be coarsely gridded and could display very different structures/shapes than the fine scale, densely gridded Ti's. Beware though of consistency between these different Ti's.

Data template issues:

The demand in memory (RAM) and processing time (cpu) of a snesim simulation is conditioned mainly by the data template size. The number of node n of that data template determines the size (RAM demand) of the search tree, and also the cost of constructing it (an one-time cost) and reading from it (a cost proportional to the size of the simulation grid).

At present the snesim approach accepts experimental DEV's that may only partially fill the data template: not all nodes are informed. Whenever a data template node is not informed, all K possible outcomes for that missing datum are considered and the corresponding conditioning probabilities are averaged; therefore the search tree is read K times starting from the tree node corresponding to the template uninformed node. Thus the more incomplete the experimental DEV, the more cpu time it takes to retrieve from the search tree the corresponding conditional probability distribution (pdf); and this grows exponentially worse as the template size n increases and the number K of outcome/classes per node increases.

Consequently, a major gain in speed would be obtained if all experimental DEV's were full. One solution, that taken by the neural net approach of Caers (1998), is to fill-in all simulation grid nodes at the start: this can be done by drawing from the marginal target distribution of the attribute variable. Sequential simulation would then iteratively perturb the central value of each template conditional to the present template full DEV. What is gained in speed from having a full DEV is lost by the necessity of iterating several times through the entire simulation grid, not to mention issues of convergence.

Strebelle (2002) has suggested defining subgrids within each of the simulation multiple grids. The simulation path proceeds on each of these subgrids in such a way to maximize the number of data template nodes informed from previously simulated nodes. This subgrid solution has been implemented in SCRF version 10.0 of the snesim program. However, subgrids call for the burden of additional search trees.

In the next section 5, a new snesim algorithm is proposed which always considers full data templates, but data templates informed with probability values instead of hard simulated values. The key modification is to be able to use probability values, instead of hard outcome values, to visit a search tree.

As for the geometry of data templates, experience has confirmed the established practice of data search neighborhoods used in kriging, which is isotropic template geometries are just fine. This is true even for modeling highly anisotropic elongated structures, such as fluvial channels.

Local transforms:

As discussed before in section 3 – Stationarity Issues, stationary training image can be rotated then squeezed (affinity transform) as needed, see Figure 7. The experimental DEV can be transformed (rotation + affinity) from its actual data locations to adapt to those of the single stationary Ti; the problem is that such transform calls for data relocation with the consequent inaccuracy which may be severe for the largest coarsest multiple grid.

The snesim version 10.0 has taken the alternative solution which is to transform the original Ti: more precisely if 5 classes of angle transform and 3 classes of anisotropy ratio are considered, 15 different search trees will be generated from the single original Ti, for each multiple grid. Hence, if that transform is to be applied at all scales and 5 multiple grids are considered, there will be a total of $15 \times 5 = 75$ different search trees. If a prior catalog of Ti's and search trees does not exist (present situation), the management (construction, loading into RAM, unloading) of so many search trees is demanding

and cpu-time consuming. In 3D, a local transform may need 2 rotations, hence 2 angle parameters (one for azimuth, the other for dip identification), and 2 affinity transforms, hence 2 anisotropy ratio parameters, for a total of 4 parameters. The total discretization of the corresponding 4 dimensional parameter space should not much exceed 25 to limit the number of corresponding search trees; but 25 corresponds to an extremely coarse discretization. Fortunately, in subsurface applications, data are rarely abundant or accurate enough to allow a high resolution definition of local anisotropy directions and corresponding anisotropy ratios.

Target proportions:

The global proportions or marginal histogram of the attribute variable over the various multiple grid Ti's need not identify the proportions targeted for the simulation. If the difference between Ti proportions and target proportions is not too different, a servomechanism allows correcting gradually the conditional proportions towards meeting the target proportions, as the sequential simulation progresses.

Beware however, that if the target proportions are too different from the Ti's proportions, this target may not be anymore consistent with the structures displayed by the Ti. Consider the binary variable indicator of presence/absence of channel in a mud background: a Ti displaying thin undulating fluvial channels in proportion 0.2 is not consistent with a target channel proportion of 0.5 or more.

In snesim v.10.0 the servomechanism can be turned up from 0 (no correction) to 0.99, the maximum 1 entailing exact target reproduction ignoring the Ti information. The program also allows variable vertical target proportions; this is useful for simulation of 3D layered reservoirs where target facies proportions may vary from one horizontal layer to another.

Performance:

To provide a rough measure of the relative cpu costs of the varions steps of the snesim algorithm, run times for generating a 2D simulated realization of the type of Figure 7c are:

Code: Still experimental (not optimized) C^{++} version of snesim v.10.0, Linux OS.

Hardware: 1.7 GHz Intel Pentium 4, 512 Mb RAM

Ti and simulated field size: 250×250 nodes, 5 multiple grids

Template size: 25 nodes for each multiple grid, 3 rotation angle categories, and 3

affinity ratio categories, 2 subgrids for each of the 4 finer multiple grids, thus 9 search trees are generated for the coarsest grid and $9 \times 2 = 18$ for each of the 4 other multiple grids

Total time: 172 seconds distributed as:

- 165 s. for inference (rotation + affinity + search tree) construction, reading and unloading out of RAM
- 5 s. for actual simulation
- 2 s. remaining

Remarks:

There is a total of 81 search trees generated, thus the time spent per search tree is about 2 seconds $\approx 165/81$. This is clearly where algorithm optimization should take place. Ideally, the search tree (s) construction should be an upfront task performed outside the loop for each realization (see Figure 8); if a catalog of such search trees already exists, then the cost of search tree construction would be replaced by that of I/O reading from hard disk.

The cost of reading the conditional probabilities would be considerably reduced if the data template was aways full (all its nodes are informed), see proposal in the next section 5.

Our conjecture is that algorithm modification and code optimization could cut the run time given above by a factor of 10 (more in Fortran) allowing generating a 3D multimillion nodes realization within 10 to 20 min. within the next 2 years.

5 Improving the snesim algorithm

Strebelle's snesim algorithm will remain a landmark in the development of geostatistics being the 1^{st} operational multiple-point simulation algorithm. The rigor and simplicity of the search tree concept combined with the well established sequential algorithm have proven to be a remarkable match.

As is often the case, one's strength is also one's weakness:

- the search tree allows retrieving the number of training image data events matching exactly the experimental one, but then it does not allow interpolating between Ti data events. There is no measure of similarity between DEV's; either they match exactly or valuable data must be dropped until such exact match is obtained.

- the sequential simulation is non-iterative, hence fast: each uninformed node is visited only once, its simulated value is then frozen as hard data. This can create conflicts and discontinuities when nodes on a coarse grid are simulated independently one from another due to non-overlapping data templates.

- pointwise algorithms proceed simulating one point at a time, as opposed to object-based algorithms. This allows for easy data conditioning since it suffices to freeze those points/grid nodes that are already informed, but sets of contiguous simulated values may not anymore display the crisp geometry of expected objects.

Consequently, the snesim algorithm suffers from the present drawbacks:

- it requires large, difficult to build, training images able to display the large variety of DEV's expected in a real reservoir. Large template sizes are required to capture and reproduce large scale continuity; correspondingly, very large Ti's are needed to provide enough replicates of DEV's defined on such large templates. We would like to expand from the limited DEV sample provided by a Ti. We would like also to filter that DEV sample, retaining only those DEV's reflecting the "essence" of the Ti and rejecting those associated to specific, non exportable, occurences.

- the present sequential simulation does not allow self correction, except for a post-processing of the simulated field. That post-processing amounts to a 2^{nd} iteration; we would like such iteration to be fast (few iterations without issues of convergence) and to be part of the simulation algorithm rather than an afterthought correction.

- without resorting to dropping onto the field whole objects as in object-based (Boolean) algorithms, we might consider simulating sets or whole templates of contiguous points. The initial simulation of those points might be soft, simulating probability values instead of hard values, which would make the first simulation pass "soft".

