DOWNSCALING OF SATELLITE REMOTE SENSING DATA: APPLICATION TO LAND COVER MAPPING

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF GEOLOGICAL AND ENVIRONMENTAL SCIENCES
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Alexandre Boucher
August 2007

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(André G. Journel) Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Karen C. Seto)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Paul Switzer)

Approved for the University Committee on Graduate Studies.

Abstract

Many satellite images have a spatial resolution coarser than the extent of land cover patterns on the ground, leading to mixed pixels whose composite spectral response consists of responses from multiple land cover classes. Spectral unmixing procedures only determine the fractions of such classes within a coarse pixel without locating them in space. Downscaling, also known as super-resolution or sub-pixel mapping, turns these proportions into a fine resolution map of class labels; this map should display realistic spatial structures without artifact discontinuities and the map should reproduce the coarse resolution fractions and any available fine scale data at their locations.

Sub-pixel mapping is cast as an under-determined inverse problem: that of reconstructing a fine scale, high resolution spatial map of class labels from a set of coarse resolution class fractions. That problem is undetermined, in that many different fine resolution maps can lead to an equally good reproduction of the available coarse fractions. Thus, the unknown fine resolution land cover map is regarded as a realization of a random set; solving the previous inverse problem amounts to drawing equi-probable realizations from that random set.

Simulated realizations are generated using the geostatistical paradigm of sequential simulation. At any pixel along a path visiting all fine scale pixels, a class label is simulated from a local probability distribution made conditional to: (i) the coarse class fraction data, (ii) any simulated land cover classes at fine pixels previously visited along that path, and (iii) a prior structural model. Being non-iterative this downscaling procedure is fast enough to generate many (10s or 1000s) alternative solutions. Furthermore, these algorithms are flexible; they can handle different geometric patterns simply by inputting different structural models.

Two algorithms using different structural model types are proposed for the sequential simulation. The first method proposed is built on block indicator cokriging which allows evaluating the previous local probability distributions by a form of kriging; the structural model is then a series of class labels indicator variograms. The second method is based on the multiple-point simulation algorithm SNESIM where the local probability distributions are read from a training image; the structural function is then that training image which can be seen as an analog image depicting the patterns deemed present at the fine resolution. For both algorithms a servo-system keeps track of the number of simulated classes inside each coarse fraction and ensures exact reproduction of the coarse fraction data.

The classical training image-based approach is modified to provide a more direct conditioning to coarse fractions. The fine scale patterns of the training image are pre-classified into a few partition classes based on their coarse fractions. All patterns within a partition class are recorded by a search tree; there is one tree per partition class. At each fine scale pixel along the simulation path, the coarse fraction data is retrieved first and used to select the appropriate search tree. That search tree contains the patterns relevant to that coarse fraction data.

Two case studies derived from Landsat TM imagery demonstrates the two approaches proposed. The resulting alternative downscaled class maps all honor the coarse proportion data, any fine scale data available, and exhibit the spatial patterns called for by the input structural model. When that structural model is incompatible with the sensor data the pattern reproduction is poor. Fine scale data such as water, roads and previously mapped fine scale pixels are shown to be well reproduced in the downscaled maps. The probabilistic servo-system is shown to be an effective mechanism to ensure exact coarse fraction reproduction. It has, however, the drawback of generating artifact noise especially when the structural model is a training image. Such noise can be reduced by some post-processing.

Acknowledgements

I have felt privileged to attend Stanford University, it is a wonderful place to study, research and live. The campus, the faculty and the students make this place magical, fascinating and above all motivating. The view of the dish, the Santa Cruz Range and the occasional hawks and hummingbirds from my office window will remain with me wherever I go. My stay could not have been that memorable without the following individuals.

First and foremost, I would like to acknowledge Professor André Journel. His commitment to the success of his students is simply remarkable. His mentorship and friendship has tremendously enhanced my stay at Stanford and his teaching had shaped me in many ways.

Many thanks to Professor Karen Seto who introduced me to the world of satellite remote sensing. She opened a whole new world of possibilities for me. Her infallible enthusiasm and support has been greatly appreciated.

The specific topic of this thesis was started when I visited Prof. Phaedon Kyriakidis at UCSB in the summer of 2003. That collaboration has continued much after that initial contact and he has played a very significant role in my research since then. I would also like to mention the help of Professor Peter Atkinson at the University of Southampton, UK, in improving my understanding of the super-resolution problems in satellite remote sensing.

Thanks to Prof. Jef Caers for the formative discussion on and teaching of geostatistics. I am particularly grateful to him for letting me teach his class in winter 2007. I would also like to thank the members of my PhD committee: Khalid Aziz, Paul Switzer, Karen Seto and Andre Journel. The administrative staff, Ginni Savalli, Thuy Nguyen and Elaine Andersen, have been a great help throughout my stay.

Finally, many thanks to the SCRF group, especially Sunder Krishnan, Nicolas Remy, Melton Hows, Joe Volker, Jianbing Wu, Scarlet Castro, Lisa Stright, Jenya Polyakova, Bruno Dujardin, Yongshe Liu and Ting Li for their friendship and assistance.

Special thanks to Nicolas Remy for his patience in teaching me the intricacies of SGeMS and the C++ language. He, Jianbing Wu, and I have used our time at SCRF to write a book about the software in addition to giving short courses which has been a rewarding experience.

None of this could have been possible without the support of my parents, Gaston and Francine, and brother, Frd́eric. They have always been encouraging and have taken pride in me and my studies. Finally, the most important event of this PhD is the meeting of my wife and fellow SCRF student, Amisha Maharaja. She has been a tremendous support during these years and has made my stay at Stanford a most memorable, enjoyable and lasting one.

Contents

List of Tables

List of Figures

Chapter 1

Downscaling categorical attributes

1.1 Introduction

Remote sensing techniques, such as satellite sensors or geophysical surveys, are an indirect but important source of information about surface and subsurface properties. Different sensors have different spatial resolutions whose information contents may have to be integrated.

For example, in optical satellite remote sensing a multispectral sensor carried on a satellite measures the reflectance across some sections of the electromagnetic (EM) spectrum, known as bands. That sensor is characterized by the spectral, temporal and spatial resolutions. The spectral resolution refers to the EM bandwidth. A hyperspectral sensor with hundreds of bands has more spectral resolution than the Landsat7 sensor with only seven bands. The temporal resolution refers to the time period a satellite takes to revisit a scene. Finally, the spatial resolution refers to the spatial extent of the sensor measurements. Note that spatial and temporal resolution are somewhat linked. In general, a satellite with high spatial resolution has low temporal resolution. From a single satellite one cannot have frequent, say daily, high definition images of any region. The Landsat7 satellite provides images with 30 meters spatial resolution but will pass over a region every 16 days. If that day happens to be cloudy, the next image will be available after a month.

By measuring the intensity of selected portions of the electromagnetic field, a satellite sensor records the surface reflectance of the landscape. That landscape can

then be analyzed by correlating ground features to the spectral responses. Thus, the satellite image is an indirect measurement of the landscape as reflected from some specific physical properties of its surface. If the satellite sensor has the same resolution as the ground features, a classification algorithm can convert these raw measurements into useful information, such as land cover types or physical attributes. This conversion is typically done with a statistical algorithm trained on known data. The training data are pixels for which the true values are known. Once a conversion model is calibrated, it can be used to convert the entire satellite image into the desired attributes.

Labeling pixels assumes some sort of homogeneity inside the pixel cells, which is rarely the case. When the satellite sensors have a coarser spatial resolution than that of categories to be mapped, it leads to measurements that are a composite response from multiple classes. Thus, a pixel is no longer seen as a single category, e.g. urban or non-urban, but as mixture of categories. Spectral unmixing is the procedure of determining the fractions of each of the categories present in any coarse pixel; see, for example, Richards and Jia (1999) or Tso and Mather (2001) for a survey of unmixing methods. The aim of spectral unmixing is to invert the spectral response into a proportion of components that would yield a response equal to the satellite measurement. Spectral unmixing yields only the fractions but does not locate them within the coarse pixel. The additional task of providing a fine resolution map of class labels within each coarse pixel lies in the realm of super-resolution mapping, also termed sub-pixel mapping, or downscaling; see, for example, Atkinson (2001). This downscaling is the topic of this thesis.

Instead of going through the spectral unmixing route, an alternative is to downscale the spectral measurement itself (Pardo-Igzquiza et al., 2006) to the scale of interest. These downscaled spectral measurements are then classified with a traditional classifier. Downscaling through spectral unmixing offers one main advantage over direct downscaling of the spectral measurements. That advantage is linked to the relative ease of using a prior structural model. Downscaling requires necessarily the specification of a structural model, that is a model depicting the patterns expected to be found at the fine spatial resolution. That structural model is not contained in the coarse sensor measurements or in the coarse fractions and must be provided

to the algorithm. It is easier and more intuitive to infer the spatial distribution of well understood physical categories, such as water or urban land cover, than to come up with a prior model for the spatial distribution of satellite sensor measurements. There are plenty of ground truth measurements, maps, photographs about the fine scale spatial patterns of the categories. This is not the case for the spatial distribution of spectral measurements. Fine scale spatial distribution models can be obtained from analog or training image or from expert knowledge. Note that the specification of a prior structural model cannot be avoided no matter the downscaling algorithm used. Assuming a structural model for the spectral measurements has the same implications than assuming it for the categories to be mapped. At the very least, the latter assumption is easier to refute.

In this thesis, downscaling class labels is seen from an inverse problem perspective (Menke, 1989; Bertero, 1985; Tarantola, 2005): that of reconstructing a fine scale map of class labels from a set of coarse class fractions. The forward problem of computing coarse fractions from a fine resolution map of class labels is trivial being a simple linear averaging. The inverse problem, however, is under-determined, in that it has multiple plausible solutions: many fine resolution class maps can lead to an equally good reproduction of the available coarse fractions. To solve such an under-determined inverse problem, one needs to invoke additional prior information that will resolve the inherent ambiguity. This prior information should pertain to the fine (target) spatial resolution, so that it constrains the "space" of possible solutions to the previous inverse problem. In what follows, this fine resolution prior information will be referred to as a model of spatial structure, structural model, or structural information.

The most primitive form of spatial structure is the rather unrealistic assumption of classes randomly distributed at the fine spatial resolution. Another and opposite type of prior information stems from an assumption of maximum class autocorrelation (grouping) at the target resolution, which underpins several approaches for super-resolution mapping (Atkinson et al., 1997; Verhoeye, 2002; Atkinson, 2001; Tatem et al., 2001; Mertens et al., 2003). This prior structural model might be appropriate when the spatial extent of patterns is much larger than that of the coarse pixel, a scenario termed H-resolution by Jupp et al. (1988). Such a model, however,

is too rigid, in that it cannot be adapted to generic cases such as a fragmented landscape where some patterns or objects are smaller than the spatial resolution of the satellite sensor.

Prior structural information can also be specified as parametric indicator variogram models (Atkinson, 2001; Makido and Shortridge, 2005; Tatem et al., 2002, 2003; Atkinson, 2004). Prior spatial information has also been specified in the form of interactions between predefined groups of pixels (cliques) linked to the parametric energy function of a Markov Random Field model (Tso and Mather, 2001; Kasetkasem et al., 2005).

In this work, the coarse resolution fractions are assumed known; how such fractions are derived is not addressed in this thesis. The common situation where the analyst has access to a typically small set of known class labels (hard data) at the fine scale is also addressed. These known class labels could have be obtained via ground surveys or GIS database. Such sparse fine resolution data provide information on the actual location of some classes at the target resolution, and hence should be reproduced exactly in the final maps.

Once a prior model of spatial structure has been specified, a probabilistic formulation of the inverse problems seeks to determine at any unsampled location the conditional probability distribution of the unknown class label given the available data. That probability distribution encapsulates the uncertainty about the unknown attribute values given the current level of information (Tarantola, 2005). A probabilistic approach is chosen because it allows to model and assess the inherent indetermination of the downscaling process.

In the context of downscaling categorical attributes, this amounts to determining the multivariate and multi-pixel conditional probability of obtaining any particular spatial combination of class labels at the target (finer) resolution, given the abundant coarse fractions and possibly some class labels sparsely sampled at the target resolution. That conditional distribution is typically complex with multiple modes and may not be analytically tractable. Instead of retrieving a single summary measure from the conditional distribution, such as its mean or mode, that distribution is "explored" by generating multiple samples or realizations from it (Mosegaard and Tarantola, 1995; Sambridge and Mosegaard, 2002; Kaipio and Somersalo, 2004; Tarantola, 2005). This

amounts to generating alternative simulated super-resolution maps of class labels that are consistent with all the information available; that is, the prior structural model, the coarse fractions, and the fine class labels if available. These simulated super-resolution maps can then be used to determine the likelihood of occurrence of patterns of classes over various groups or templates of pixels, by calculating the frequency of occurrence of such class patterns over the realizations.

More importantly, by using these super-resolution maps as input to an engineering or physical process simulation model, e.g., in a wildfire propagation simulator or a flow simulator, one can build a probability distribution for the model outputs in a Monte Carlo framework (Bachmann and Allgöwer, 2002). By fixing other input variables to a nominal set of values, one could also assess via Monte Carlo simulation the sensitivity of that engineering or physical process to the uncertainty of the fine scale map; see Crosetto et al. (2001) for a discussion of uncertainty and sensitivity analysis techniques in a remote sensing context. See also Atkinson and Graham (2006) for sub-pixel mapping in global remote sensing of disease.