The research done recently at SCRF (2002-03) has been multiprong. Various alternatives are being tried to offset the previous limitations of the snesim approach. The pattern

simulation (program simpat) research line of Arpat (2003, in this SCRF report) is possibly the most innovative. Beyond details, simpat provides the following critical improvements over the present snesim program:

1. simpat matches experimental DEV's to classes of training image DEV's. That match involves probability values at each node of the data template instead of hard values as in snesim. Such prior classification of training DEV's and soft match allows for interpolation and filtering of the Ti DEV's.

2. from one node to another, from one multiple grid to another, simpat passes probability values as opposed to the hard values passed by snesim. Thus, early simulated values are not frozen at the risk of generating later discontinuities. Hard simulated values are generated by simpat only through a 2^{nd} pass over all nested guides: this amounts to an iteration.

3. in the simpat algorithm all nodes of each template are simultaneously simulated as probability values, as opposed to snesim simulating a hard value at the sole central location of the template. Simulating all nodes of each template guarantees within-template continuity, then because these values are probabilities they are easily updated when a second template overlaps the first one.

Point 1 above amounts to a classification of DEV's which calls for defining a distance measure between any two DEV's. Classification voids the need for a search tree, a concept at the very basis of the snesim algorithm. Points 2 and 3 do not call for any classification and could be accommodated were the snesim algorithm be expanded to:

2a. accepting, as data, probability values, then passing updated probability values instead of hard values. Hard values would be generated/simulated only during a 2^{nd} pass through the simulation grids.

3a. the probability values at all locations of the template should be updated simultaneously instead of the present snesim simulating only the central location.

Leaving aside for now the critical issue (Point 1) of interpolation and filtering of Ti DEV's, an issue associated to the definition of what constitute the yet undefined concept of "essence" of a Ti, the following proposed modifications would allow snesim to accept and pass probability values.

A soft semi-iterative snesim algorithm:

Presently, snesim accepts and passes only hard data values, 0 and/or 1 in the case of a categorical variable; such hard data values can be seen as extreme probability values.

Passing probability values to the central location of each template is easy, it suffices to pass the conditional proportion value(s) read from the search tree instead of drawing a hard simulated value from it. Accepting probability values is a tougher challenge because the search tree is a repository of hard, not probabilistic, Ti DEV's.

We suggest the following new semi-iterative snesim algorithm based on two stages, a sequential probability updating stage to fill-in all nodes of all grids with mutually consistent probability values conditioned to the original sample values, and a simulation stage similar to that of the present snesim algorithm, see flowchart in Figure 10.

(1) Probability updating stage:

Consider again Figure 4 and the structure of a search tree. Each node of that tree corresponds to a particular location within the data template with K branches starting from it, if there are K possible categories for the variable being simulated. The hard datum value at each template location determines which branch to take to reach the next node of the tree; if that datum value is a probability value (the K branches probability values summing up to 1), one can draw from that probability distribution the indicator of the branch to take. We would keep drawing at each node of the template to determine the next branch/route and go as far(*) as the search tree allows. Note that an original sample value frozen as a hard datum value at any given node corresponds to the particular case of a deterministic draw with probability 1 for a specific branch. Once all locations(*) of a specific template centred at $\mathbf{u}$ have been so visited, a specific hard DEV has been identified in the search tree, the corresponding central value conditional probability is retrieved and used to update whatever probability value was at that central value. That probability value is then passed along (no drawing from it!).

Remarks

(*). At each grid node $\mathbf{u}$ the final conditioning hard DEV generated is of size $t(\mathbf{u}) \leq T$, where T is the template size. Indeed the search tree generally does not contain all possible K^T DEV's, i.e. all branches starting from the tree root do not have maximum length T.

• The previous route simulating one's way through all branches of a data template requires a fully informed template, with all its T locations informed with a probability distribution

function (pdf). These pdf's are either passed from the previous coarse grid, or they can be interpolated from an indicator kriging based on neighboring original hard sample data and coarse grid pdf data. Actually a much faster inverse-indicator variogram weighting interpolation should be sufficient since these interpolated pdf values will be updated with search tree probability values.

• Original hard sample data values exactly located on a grid node are frozen there, i.e. with a prior pdf equal to 0 and 1. These hard sample nodes are skipped in the sequential simulation, that is they are never updated. If a hard sample datum is not located exactly on a grid node, it is **not** relocated: it will influence the previous pdf interpolation process from its actual location.

• **Updating template pdf values**: Once the template central value at $\mathbf{u}$ has been updated with a pdf read from the search tree, one should update its neighbor template pdf values for consistency. The following sequential algorithm is suggested for that updating:

Let $i(\mathbf{u}+\mathbf{h}_\alpha)$ be the indicator values drawn at the $t(\mathbf{u})$ template locations $\mathbf{u}+\mathbf{h}_\alpha$, $\alpha = 1, \cdots, t(\mathbf{u})$; these indicator values define the path taken through the branches of the search tree, hence they define the pdf $p(\mathbf{u})$ attached to the central location $\mathbf{u}$ of that template; $p(\mathbf{u})$ is read from the search tree. Using the classical sequential algorithm, at each within-template location $\mathbf{u}'_\alpha = \mathbf{u}+\mathbf{h}_\alpha$, $\alpha = 1, \cdots, t(\mathbf{u})$,

- discard both the prior pdf $p(\mathbf{u}'_\alpha)$ and the indicator $i(\mathbf{u}'_\alpha)$
- position a new template centred at $\mathbf{u}'_\alpha$
- apply the previous algorithm, using the neighboring indicator $i(\mathbf{u}+\mathbf{h}_{\alpha'})$, $\alpha' \neq \alpha$, or drawing from pdf values when such indicators are not present, to determine the conditioning DEV for that new template
- read the corresponding conditional probability from the search tree and update $p(\mathbf{u}'_\alpha)$. Draw from that probability a new indicator value $i(\mathbf{u}'_\alpha)$
- end this inner loop when all locations $\mathbf{u}_{\alpha'} = \mathbf{u}+\mathbf{h}_\alpha$, $\alpha = 1, \cdots, t(\mathbf{u})$ of the original template have been visited and their probabilities and indicators are updated.

The central location $\mathbf{u}$ and its template neighbor locations $\mathbf{u}'_\alpha$, $\alpha = 1, \cdots, t(\mathbf{u})$ are **excluded** from the random path visiting the present multiple grid, i.e. they are not anymore updated on that grid. They will be at the next finer grid.

Proceed to the next location $\mathbf{u}$ not excluded along the random path and repeat the previous process, i.e. update the prior pdf at $\mathbf{u}$ and all its within-template location $\mathbf{u}'_\alpha$ which have **not** been updated yet. Hence over anyone of the multiple grids, a node location will have its prior probability values updated once and only once. That updating uses probability values from the search tree, hence pdf values that carry the mp statistics of the training image. Only those grid nodes exactly co-located with an original hard sample datum are not updated, their probability (0 or 1) are hard and frozen never to be changed, ensuring data exactitude.

This algorithm

- accepts and passes probability values at each grid node
- updates the prior probability values once and only once within each multiple grid
- updates consistently (in an inner loop) the template central value and its non-already updated template neighbor locations.

When a grid is completed, all its nodal probabilities are passed to the next finer grid. Each node of that finer grid not informed is first interpolated for a prior probability value. The process is then repeated to update **all** nodes of that finer grid once and only once. Note that all probabilities passed from the coarser grid are updated again.

When the last (finest) multiple grid has been fully updated, we have a fine grid filled with probability values, with the only exception of hard sample data frozen whenever their locations coincide with a grid node. The second stage of actual simulation can now proceed.

(2) Simulation stage:

Starting again with a random path visiting the coarsest grid, the previous process of gathering pdf values from the search tree is repeated, but this time:

- a hard simulated value is drawn from each pdf read from the search tree. That hard simulated value is then frozen (never to be changed) and passed along.

The inner loop consisting of simulating hard values at all within-template locations, before proceeding to the next central location, is maintained. This should allow consistency of close-by simulated values on any grid.

The hard simulated nodal values of one grid are passed to the next grid. Each template of that finer grid is full with either hard data (from the coarser grid) or probability values. The previous simulation process is repeated, again with hard values drawn from the search tree pdf's. The process continues down to the last and finest grid: one fully simulated realization of the study field has been generated.

In summary:

The proposed new snesim algorithm includes two stages:
(1) a probability updating stage where a consistent probability field is generated with a progressive probability updating as read from the search tree. Probabilities are accepted and passed along. At this stage the probabilities are used only to draw the route within the search tree for each template.
(2) a simulation stage where hard simulated values are drawn from the probability field after an ultimate 2^{nd} updating pass reading from the search tree.

Awaiting for a better shorter name this proposed new algorithm is called "soft semi-iterative snesim" in that:

- it remains based on a search tree which is a faithful record of the training image DEV's, hence the appellation snesim

- initially, probability values are passed from one grid node to another, hence the qualifier "soft"

- those probability values are sequentially updated to ensure within-template consistency at all multiple grids, however this iteration is not controlled by any dubious convergence criterion, hence the qualifier "semi-iterative".

6 Concluding remarks

When modeling a spatial phenomenon, whether a mineral deposit or an hydrocarbon reservoir, there is much more to information than the local sample data available. These

local data would have triggered a "structural" interpretation about how these data relate one to another and to the unsampled values. That interpretation, if based on prior experience in studying similar deposits, represents extremely valuable information. Any interpolation or simulation process aiming at filling-in values between sampled data **necessarily** calls for a structural model relating data and unknown values. That structural model should capitalize on the prior structural interpretation of the expert interpreter, a geologist or geophysicist. The impact of the structural model chosen is absolutely critical in early exploration stages when actual (local) data are few. Too few data can't speak for themselves, and it would be foolish not to capitalize on prior experience because it is deemed uncertain. Accepting the prior structural model uncertainty and trying to assess it, is better than accepting models based on arbitrary criteria such as maximum smoothness of the interpolated surface or, at the other extreme, maximum entropy of the data all the way to data independence.

The problem with using prior structural information is that this information is fuzzy, typically not numerical nor parametric: it does not take the form of a variogram or a histogram of channel width and sinuosity parameters. When it comes to prior experience and structural information, geologists think in visual terms of shapes, patterns of spatial distributions, what we proposed to call a "training image". Training images are conceptual visual representations of how heterogeneities could be distributed in the actual deposit or reservoir. The fuzzy conceptual image of the geologist could be ascertained by having him choose from a catalog of visually explicit, numerical, training images, or helping him construct the training image(s) best depicting his prior vision using object-based (Boolean) simulation algorithms. That phase of choosing or building a training image, i.e. elaborating a structural model, should not be encumbered with concerns of local accuracy: the actual local data need not (should not) be honored at their exact locations. One should see our proposed reservoir modeling flowchart as constituted of two distinct phases:

1. elaborating a structural model that relates data and unknowns, more precisely a numerical, visually explicit, training image of what heterogeneities are deemed to look like in the actual reservoir

2. applying that structural model to the actual data. Instead of gradually "morphing" the training image to match these data, we suggest to construct stochastically the actual reservoir numerical representation(s) one pixel at a time accounting for the

multiple-point structures displayed by the training image.

That dichotomy between structural and local information is pervasive in geostatistics. In kriging, the structural information relating data to unknown is the variogram; in simulation it is the random function model traditionally characterized by variogram(s). The kriging or stochastic simulation process consist of filling-in one pixel at a time the unsampled grid nodes. The limitation of traditional geostatistics come from the structural model being limited to 2-point statistics which can relate only two data locations together or one datum location with one unsampled grid node. This is fine for modeling amorphous heterogeneities such as distribution of grades or petrophysical properties within a statistically homogeneous region or facies, it is insufficient for modeling categorical variables associated to shapes and geometrical patterns. There is much more to facies and shape distribution than 2-point statistics, and that "more" can be delivered by training images selected or built from prior geological expertise.

Instead of borrowing specific multiple-point statistics from a training image, such as 2-point, 3-point, $\cdots$, n-point statistics and using them to construct conditional probability distributions, the avenue presently chosen in modern geostatistics is much more direct: borrow directly those conditional distributions. The training image is scanned for replicates of the multiple-point data event, the relation of these replicates with any neighboring training values provide the required structural information and, directly, their distributions conditional to the data event. How a training image is scanned, how the resulting structural information is stored in a search tree, which templates for the data events should be retained, are all critically important details of implementation. The important novel methodological stance is accepting the numerical training image(s) representation of the random function structural model, as opposed to the traditional analytical models (all Gaussian-related) or the implicit, algorithm-driven models (e.g. random functions associated to most iterative simulation algorithms).

A crucial consequence, not yet fully developed, of the reliance on training image-defined structural models is an healthy return to what really matters when it comes to uncertainty. The major source of uncertainty, that which may create severe biases and affects the 1^{st} digit of results, is related to the choice of the training image or structural random function model. It is **not** the fluctuations between the various realizations of the final reservoir representation, those fluctuations being associated to the various ways the same prior structural model can be made to match the local data. The major uncertainty, particular in early exploration stages, is in the structural model/training im-

age retained. Geostatisticians will do well toning down their touting of unconsequential random seed-based fluctuations of simulated realizations and focus on the essential, the uncertainty about the structural model retained. Then, because there is much more to a structural model then mere 2-point statistics, pertubation of a variogram model is almost always a futile exercise. What should be done is consideration of widely different geological scenarios, different structural models, yet all triggered by the same local data. The consequence of geological scenario uncertainty typically overwhelms that due to within-scenario uncertainty; the latter could be taken care of by the fluctuations of geostatistical realizations conditioned to the local data.

References

[1] Anderson, T. (1958), An introduction to multivariate statistical analysis, publ. Wiley, N.Y.

[2] Arpat, B. (2003), A patter recognition approach to multiple-point simulation, Report no. 16, Stanford Center for Reservoir Forecasting, Stanford University

[3] Berendsen, H. & Stouthammer, E. (2001), Paleogeographic development of the Rhine-Meuse delta, the Netherlands, a publication of the Geography Dpt., Utrecht Univ., Netherlands

[4] Berendsen, H. & Stouthammer, E. (2001), Avulsion frequency, avulsion duration and interavulsion period of the Holocene channel belts in the Rhine-Meuse delta, J. of Sedimentary Research, vol. 71, no. 4, pp. 589-598

[5] Bratvold, B., Holden, L., Svanes, T. & Tyler, K. (1994), STORM: Integrated 3D stochastic reservoir modeling tool for geologist and reservoir engineers, SPE paper no. 27563

[6] Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984), Classification and regression trees, publ. Wadworth, Monterey

[7] Caers, J. (1998), Stochastic simulation using neural networks, Report no. 11, Stanford Center for Reservoir Forecasting, Stanford University

[8] Caers, J. & Journel, A.G. (1998), Stochastic reservoir simulation using neural networks trained on outccop data, SPE paper no. 49026

[9] Caers, J., Strebelle, S. & Payrazan, K. (2003), Stochastic integration of seismic data and geological scenarios: A W. Africa submarine channels saga, The Leading Edge, vol. 22, no. 3, pp. 192-196

[10] Chiles, J.P. & Delfiner, P. (1999), Geostatistics: Modeling spatial uncertainty, publ. Wiley, N.Y.

[11] Deutsch, C.V. (2002), Geostatistical reservoir modeling, publ. Oxford Press, N.Y.

[12] Draper, N. & Smith, H. (1966), Applied regression analysis, publ. Wiley, N.Y., 2nd ed. 1981

[13] Earth Decision Sciences (2002), Gocad 2.0 User Manual, www.earthdecision.com, Houston, TX.

[14] Galloway, W. (1996), Terrigenous clastic depositional systems: Applications to fossil fuel and groundwater resources, publ. Springer, N.Y.

[15] Golberger, A.S. (1962), Best linear unbiased prediction in the generalized regression model, J. Am. Stat. Assoc., vol. 57

[16] Goovaerts, P. (1997), Geostatistics for Natural Resources Evaluation, publ. Oxford Press, N.Y.

[17] Haldorsen, H. &Damsleth, E. (1990), Stochastic modeling, J. of Pet. Technology, April 1990, pp. 404-412

[18] Haldorsen, H. &Damsleth, E. (1993), Challenges in reservoir characterization, AAPG Bulletin, 77(4), pp. 541-551

[19] Halmos, P.R. (1951), Introduction to Hilbert space and the theory of spectral multiplicity, publ. Chelsea, N.Y.