To some, generating several, instead of a "best", single, downscaled class label maps may appear as a drawback. Generating a single best map requires definition of what "best" means with regard to the constituents of that map. Often, "best" is cast in terms of per-pixel accuracy with respect to the true class label map or to some validation samples obtained from it. Alternatively, one might be concerned with the reproduction of spatial patterns of land cover, such as the fragmentation or compactness (textural) characteristics of the mapped classes. In addition and most importantly, a single super-resolution map does not reveal the uncertainty associated with its construction. Most of that uncertainty stems from the lack of abundant data at the target fine spatial resolution and the difficulty to infer a model of spatial structure at that target resolution. Sub-pixel mapping is almost never the end-goal per se; instead, the maps are intended to serve as inputs to detailed spatial analysis operations, coupled environmental models, engineering and decision support systems, all requiring input data at the fine resolution. It is therefore essential to explore the consequences of the uncertainty associated with downscaling on the outcomes of the final decisions or model outputs; this calls for an uncertainty propagation or uncertainty analysis. For example, in forest fire management, one can explore the impact

of different land cover patterns on fire propagation. This could be achieved by running the fire-spread simulator using multiple alternative super-resolution realizations of land cover, and studying, say, the spatial distribution of the resulting simulated fire fronts.

In satellite remote sensing application, one of the earliest approaches to downscaling land covers is that of Verhoeye (2002), who proposed a deterministic solution based on linear programming. This approach, however, does not acknowledge the existence of multiple super-resolution maps due to multiple optima in the linear programming formulation. In general, most existing algorithms for super-resolution land cover mapping are iterative in nature, and (rightfully so) yield different maps depending on the initial map used in the iteration procedure. Mertens et al. (2003) adopted the same objective function as Verhoeye (2002), but used a genetic algorithm to search for plausible super-resolution maps. Tatem et al. (2001) trained a Hopfield neural network to iteratively optimize the initial super-resolution map with the simultaneous objectives of coarse fraction reproduction and spatial auto-correlation maximization; that method was successfully tested on an actual case study (Tatem et al., 2003). Mertens et al. (2004) used wavelets to account for the resolution difference between fine class labels and coarse fractions. At the fine scale, a neural network was trained to estimate the wavelet coefficients, from which a super-resolution map was reconstructed. Tatem et al. (2002) extended their neural network approach to account for indicator variogram models.

Atkinson (2001) and Makido and Shortridge (2005) adopted a swapping algorithm, as used in spatial simulated annealing, to construct plausible super-resolution maps. In these latter works, the coarse fractions were matched exactly by construction and preserved by the swapping. This was achieved by applying the swapping algorithm to an initial purely random super-resolution map but displaying the correct class fractions within each coarse pixel. Atkinson (2004) generated class labels that reproduce bivariate transition probabilities by swapping fine spatial resolution class labels. That method was applied to mapping Malaysian shoreline (Muslim et al., 2006).

The common concern with the above iterative approaches is their slowness and unknown rate of convergence, a heavy computational burden associated with the

repetitive evaluation of mismatch between simulated and observed coarse fractions, and most importantly between simulated and expected spatial structure. Other more complex sampling methods, such as Markov Chain Monte Carlo methods and simulated annealing, are also iterative, and can become computationally prohibitive due to slow convergence; see, for example, Kaipio and Somersalo (2004). In addition, none of the existing approaches for super-resolution mapping can account for fine resolution data in the form of a sparse set of class labels at informed fine pixels. The recently developed probability perturbation method of Caers and Hoffman (2006), although still iterative, appears to be less computational expensive, and warrants further attention in the context of super-resolution mapping.

1.2 Notation

At this point the basic notation used throughout the thesis is introduced. Let $c(v)$ denote the, usually unknown, class at a generic fine resolution pixel $v = v(\mathbf{u})$, with $\mathbf{u}$ being the coordinate vector of its centroid; the area of that pixel v is denoted as $|v|$. It is assumed that at this fine resolution $c(v)$ can take one of K mutually exclusive and collectively exhaustive labels, i.e., $c(v) = k$, with $k = 1, \ldots, K$. The set of all true class labels constitutes the unavailable super-resolution image, and can be arranged in a $(M \times 1)$ vector $\mathbf{c} = [c(v_m), m = 1, \ldots, M]^T$, where superscript T denotes transposition, $M = M_x M_y M_z$ denotes the number of fine resolution pixels, and $v_m = v(\mathbf{u}_m)$. The presence or absence of the k-th class label at pixel v can be coded by a binary class indicator $i_k(v)$:

$$i_k(v) = \begin{cases} 1 \text{ if } \ c(v) = k \\ 0 \text{ otherwise} \end{cases} \tag{1.1}$$

The set of all true indicators for the k-th class can be arranged in a $(M \times 1)$ vector $\mathbf{i}_k = [i_k(v_m), m = 1, \ldots, M]^T$; there are K such vectors (binary images), one per class.

Let $a_k(V)$ denote the fraction of the k-th class at a generic coarse resolution pixel $V = V(\mathbf{s})$, with $\mathbf{s}$ being the coordinate vector of its centroid. The set of all fraction values for the k-th class constitutes the available coarse resolution fraction

image, and can be arranged in a $(N \times 1)$ vector $\mathbf{a}_k = [a_k(V_n), n = 1, \ldots, N]^T$, with $N = N_x N_y N_z$ denoting the number of coarse pixels, and $V_n = V(\mathbf{s}_n)$; there are K such fraction vectors (images), one per class. Denote as $F = |V|/|v| = M/N >> 1$ the resolution ratio between the coarse and fine resolution images, and assume that both images are co-registered in such a way that there are always F fine resolution pixels $\{v_m, m = 1, \ldots, F\}$ in any coarse resolution pixel V. The full notation v_m^n indicates that the fine pixel v_m belongs to the coarse pixel V_n.

In addition to the coarse resolution fraction images, consider the case whereby the analyst has access to class labels obtained at a subset of $G << M$ fine resolution pixels. Denote as $j_k(v_g)$ the sampled k-th class indicator at an informed fine pixel $v_g = v(\mathbf{u}_g)$; again, $j_k(v_g) = 1$, if $c(v_g) = k$, zero if not. The G indicators for the k-th class are arranged into a $(G \times 1)$ vector $\mathbf{j}_k = [j_k(v_g), g = 1, \ldots, G]^T$; there are K such vectors, one per class. It should be stressed that these latter fine resolution data are not required for downscaling; if such data are available, however, they should readily be incorporated and honored.

In order to condense notation for subsequent discussion, denote as $\mathbf{d}_k = [\mathbf{a}_k^T \ \mathbf{j}_k^T]^T$ the $((N + G) \times 1)$ vector containing both coarse resolution fraction data and fine resolution sample indicators for the k-th class. There are K such vectors, one per class, which can be arranged in a single $((N+G) \times K)$ matrix $\mathbf{d} = [\mathbf{d}_k, k = 1, \ldots, K]$. With this concise notation, the downscaling mapping objective can be formulated as the task of finding the $(M \times 1)$ vector $\mathbf{c}$ from the $((N + G) \times K)$ matrix $\mathbf{d}$. Note again, that when the fine resolution sample indicators are not available, matrix $\mathbf{d}$ is of dimension $(N \times K)$ consisting only of class fraction values.