[20] Hasting, W. (1970), Monte Carlo sampling methods using Markov chains and their applications, Biometrika, vol. 57, pp. 97-109

[21] Isaaks, E. (1990), The application of Monte Carlo methods to the analysis of spatially correlated data, Ph.D thesis, Stanford University

[22] Journel, A.G. (1974), Geostatistics for conditional simulation of orebodies, in Economic Geology, vol. 69, pp. 673-687

[23] Journel, A.G. (1976), Ore grade distributions and conditional simulation: Two geostatistical approaches, Advanced Geostatistics in the Mining Industry, ed. Guarascio et al., publ. Reidel, Dordrecht, pp. 195-202

[24] Journel, A.G. (1983), Non-parametric estimation of spatial distributions, Math. Geology, vol. 15, no. 3, pp. 445-468

[25] Journel, A.G. (1993), Geostatistics: roadblocks and challenges, Proc. of the Geostatistics Troia, publ. Kluwer, Dordrecht, vol. 1, pp. 213-224

[26] Journel, A.G. (1996a), The abuse of principles in model building and the quest for objectivity, Proc. of 5^{th} Int. Geostat. Congress, ed. Baafi and Schofield, publ. Kluwer, Dordrecht, vol. 1, pp. 3-14

[27] Journel, A.G. (1996b), Deterministic geostatistics: A new visit, Proc. of 5^{th} Int. Geostat. Congress, ed. Baafi and Schofield, publ. Kluwer, Dordrecht, vol. 1, pp. 292-301

[28] Journel. A.G. & Huijbregts, C. (1978), Mining Geostatistics, Acad. Press, London

[29] Kim, J. & Caers, J. (2003), Joint perturbation of facies distribution and net-to-gross in history matching, Report no. 16, Stanford Center for Reservoir Forecasting, Stanford University

[30] Krige, D.G. (1951), A statistical approach to some basic mine valuation problems in the Witwatersand, J. of Chem. Metal. and Min. Soc. of S. Africa, Dec., 1951

[31] Krishnan, S. (2003), A proposal for Ph.D study, Stanford Center for Reservoir Forecasting, Stanford University

[32] Koltermann, C. & Gorelick, S. (1996), Creating images of heterogeneity in sedimentary deposits: A review of descriptive structure-imitating and process-imitating approaches, WRR, vol. 32, no. 9, pp. 2617-58

[33] Liu, Y. (2003), The practice of mp simulation using snesim v.10.0 program, Report no. 16, Stanford Center for Reservoir Forecasting, Stanford University

[34] Liu, Y., Harding, A. & Abriel, B. (2003), Multiple-point simulation integrating wells, 3D seismic data and geology, Report no. 16, Stanford Center for Reservoir Forecasting, Stanford University

[35] Luenberger, D. (1962), Optimization by vector space methods, publ. Wiley, N.Y.

[36] Mallet, J.L. (2002), Geomodeling, publ. Oxford Press, N.Y.

[37] Matern, B. (1960). Spatial variation, Meddelanden fra statens, vol. 49, no. 5, Stockholm, 2^{nd} ed. (1986), publ. Springer, Berlin

[38] Matheron, G. (1962-63), Traité de geostatistique appliquee, 2 vol., publ. Technip, Paris

[39] Matheron, G. (1965), Les variables regionalisies et leur estimation, publ. Masson, Paris

[40] Matheron, G. (1969), Le krigeage universel, Les Cahiers du Centre de Morphologie Mathématique, Fasc. 1, Fontainebleau, France

[41] Matheron, G. (1973), The intrinsic random functions and their application, Advances in Applied Probability, vol. 5, pp. 439-468

[42] Miall, A. (1996), The geology of fluvial deposits, publ. Springer Verlag, N.Y.

[43] Remy, N. (2002), A proposal for Ph.D study, Stanford Center for Reservoir Forecasting, Stanford University

[44] Roberts, E. (1990), Programming abstractions in C, publ. Addison Wesley, N.Y.

[45] Srivastava, R.M. (1992), Reservoir characterization with probability field simulation, SPE paper no. 24753

[46] Srivastava, R.M. (1993), Iterative methods for spatial simulation, in Report no. 5, Stanford Center for Reservoir Forecasting, Stanford University

[47] Srivastava, R.M. (2002), Probabilistic discrete fracture network models for the Whiteshell research area, Report no. 06819, available from Ontario Power Generation, Toronto, Canada

[48] Stoyan, D., Kendall, W. and Mecke, J. (1987), Stochastic geometry and its applications, publ. Wiley, N.Y.

[49] Strebelle, S. (2000), Sequential simulation drawing structions from training images, Ph.D thesis, Stanford University

[50] Strebelle, S. (2002), Conditional simulation of complex geological structures using multiple-point statistics, Math. Geology, vol. 34, no. 1, pp. 1-21

[51] Tetzlaff, D. & Harbaugh, J. (1989), Simulating clastic sedimentation, publ. Van Nostrand, N.Y.

[52] Tran, T. (1994), Improving variogram reproduction on dense simulation grids, Computers & Geoscience, vol. 25, no. 7, pp. 1161-68

[53] Wen, R., Martinius, A., Nass, A. & Ringrose, P. (1998), Three dimensional simulation of small scale heterogeneity in tidal deposits, Proc. of 4^{th} IAMG Conf., ed. Buccianti et al., Naples, pp. 129-134

[54] Yaglom, A. (1962), An introduction to the theory of stationary random functions, publ. Prentice Hall, N.J., Reprint (1973), publ. Dover, N.Y.

[55] Williams, (1959), Regression analysis, publ. Wiley, N.Y.

[56] Zhang, T. (2002), Rotation and affinity invariance in multiple-point geostatistics, Report no. 15, Stanford Center for Reservoir Forecasting, Stanford University

[57] Zhang, T. & Journel, A.G. (2003), Merging prior structural interpretation and local data: the mp geostatistics answer, Report no. 16, Stanford Center for Reservoir Forecasting, Stanford University

[58] Zhang, T. (2003), A PCA approach to detecting spatial patterns from a training image, Report no. 16, Stanford Center for Reservoir Forecasting, Stanford University

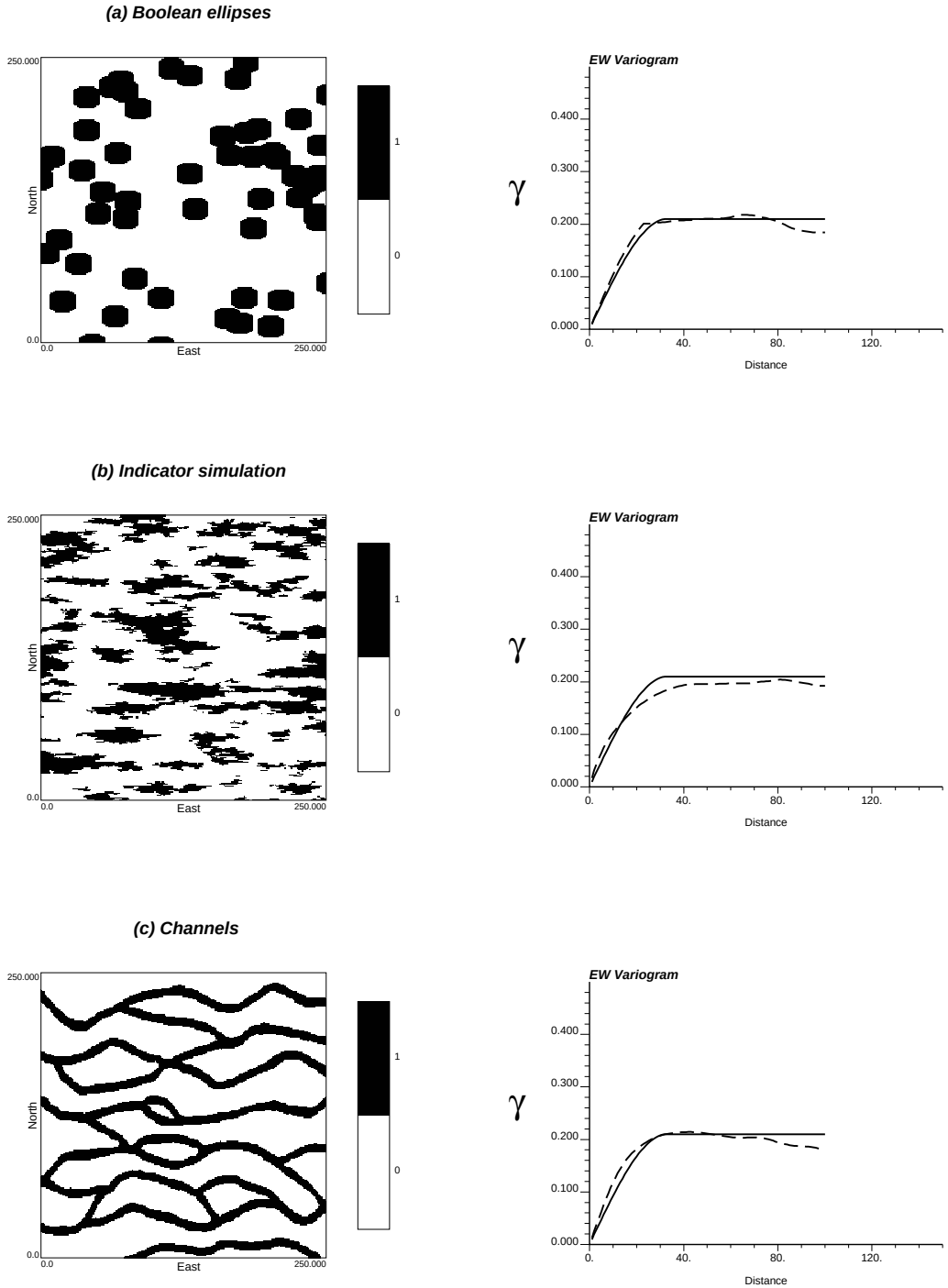


Figure 1: Different patterns of spatial variability sharing the same variogram

(a). Boolean ellipses (GSLIB program ellipsim)

(b). Indicator simulation (GSLIB program sisim)

(c). Pieces of channels (Object-based program fluvsim)

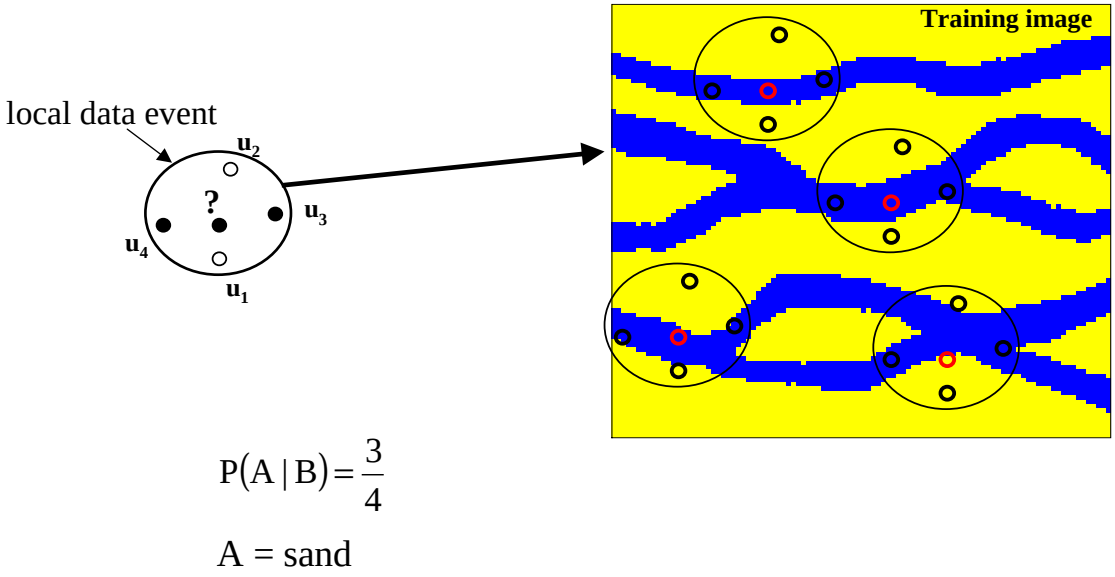


Figure 2: Scanning the training image to find replicates of a specific data event (here a 4-point DEV)

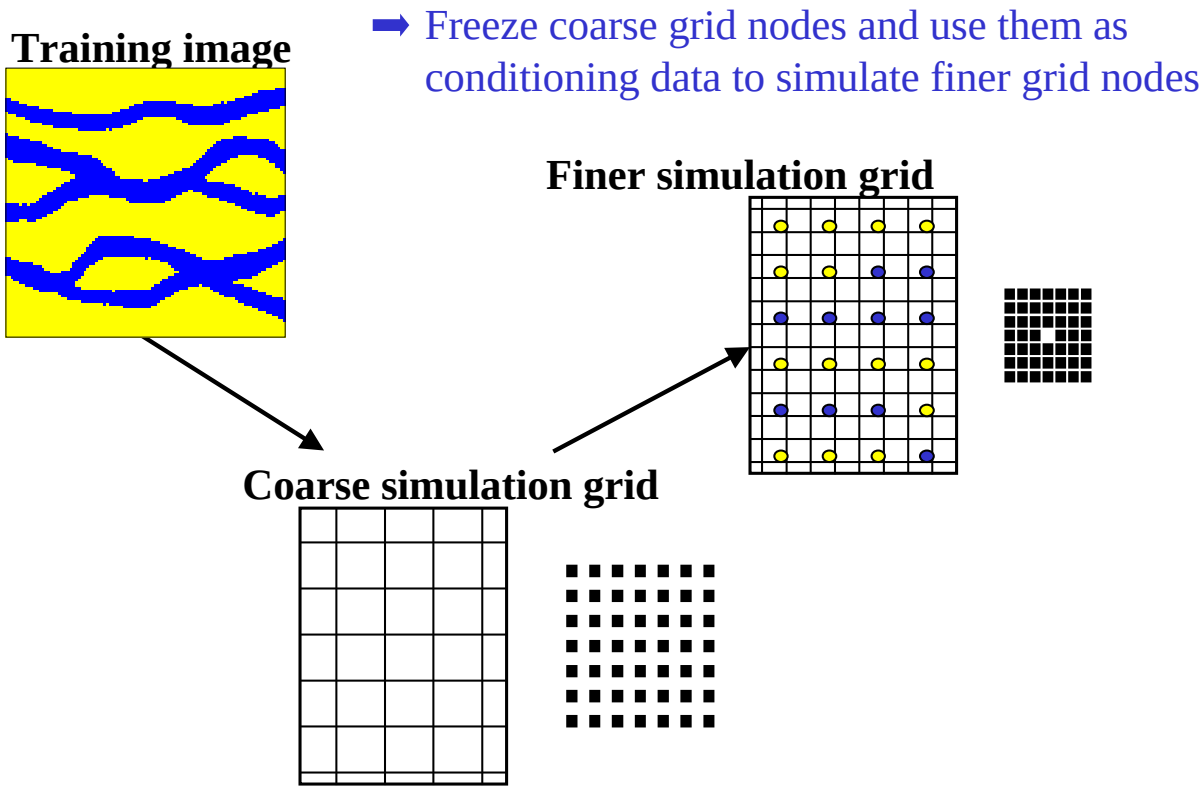


Figure 3: Multiple grids for sequential simulation and the corresponding rescaling of data template (note that the same sample value would be relocated to different nearest node depending on which multiple grid is considered)