Turning attention to the coarse resolution data, the fraction value $a_k(V_n)$ for the k-th class at the n-th coarse pixel V_n is formally defined as the average of the k-th class indicators at the F fine pixels within that coarse pixel V_n:

$$a_k(V_n) = \frac{1}{F} \sum_{m=1}^{F} i_k(v_m), \quad v_m \in V_n \tag{1.2}$$

The spatial distribution of the k-th class fractions is again partially characterized by their mean (expected value) and a measure of their spatial structure. More

precisely, the mean $\bar{a}_k$ of the k-th class fraction values equals the corresponding proportion π_k of same class indicators:

$$\bar{a}_k = \frac{1}{N}\sum_{n=1}^{N} a_k(V_n) = \frac{1}{N}\sum_{n=1}^{N}\left[\frac{1}{F}\sum_{m=1}^{F} i_k(v_m)\right] = \frac{1}{M}\sum_{m=1}^{M} i_k(v_m) = \pi_k \tag{1.3}$$

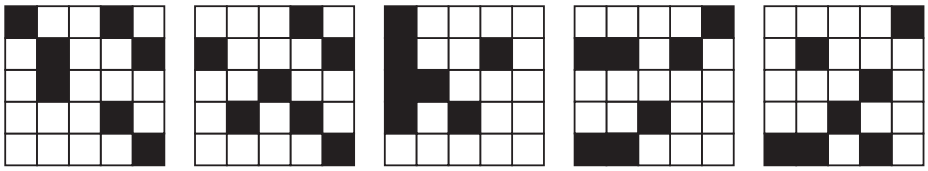
1.3 Proportion versus probability

Note that the coarse fraction $a_k(V_n)$ is the proportion of fine spatial resolution indicator $i_k(v_m)$ that are contained within location V_n. It should not be confused with a probability which refers to an event. For example, assume that location V has been measured with value p_k for class k. Consider the two following interpretations for p_k : (a) p_k is a proportion $a_k(V) = p_k$ and (b) p_k is a probability $\Pr\{I(V) = 1\} = p_k$. The latter probabilistic interpretation refers to the likelihood of a specific event $I(V) = 1$ to occur, that is that the coarse pixel V has label k, i.e. all sub-pixels are also of label k. On the other hand, considering p_k as a proportion automatically refers to a smaller support, say $|v| < |V|$, where the events occur. The proportion p_k fixes the quantity of succesful events i.e. the number of 1's occuring at the finer resolution

$$\frac{|v|}{|V|\sum_{n=1}^{N}(i(v_n))} = p_k \tag{1.4}$$

There cannot be a proportion without events occuring at smaller support. Hence the notion of proportion cannot be discussed without reference to the support of the measure itself and to the support of the underlying event. Figure 1.1 explain the differences between $a_k(V_n)$ being interpreted as a proportion instead of a probability with a series of five simulations.

Morevover, a proportion need not relate to a probabilistic model, the proportion is an (indirect) observation. On the other hand a probability always refers to a probabilistic model, a binomial model for instance. One cannot measure a probability as it is not a physical measurement but a conceptual construction.



(a) Proportion

(b) Probability

Figure 1.1: Five simulations of a coarse pixel V with a measure observation p_k. In (a) p_k is interpreted as a proportion, hence each sub-pixel simulation has the same number of black and white pixels, hence the same proportion. In (b) p_k in interpreted as a probability, if it is a binomal variable it can take either 1 or 0 value with probability p_k.

1.4 Thesis Outline

This thesis describes two methods for downscaling categorical attributes with examples drawn from satellite imagery. In Chapter 2, the sequential simulation framework in conjunction with a servo-system is presented as a general method for downscaling. To this purpose a structural model and an algorithm to compute the local conditional mass probability function (CPMF) needs to be specified. In Chapter 3 structural model is defined through variograms and the CPMF is obtained with block indicator coKriging. In Chapter 4 the richer training image concept is used for structural model and the CPMF is computed based on the single normal equation formalism. Chapter 5 gives guidelines on which structural model to choose, variogram versus training image, as function of the complexity of the spatial distribution of the class labels at the target resolution. Finally, Chapter 6 provides an in-depth discussion for each method and a critical comparison of their performances.

Chapter 2

Downscaling with sequential simulation

This chapter develops the generic sequential simulation formalism to generate fine scale simulated maps. A new servo-system to enforce the reproduction of the coarse fraction within that formalism is also presented.

2.1 Sequential Simulation

Consider the task of mapping a set of M fine resolution class labels $\{c(v_m), m = 1, \ldots, M\}$ from a set of N coarse fractions $\{\mathbf{a}(V_n), n = 1, \ldots, N\}$ with $M >> N$. Here, $c(v_m)$ denotes the class label at the m-th fine pixel v_m. This label can take one of the K mutually exclusive and collectively exhaustive states $k = 1, \ldots, K$;
$\mathbf{a}(V_n) = [a_k(V_n), k = 1, \ldots, K]^T$ denotes the $[K \times 1]$ coarse fraction data vector at the n-th coarse pixel V_n. From a stochastic inverse problem theory perspective the unknown fine resolution class map is modeled as a realization of a random set (RS) (Boucher and Kyriakidis, 2006). Solutions of the downscaling of class label inverse problem are then generated as stochastic realizations from that RS, constrained to match or reproduce within some tolerance the coarse fraction data. The reader can consult Hansen et al. (2006) for solutions to similar Gaussian inverse problems.

More precisely, the unknown class label at the m-th fine pixel v_m is associated with a set of k indicator values defined as $i_k(v_m) = 1$ if $c(v_m) = k$, zero otherwise.

The map of the M indicators for the k-th class can be denoted as a $[M \times 1]$ vector $\mathbf{i}_k = [i_k(v_m), m = 1, \ldots, M]^T$; there are K such vectors, one for each class. This indicator vector $\mathbf{i}_k$ is regarded as a joint realization of M indicator random variables $I_k(v_1), \ldots, I_k(v_M)$. Stochastic downscaling of class label amounts to generating simulated fine resolution maps, i.e., a set of, say, S synthetic indicator vectors, from the M-variate PMF of the M indicator random variables, conditional to the N coarse fraction data $\mathbf{a}_k = [a_k(V_n), n = 1, \ldots, N]^T$ for the same class:

$$f(\mathbf{i}_k|\mathbf{a}_k) = \Pr\{I_k(v_1), \ldots, I_k(v_M)|\mathbf{a}_k\}\,,\ k = 1, \ldots, K \tag{2.1}$$

In the general case, the M-variate conditional PMF of Equation 2.1 is analytically intractable, hence difficult to sample. A practical solution is to generate samples from that PMF via a recursive application of Bayes' rule; see, for example, Johnson (1987). This recursive application of Bayes' rule is known in geostatistics as sequential indicator simulation; it decomposes the single M-variate conditional PMF into a series of M univariate or local (pixel-specific) conditional PMFs (CPMFs) as:

$$\begin{aligned} f(\mathbf{i}_k|\mathbf{a}_k) = {} & \Pr\{I_k(v_1) = i_k(v_1)|\mathbf{a}_k\} \\ & \cdot \Pr\{I_k(v_2) = i_k(v_2)|\mathbf{a}_k, i_k(v_1)\} \\ & \cdot \Pr\{I_k(v_3) = i_k(v_3)|\mathbf{a}_k, i_k(v_2), i_k(v_1)\} \\ & \cdots \\ & \cdot \Pr\{I_k(v_M) = i_k(v_M)|\mathbf{a}_k, \\ & \qquad i_k(v_{M-1}), i_k(v_{M-2}), \ldots, i_k(v_1)\} \end{aligned} \tag{2.2}$$

Simulation from the conditional PMF $f(\mathbf{i}_k|\mathbf{a}_k)$ is greatly simplified, since one needs to sequentially generate M class labels from the M local CPMFs, instead of generating the M labels all at once from $f(\mathbf{i}_k|\mathbf{a}_k)$. In sequential simulation, there is no restriction on the decomposition sequence of Equation 2.2. That sequence is called the path of the simulation, and is typically random. Previously simulated class labels, along with the original coarse fractions are used as conditioning data for determining the local CPMFs at any unvisited fine pixel along the random path. For more details regarding sequential simulation, the reader is referred to Deutsch and Journel (1998).

As simulation progresses along the random path, conditioning involves more data, i.e., more previously simulated class labels, hence becomes more difficult. This problem is circumvented by invoking a screening approximation: at the fine pixel v_m only relevant previously simulated class labels are considered as conditioning data for determining the local CPMF. Relevant here implies that such labels correspond to fine pixels that fall within a pre-defined neighborhood or window $W(v_m)$ around pixel v_m. More precisely, let $\Omega_{(m-1)}$ denote the set of simulated class labels prior to the m-th step of the sequential simulation decomposition of Equation 2.2, and let $\Omega_{(0)}$ denote the initial set of conditioning data, i.e., the coarse fraction vector $\mathbf{a}_k$ for each class. If one defines as $\Omega_{(m-1)}^{W(v_m)} \in \Omega_{(m-1)}$ the subset of already simulated labels at pixels lying within neighborhood $W(v_m)$, then Equation 2.2 is re-written as:

$$f(\mathbf{i}_k|\Omega_{(0)}) = \prod_{m=1}^{M} \Pr\left\{I_k(v_m) = 1|\Omega_{(m-1)}^{W(v_m)}\right\},\ k = 1, \ldots, K \tag{2.3}$$

Typically, the neighborhood $W(v_m)$ consists of some template geometry, e.g. an ellipse, defined by a group of N_W coarse pixels with $N_W << N$. A user-defined number controls the maximum number of previously simulated labels falling within $W(v_m)$ that are retained, along with the fraction data at the N_W coarse pixels, to determine the local CPMF $\Pr\left\{I_k(v_m) = 1|\Omega_{(m-1)}^{W(v_m)}\right\}$ for each class at the current fine pixel v_m. A simulated class label is then generated from the K local CPMFs at pixel v_m, and this class label is used for conditioning at subsequent fine pixels visited along the simulation path. Another realization is generated by changing the simulation path, i.e. the order of visiting the M fine pixels, and by using a different set of random numbers to draw from the local CPMFs.

Note that sequential simulation for downscaling class label can readily accommodate prior information at the fine resolution, by including it in the initial conditioning data set $\Omega_{(0)}$. In summary, the sequential simulation algorithm for sampling from the multivariate conditional PMF of Equation 2.1 to generate simulated fine resolution class label maps proceeds as:

1. Define the initial conditioning data $\Omega_{(0)} = \mathbf{a}_k, k = 1, \ldots, K$

2. Choose a path for visiting the M fine pixels

3. At the current fine pixel v_m:

   (a) Get the neighboring conditioning data set $\Omega_{(m-1)}^{W(v_m)}$

   (b) Estimate $\Pr\left\{I_k(v_1) = 1|\Omega_{(m-1)}^{W(v_m)}\right\}$, i.e., the local CPMF for each class

   (c) Draw a simulated label $c^{(s)}(v_m)$ from the above K local CPMFs

   (d) Add $c^{(s)}(v_m)$ to the set of conditioning data $\Omega_{(m)} = \Omega_{(m-1)} \cup c^{(s)}(v_m)$

4. Repeat steps 2-3 to generate another fine resolution realization

In most cases the CPMF $\Pr\left\{I_k(v_1) = 1|\Omega_{(m-1)}^{W(v_m)}\right\}$ by itself does not guarantee the coarse fractions reproduction on each simulated realization, a correction must be applied through a servo-system.

2.2 Servo-system

Given that sequential simulation can only reproduce coarse fraction in expected value over a large number of realizations, a progressive correction algorithm is proposed to ensure such reproduction on each realization. The idea is to code into a probability the likelihood of drawing a specific class given the coarse fraction constraint. At any point along the random path, the coarse proportion constraint is transformed into a probability of reaching that constraint. Consider the m-th fine pixel v_m^n visited along the simulation path; superscript n indicates that this pixel lies within the coarse pixel V_n. Let $p_k^R(v_m^n)$ denote the current (prior to visiting the m-th pixel v_m) remaining proportion of fine pixels within V_n that, given the coarse fraction $a_k(V_n)$, should be simulated as belonging to the k-th class:

$$p_k^R(v_m^n) = \zeta(int[a_k(V_n)F] - \sum_{m=1}^{F_{m-1}(V_n)} i_k(v_m^n)) \tag{2.4}$$

where $int[\cdot]$ denotes rounding to the nearest integer, $F_{m-1}(V_n)$ denotes the number of fine pixels within V_n that have already been visited prior to arriving at the m-th pixel v_m^n.

To reproduce the coarse proportion exactly, $\zeta(\cdot)$ is set as :

$$p_k^R(v_m^n) = \frac{int[a_k(V_n)F] - \sum_{m=1}^{F_{m-1}(V_n)} i_k(v_m^n)}{F - F_{m-1}(V_n)}, \quad \in [0,\ a_k(V_n)] \tag{2.5}$$

The running probability $p_k^R(v_m^n)$ can be regarded as the probability of simulating a fine pixel as belonging to the k-th class, given: (i) the number of fine pixels within V_n already simulated as belonging to that class, and (ii) the target coarse fraction value $a_k(V_n)$. This running probability can thus be viewed as a saturation probability for a particular coarse pixel v_m in the k-th class; itdepends on how many fine pixels have already been simulated as belonging to class k within that coarse pixel.