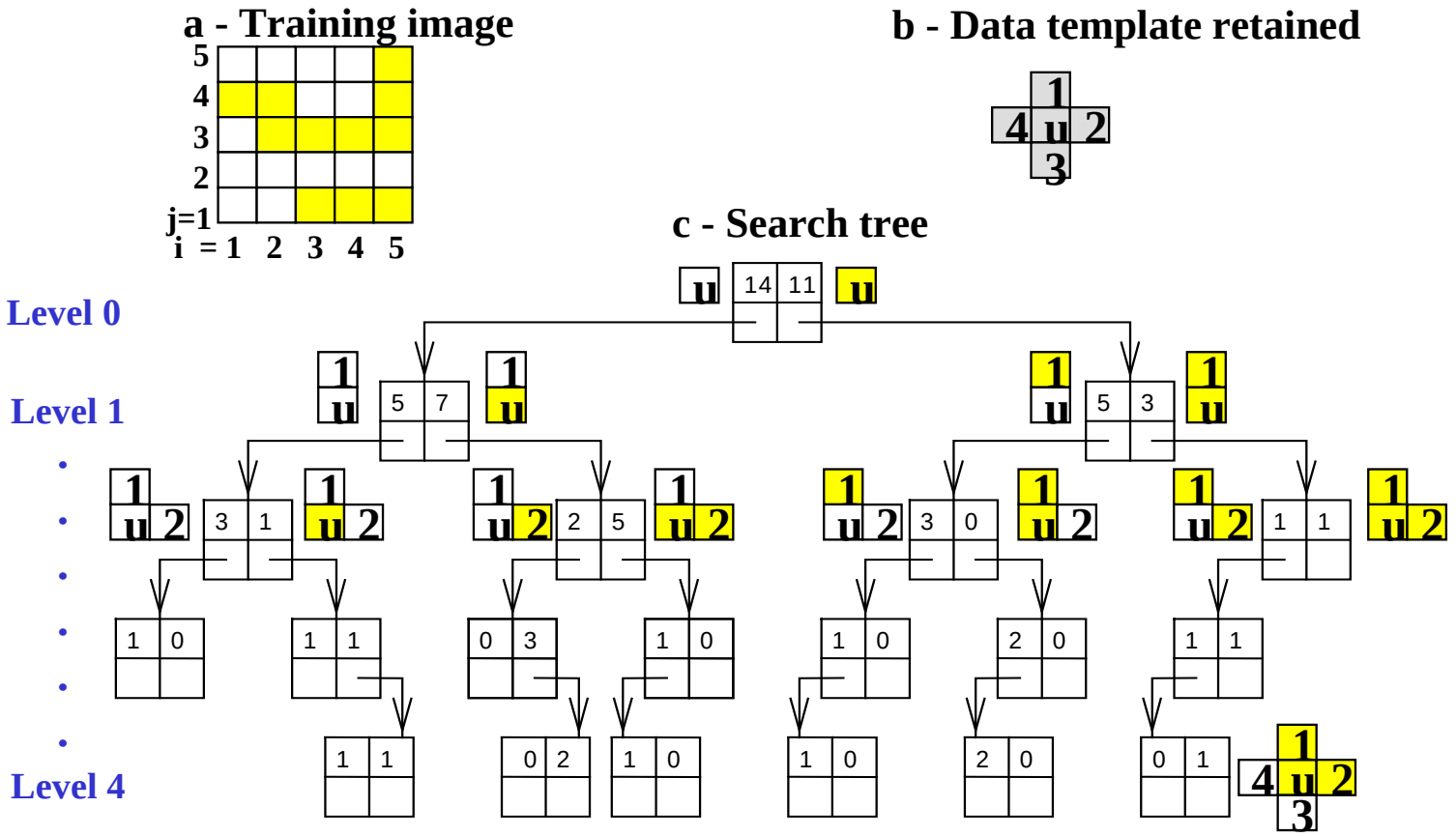
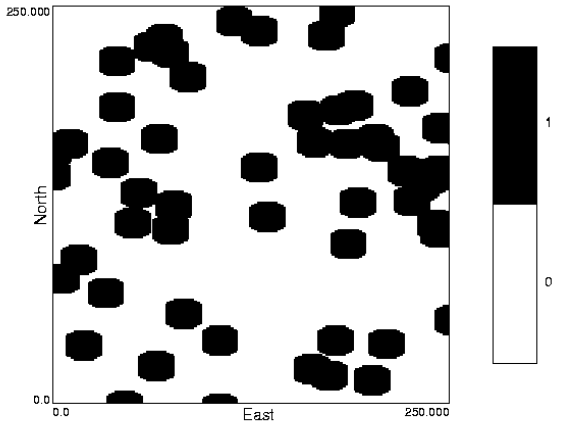


Figure 4: Ordering of grid nodes within a data template and the corresponding search tree. Template of size 4 used to scan a training image of size 5×5 generating a search tree with 4 levels of tree nodes each with 2 branches (since variable is binary) (taken from Strebelle, 2000)

a. Random distribution of ellipses



b. Rhein-Meuse fluvial channels
(from Berendsen & Stouthammer, 2001)

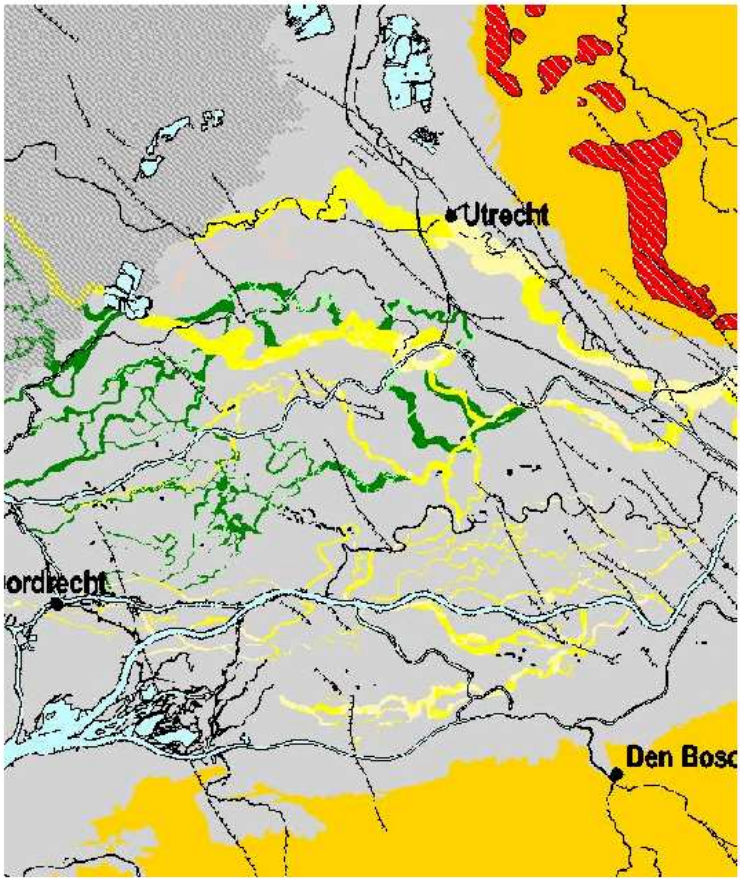


Figure 5: Stationarity of training image
(a). A stationary realization of elliptical objects of size small relative to the Ti size
(a). The actual Rhein-Meuse channel deposition displays location-specific patterns which extend over the entire Ti

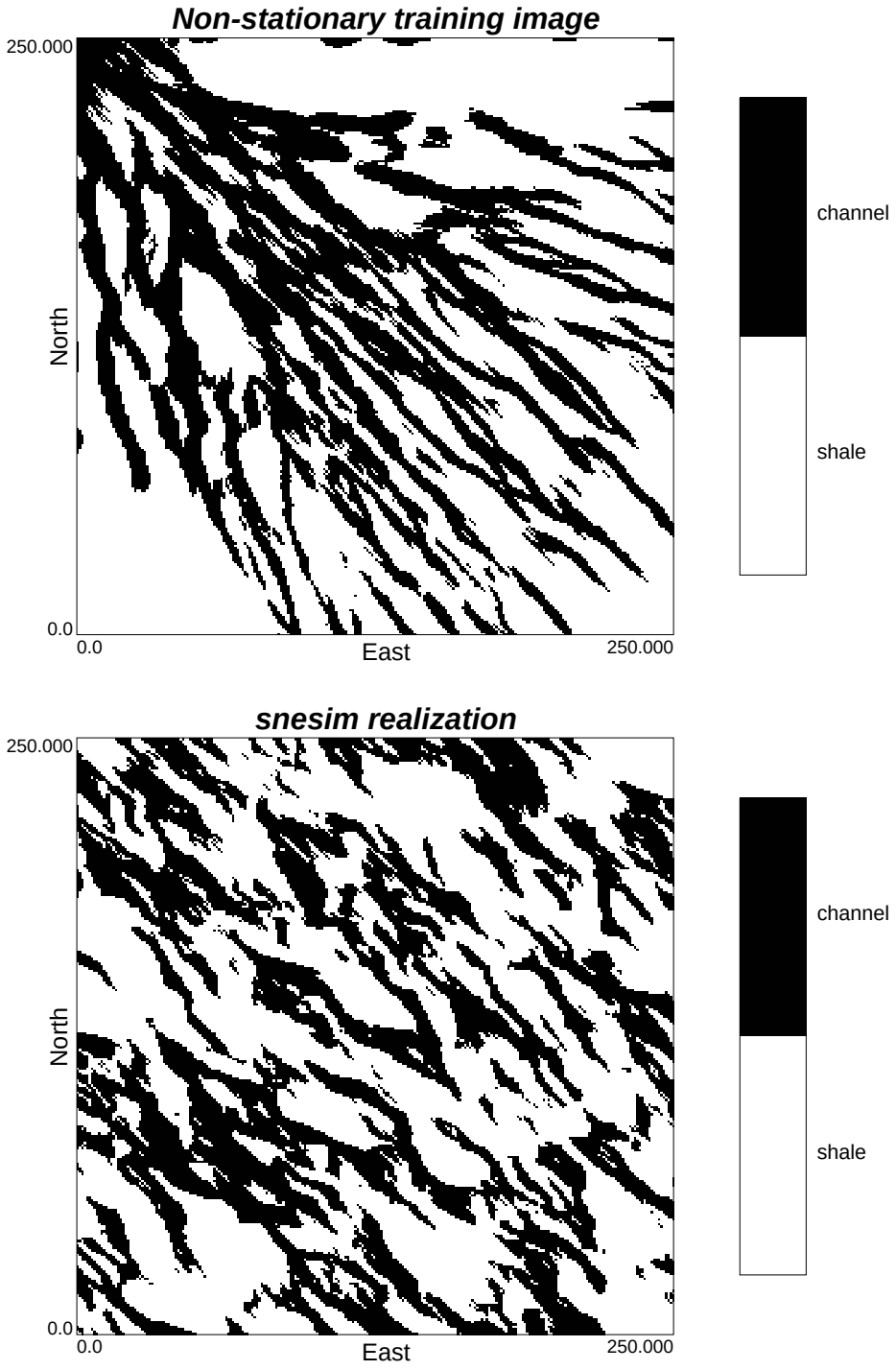


Figure 6: Non-stationary delta fan used as training image and one resulting snesim simulated realization (The non-stationary directions and channel widths of the Ti have been averaged out through scanning)

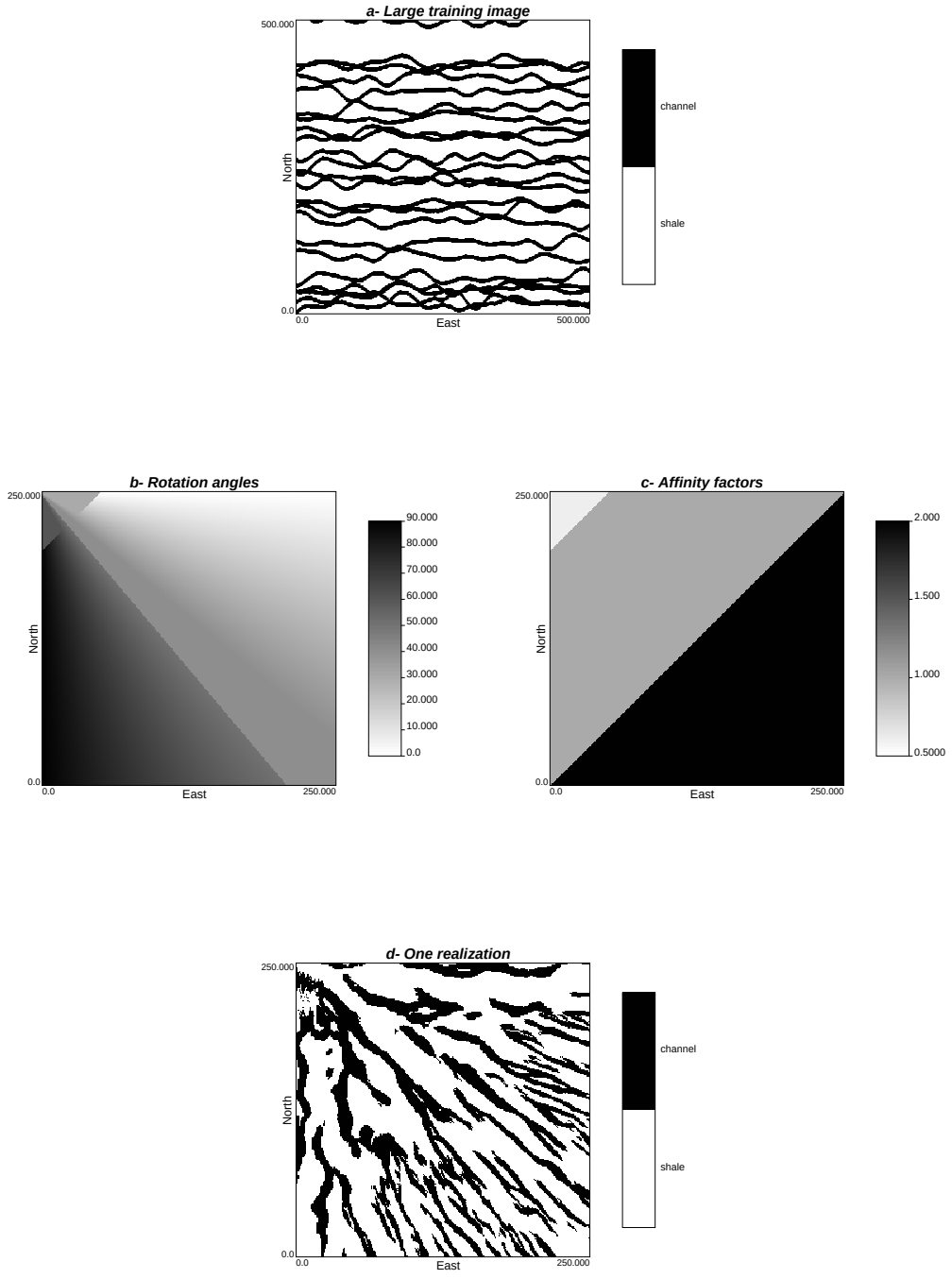


Figure 7: Stationary training image (a) and local angle (b) and affinity (c) transforms allowing generating a non-stationary delta fan (d)