The servo-system combines the the conditional probability derived from the structural information denoted here as $p_k^S(v_m^n) = \Pr\left\{I_k(v_1) = 1 | \Omega_{(m-1)}^{W(v_m)}\right\}$ with the running probability $p_k^R(v_m^n)$ of Equation (2.5) using the tau model (Journel, 2002) :

$$p_k^F(v) = \left[1 + \left(\frac{1 - p_k^S(v_m)}{p_k^S(v_m)}\right)^{\tau_S} \left(\frac{1 - p_k^R(v_m)}{p_k^R(v_m)}\right)^{\tau_R} \cdot \left(\frac{1 - \pi_k}{\pi_k}\right)^{1-\tau_S-\tau_R}\right]^{-1} \tag{2.6}$$

the reader is also referred to Bordley (1982); Benediktsson and Swain (1992) for a different perspective on that model. Consider the following scenarios :

- When no fine pixel within V_n has been simulated as belonging to class k, $p_k^R(v_m^n) = a_k(V_n)$, and Equation 2.6 leaves the original $p_k^S(v_m^n)$ unchanged.

- As simulation progresses along the random path, the correction of Equation 2.6 may become more important if $p_k^R(v_m^n)$ differs significantly from the target proportion $a_k(V_n)$.

- When the proportion of fine pixels within V_n simulated as belonging to class k reaches the target fraction $a_k(V_n)$, then $p_k^R(v_m^n) = 0$, and the combined probability of Equation 2.6 is also zero.

- The corollary is that when only category k is left to be simulated, $p_k^R(v_m^n) = 1$ and $p_{k'}^R(v_m^n) = 0, \forall k' \neq k$, hence $p_k^F(v) = 1$

This algorithm therefore constrains the proportion of simulated fine resolution class labels within V_n to reproduce exactly the corresponding coarse fraction $a_k(V_n)$.

Note that this servo system ensures coarse fraction reproduction no matter the exponent τ_R with the exception of $\tau_R = 0$ That latter zero value amounts to ignore any requirement for exact coarse fraction reproduction.

The severity of the updating of a data event-derived probability $p_k^S(v_m^n)$ by the servo-system can be controled either through the probabilistiv mismatch function in Equation 2.4 or through the tau-exponents (Equation 2.6). For instance the intensity of correction can be reduced by generating a mismatch-derived probability $p_k^R(v_m)$ close to the marginal π_k or decrease the tau-exponent τ_R below one. Modyfing the probability changes the information content provided by the constraints while changes in the tau-exponent model an interaction, (redundancy or compounding) between sources of information. Nonetheless, the tau-exponent should not be used as substitute for a poor mismatch function.

The proposed probabilistic servo-system is novel and and based on the coding of the mismatch between the target and the simulated proportions into a probability. The enforcement of the constraint is then left to the merging of probabilities with the rigorous tau-model. Previous servo-systems simply used the mismatch as a ad-hoc parameter to modify the data event-derived probability, for example Strebelle (2002) scale that probability with an exponential function itself scaled by the mismatch values while Soares (1998) add an additive term to the data event-derived probability ($p_k^S(v_m)$) to modify it in the right direction. Furthermore these servo-systems could ensure exact reproduction of the constraint, a necessity for downscaling applications.

Chapter 3

Downscaling with variogram

This chapter describes the novel approach proposed in Boucher and Kyriakidis (2006, 2007) for super-resolution land cover mapping based on the geostatistical methods of indicator kriging (Journel, 1983) and sequential indicator stochastic simulation (Journel and Alabert, 1989), accounting explicitly for the resolution difference between the available coarse fractions and the sought-after fine scale class labels. Indicator variograms and block indicator cokriging are used to approximate the probability that a pixel at the fine spatial resolution belongs to a particular class, given the coarse resolution fractions and (if available) a sparse set of class labels at some informed fine pixels. Such kriging-derived probabilities are used in sequential indicator simulation to generate synthetic maps of class labels at the fine resolution pixels, see Section 2.1. The proposed approach: (i) is non-iterative and computationally inexpensive, (ii) offers exact (within round-off errors) reproduction of coarse resolution fractions, (iii) ensures exact reproduction of observed class labels at informed fine resolution pixels that might be available, and (iv) closely reproduces a set of indicator variogram models linked to transition probabilities of class labels from one fine pixel to another. Since there are multiple solutions to downscaling class labels, the end product is a set of alternative realizations or maps of class labels having the properties listed above.

Section 3.1 derives, via indicator cokriging, the conditional probability of class occurrence at any fine resolution pixel, given the neighboring coarse fractions and possibly some fine resolution sample class labels. The cokriging formulation is extended within a modified sequential indicator simulation framework to generate alternative

realizations of fine resolution class labels. Section 3.4 illustrates the applicability of the proposed methodology for downscaling land cover mapping using data from two Landsat TM-derived scenes over China.

3.1 Variograms as structural model

The spatial distribution of the k-th class at the fine resolution is partially characterized by: (i) its proportion, and (ii) a measure of its spatial structure or texture. Denote as π_k the stationary proportion of the k-th class, and as $2\gamma_k^v(\mathbf{h}) = 2\gamma_k^v(\mathbf{u} - \mathbf{u}')$ the stationary variogram of the k-th class indicators between any two generic (informed or not) fine resolution pixels $v(\mathbf{u})$ and $v(\mathbf{u}')$, whose respective centroids $\mathbf{u}$ and $\mathbf{u}'$ are separated by vector $\mathbf{h} = \mathbf{u} - \mathbf{u}'$; superscript v denotes that the indicator variogram $2\gamma_k^v(\mathbf{h})$ pertains to the fine spatial resolution of v-pixels. The indicator variogram between two particular informed fine pixels v_g and $v_{g'}$ is then denoted as $2\gamma_k^v(\mathbf{u}_g - \mathbf{u}_{g'}) = 2\gamma_k^v(\mathbf{h}_{gg'})$, whereas the indicator variogram between a particular uninformed fine pixel v_m and a particular informed fine pixel v_g is denoted as $2\gamma_k^v(\mathbf{u}_m - \mathbf{u}_g) = 2\gamma_k^v(\mathbf{h}_{mg})$.

The indicator variogram $2\gamma_k^v(\mathbf{h})$ relates to the joint probability of any two generic fine pixels separated by vector $\mathbf{h}$ to have different class labels. That indicator variogram is readily linked to the probability of transition of class k from any pixel v to a different class k' at any other pixel v' when the two pixel centroids are separated by vector $\mathbf{h}$; see, for example, Carle and Fogg (1996). The joint spatial dissimilarity of two different classes $k \neq k'$ can be characterized by their indicator cross-variogram $2\gamma_{kk'}^v(\mathbf{h})$; although possible, it is not considered here.