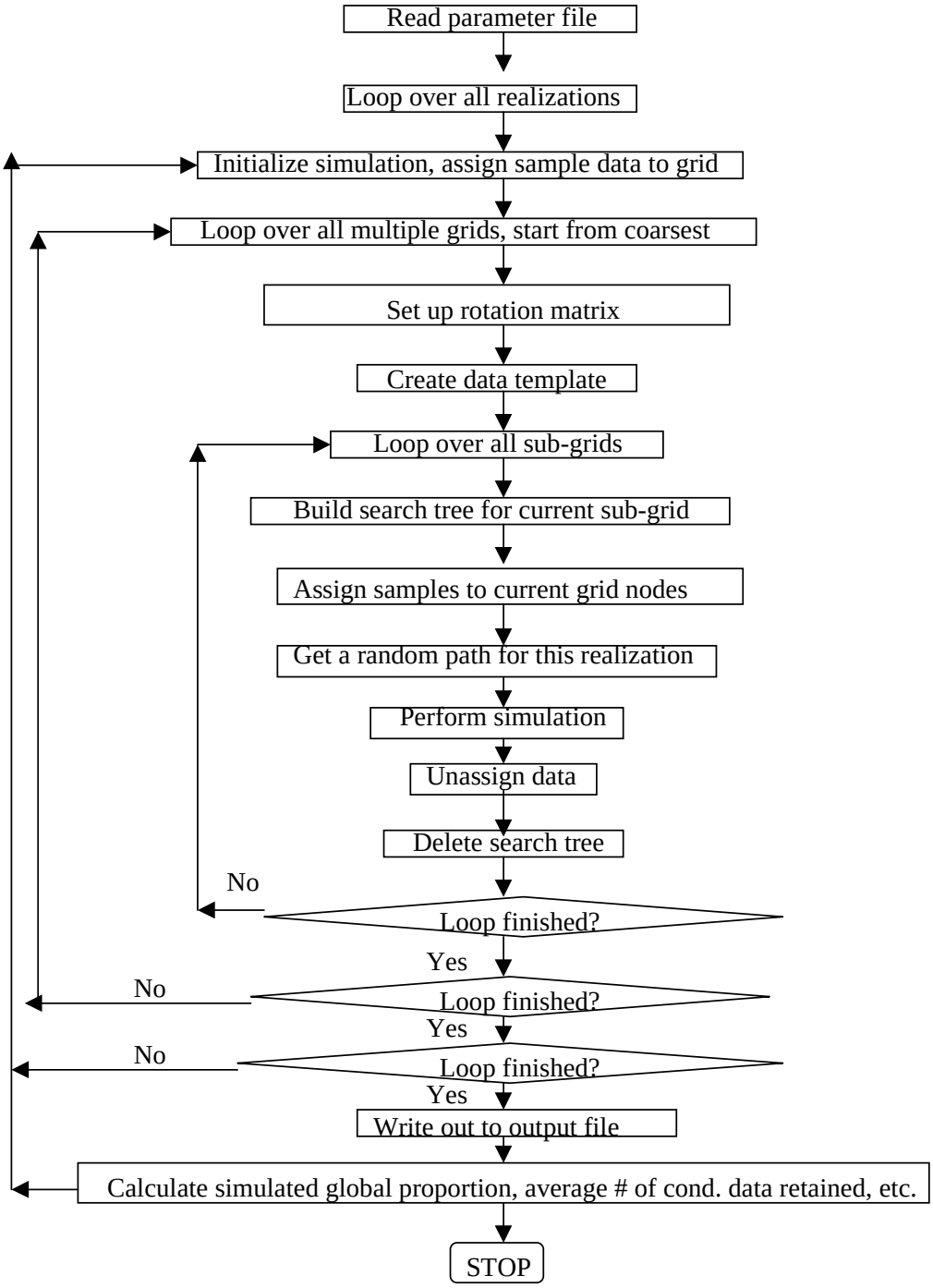


Figure 8: Flowchart of the main snesim program, v.10.0
(taken from Zhang and Journel, 2003, in this SCRF 2003 report)

```
                  Parameters for SNESIM
                  *********************

START OF PARAMETERS:
data.dat                        - file with original data
1  2  3  4                      - columns for x, y, z, variable
3                               - number of categories
0   1   2                       - category codes
0.25  0.25  0.50                - (target) global pdf
0                               - use (target) vertical proportions (0=no, 1=yes)
vertprop.dat                    - file with target vertical proportions
0.5                             - servosystem parameter (0=no correction)
0                               - debugging level: 0,1,2,3
snesim.dbg                      - debugging file
snesim.out                      - file for simulation output
1                               - number of realizations to generate
50    0.5    1.0                - nx,xmn,xsiz
50    0.5    1.0                - ny,ymn,ysiz
1     0.5    1.0                - nz,zmn,zsiz
69069                           - random number seed
16                              - max number of conditioning primary data
10                              - min. replicates number
1     1.0                       - condtion to LP (0=no, 1=yes), weight factor
localprop.dat                   - file for local proportions
1                               - condition to rotation and affinity (0=no,1=yes)
rotangle.dat                    - file for rotation and affinity
3                               - number of affinity categories
1.0  1.0  1.0                   - affinity factors (X,Y,Z)
1.0  0.6  1.0                   - affinity factors
1.0  2.0  1.0                   - affinity factors
5                               - number of multiple grids
train.dat                       - file for training image
100  100  10                    - training image dimensions: nxtr, nytr, nztr
1                               - column for training variable
10.0   10.0   5.0               - maximum search radii (hmax,hmin,vert)
0.0   0.0   0.0                 - angles for search ellipsoid
```

Figure 9: A snesim v.10.0 input parameter file
(taken from Liu, 2003, in this SCRF 2003 report)

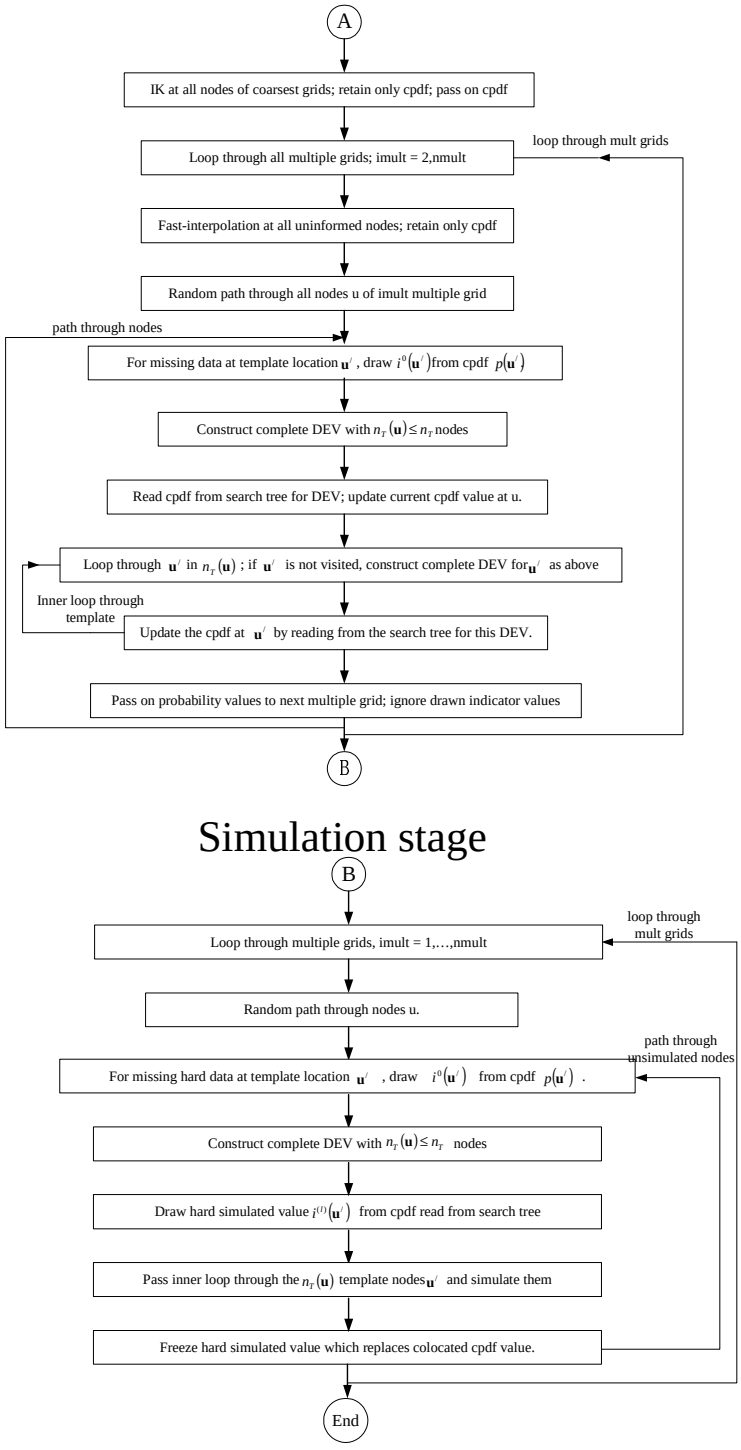


Figure 10: Flowchart of proposed soft semi-iterative snesim

(A). Probability updating stage

(B). Simulation stage

(taken from Krishnan research proposal, 2003)