In geostatistical practice, it is customary to work with parametric semivariogram models, e.g., spherical or exponential for the case of indicator semivariograms. Such parametric models allow the computation of semivariogram values for arbitrary lag vectors $\mathbf{h}$, but come at the expense of an often cumbersome modeling phase. Difficult inference (parameter estimation) is particularly true for the case of jointly modeling a set of K indicator semivariogram models, since one has to satisfy certain conditions to obtain a permissible set of K such models (Journel and Posa, 1990; Goovaerts, 1997). An alternative to parametric variogram is to calculate K experimental variogram

maps from analog images. Both parametric variograms and variogram maps are discussed next.

3.1.1 Parametric variograms

In a parametric setting, the fine spatial resolution structure for each class k is defined with a parametric model $2\gamma_k^v(\mathbf{h};\boldsymbol{\theta})$. Here $\boldsymbol{\theta}$ denotes a vector of model parameters, such as sill, range, nugget contribution, and possibly anisotropy direction and ratio (Goovaerts, 1997). The inference of such a variogram model could be based on fine resolution information, such as aerial photographs acquired in similar scenes and/or possibly ground-based data. Alternatively, one could have information on the functional form of such a variogram model, say, exponential with a nugget component, and then iteratively fit this model to the coarse fractions, along the lines used by Journel and Huijbregts (1978) and Atkinson and Curran (1995). It should be noted here that ground-based surveys rarely yield dense sampling over the study area; they only provide partial information on the occurrence of class labels at a sparse set of fine pixels. In most real-world situations, the sampling density of these ground-based surveys would not be sufficient to allow quantifying fine resolution spatial variability through variograms.

If such fine resolution sample data are not available, then the prior structural model might be synthesized from analogs, e.g. high resolution imagery obtained from a different sensor in a nearby region with similar classes, or in the same region but in the past. Alternatively, that prior structural information might be built by combining the spatial information from the coarse resolution fractions with some expert opinion on the fine resolution spatial structure. For example, one might deem some classes less continuous or with more high frequency variation than others. Such prior structural information could be integrated in the fine resolution indicator variogram models by adjusting their relative nugget contributions and/or ranges. Lastly, one could imagine using a synthetic map, as obtained from a forward modeling algorithm, as a possible source of fine resolution textural information.

3.1.2 Variograms tables

When an analog class image is available and all data are located at the nodes of a regular grid, one can circumvent the cumbersome modeling phase and pre-compute all indicator semivariogram values directly from that analog image. To do so, the analog image should: (i) have the same spatial resolution as the target grid, (ii) contain all K classes in approximately the correct proportions, and (iii) be at least twice as big as the largest separation vector for which indicator semivariogram values are required. From that analog image, the indicator semivariogram between pixel v_m with centroid coordinates $\{x_m, y_m\}$ and a pixel $v_{m'}$ with centroid coordinates $\{x_{m'}, y_{m'}\}$ can be readily obtained as $\gamma_k(x_m - x_{m'}, y_m - y_{m'})$. The limitation is that only indicator semivariogram values for separation vectors defined between pixels located at the nodes of the analog image grid can be calculated.

All necessary indicator semivariogram values can be obtained by scanning the K binary indicator images corresponding to the original analog class image. More precisely, one needs to compute (for each class) the experimental semivariogram values for all pairs of pixels whose corresponding separation vectors have components $\{h_x = x - x', h_y = y - y'\}$ along the x- and y-directions, respectively. An experimental indicator semivariogram raster is then constructed by arranging semivariogram values at positions $\{(m, m'), m = 0, \ldots, M_x - 1, m' = 0, \ldots, M_y - 1\}$, where M_x and M_y denote the number of nodes in the semivariogram table along the x-and y-directions, respectively. Position $(h_x = 0, h_y = 0)$ occupies the center of the indicator semivariogram map; at that central location, $\gamma_k(0, 0) = 0$, per the definition of the semivariogram. The scanning and computation of semivariogram values from the k-th binary analog image can be done very efficiently by using the Fast Fourier Transform (FFT); see Marcotte (1996) and Yao and Journel (1998) for more details.

Semivariogram values extracted from the experimental semivariogram map, however, may not always yield unique solutions to kriging systems, unless the variogram values are computed from an image eroded by the largest variogram lag being considered (Journel, 1997). In most cases, further processing is required to render that map useful for geostatistical applications. The solution lies in Bochner's theorem (Chilès and Delfiner, 1999), which states that a covariance map is positive-definite

if its Fourier coefficients are positive. The usual procedure is then to: (i) compute the forward FFT of that covariance map, i.e., compute its Fourier coefficients, (ii) correct for positive-definiteness by setting any such negative coefficients to some very small positive number, and (iii) back-transform the corrected Fourier coefficients map into a corrected covariance map using the inverse FFT (Marcotte, 1996; Yao and Journel, 1998). For indicator semivariogram maps zero Fourier coefficients are allowed, since semivariogram maps must be conditionally negative semi-definite (Chilès and Delfiner, 1999).

3.1.3 Variogram Regularization

The fine-to-fine resolution variogram only refers to the autocorrelation at the fine scale. In downscaling problem most of the data are defined at a large support as proportion data. Given that the relationship between indicators and proportion is linear, see Equation 1.2, the variogram at fine spatial support can be upscaled to account for the spatial autocorrelation across support (Journel and Huijbregts, 1978). The fine-to-coarse resolution variogram between the k-th class indicator $i_k(v_m)$ at the m-th fine pixel v_m and the k-th class fraction $a_k(V_n)$ at the n-th coarse pixel V_n is then computed from the fine resolution indicator variogram model $2\gamma_k^v(\mathbf{h};\boldsymbol{\theta})$ as:

$$2\gamma_k(v_m, V_n) = \frac{1}{F}\sum_{m'=1}^{F} 2\gamma_k^v(\mathbf{h}_{mm'};\boldsymbol{\theta}), \quad v_{m'} \in V_n \tag{3.1}$$

where the fine pixel v_m need not lie within the coarse pixel V_n.

The above fine-to-coarse resolution variogram is stationary because: (i) the fine resolution variogram $2\gamma_k^v(\mathbf{h};\boldsymbol{\theta})$ is stationary, and (ii) the coarse pixel size defining the integration domain is constant. This entails that the fine-to-coarse resolution variogram is also a function of the separation vector $\mathbf{h}_{mn} = \mathbf{u}_m - \mathbf{s}_n$ between the centroids $\mathbf{u}_m$ and $\mathbf{s}_n$ of pixels v_m and V_n, i.e., $2\gamma_k(v_m, V_n) = 2\gamma_k(v(\mathbf{u}_m), V(\mathbf{s}_n)) = 2\gamma_k^{vV}(\mathbf{u}_m - \mathbf{s}_n) = 2\gamma_k^{vV}(\mathbf{h}_{mn})$; superscript vV indicates that $2\gamma_k^{vV}(\mathbf{h}_{mn})$ is a cross-resolution variogram between fine v-pixels and coarse V-pixels. The cross-resolution variogram between the g-th informed fine pixel v_g and the n-th coarse pixel V_n is computed in the same way as in Equation (3.1), and is denoted as $2\gamma_k^{vV}(\mathbf{h}_{gn})$.

Last, the variogram between the k-th class fractions at two particular coarse pixels V_n and $V_{n'}$ can be computed from the fine resolution indicator variogram model $2\gamma_k^v(\mathbf{h};\boldsymbol{\theta})$ as:

$$2\gamma_k(V_n, V_{n'}) = \frac{1}{F^2}\sum_{m=1}^{F}\sum_{m'=1}^{F}\gamma_k^v(\mathbf{h}_{mm'};\boldsymbol{\theta}), \quad v_m \in V, \text{ and } v_{m'} \in V' \tag{3.2}$$

where, again, that coarse resolution variogram is stationary. This entails that the above coarse resolution variogram is also a function of the separation vector $\mathbf{h}_{nn'} = \mathbf{s}_n - \mathbf{s}_{n'}$ between the centroids $\mathbf{s}_n$ and $\mathbf{s}_{n'}$ of pixels V_n and $V_{n'}$, i.e., $2\gamma_k(V_n, V_{n'}) = 2\gamma_k(V(\mathbf{s}_n), V(\mathbf{s}_{n'})) = 2\gamma_k^V(\mathbf{s}_n - \mathbf{s}_{n'}) = 2\gamma_k^V(\mathbf{h}_{nn'})$; superscript V indicates that $2\gamma_k^V(\mathbf{h}_{nn'})$ is a coarse resolution variogram between coarse V-pixels.

Both Equations (3.1) and (3.2) are classically derived from the variogram definition and the functional relationship of Equation (1.2) linking fine resolution indicators with coarse resolution fractions (Journel and Huijbregts, 1978; Atkinson and Curran, 1995). In what follows, the variogram models for the fine resolution class indicators, the fine-to-coarse variograms between such class indicators and the corresponding coarse fractions, and the variograms between the coarse fractions, are used to estimate the probability of class occurrence at any fine resolution pixel.

3.2 Indicator Cokriging

Cokriging with coarse resolution fraction data $a_k(V)$ and, if available, fine resolution class indicator data yields an approximation $\hat{p}_k(v)$ to the true probability $p_k(v) = Prob\{I_k(v) = 1|\mathbf{d}_k\}$ of k-th class occurrence at any fine resolution pixel v (Journel, 1983; Goovaerts, 1997); where $I_k(v)$ denotes a binary random variable at the fine scale pixel v. That probability $\hat{p}_k(v)$ is expressed as a weighted linear combination of coarse fractions and fine indicators for the k-th class. Because all class proportions are assumed known, or estimated with sufficient confidence by the mean of the corresponding fraction values, see Equation (1.3), consider the simple Indicator Cokriging (ICK) estimate for the m-th fine pixel v_m, written as:

$$\hat{p}_k(v_m) = \boldsymbol{\eta}_k(v_m)^T\mathbf{a}_k + \boldsymbol{\lambda}_k(v_m)^T\mathbf{j}_k + \pi_k\left[1 - \boldsymbol{\eta}_k(v_m)^T\mathbf{1}_N - \boldsymbol{\lambda}_k(v_m)^T\mathbf{1}_G\right] \tag{3.3}$$

where $\boldsymbol{\eta}_k(v_m) = [\eta_n^k(v_m), n = 1, \ldots, N]^T$ denotes the $(N \times 1)$ vector of weights assigned to the N fractions for the k-th class, $\boldsymbol{\lambda}_k(v_m) = [\lambda_g^k(v_m), g = 1, \ldots, G]^T$ denotes the $(G \times 1)$ vector of weights assigned to the G informed fine pixels; $\mathbf{1}_N$ and $\mathbf{1}_G$ denote, respectively, a $(N \times 1)$ and a $(G \times 1)$ vector of ones. The known class proportion π_k receives as weight the complement to one of the sum of the weights attributed to the N class fractions and the G class indicators.

The ICK weights $\boldsymbol{\lambda}_k(v_m)$ and $\boldsymbol{\eta}_k(v_m)$ for the k-th class are obtained by solving the following system of equations (ICK system):

$$\begin{bmatrix} \boldsymbol{\Gamma}_k^{VV} & \boldsymbol{\Gamma}_k^{vV} \\ \boldsymbol{\Gamma}_k^{Vv} & \boldsymbol{\Gamma}_k^{vv} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_k(v_m) \\ \boldsymbol{\lambda}_k(v_m) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\gamma}_k^{vV} \\ \boldsymbol{\gamma}_k^{vv} \end{bmatrix} \tag{3.4}$$

where $\boldsymbol{\Gamma}_k^{VV} = [2\gamma_k^V(\mathbf{h}_{nn'}), n = 1, \ldots, N, n' = 1, \ldots, N]$ is a $(N \times N)$ matrix of fraction variogram values between all pairs of coarse pixels, $\boldsymbol{\Gamma}_k^{vv} = [2\gamma_k^v(\mathbf{h}_{gg'}), g = 1, \ldots, G, g' = 1, \ldots, G]$ is a $(G \times G)$ matrix of indicator variogram values between all pairs of informed fine pixels, $\boldsymbol{\Gamma}_k^{vV} = [2\gamma_k^{vV}(\mathbf{h}_{gn}), g = 1, \ldots, G, n = 1, \ldots, N]$ is a $(G \times N)$ matrix of variogram values between all pairs of informed fine and coarse pixels, and $\boldsymbol{\Gamma}_k^{Vv} = [\boldsymbol{\Gamma}_k^{vV}]^T$. Term $\boldsymbol{\gamma}_k^{vV} = [2\gamma_k^{vV}(\mathbf{h}_{mn}), n = 1, \ldots, N]^T$ denotes a $(N \times 1)$ vector of variogram values between the m-th uninformed fine pixel v_m and the N coarse pixels, and term $\boldsymbol{\gamma}_k^{vv} = [2\gamma_k^v(\mathbf{h}_{mg}), g = 1, \ldots, G]^T$ denotes $(G \times 1)$ vector of variogram values between the m-th uninformed fine pixel v_m and the G informed fine pixels.

Equations (3.3) and (3.4) account for both the coarse resolution class fraction data and (if available) the fine resolution class indicators. Each piece of information, be it a coarse fraction $a_k(V_n)$ or a fine indicator $j_k(v_g)$, is weighted according to its relevance to the unknown same class indicator $i_k(v_m)$ at the uninformed fine pixel v_m. That relevance is quantified by the fine-to-coarse resolution variogram $2\gamma_k^{vV}(\mathbf{h}_{mn})$ for the class fraction $a_k(V_n)$, and by the fine resolution variogram $2\gamma_k^v(\mathbf{h}_{mg})$ for the class indicator $j_k(v_g)$. The ICK weights also account for the redundancy between the data found within the neighborhood of the uninformed fine pixel v_m; that is, for: (i) the correlation between class fractions $a_k(V_n)$ and $a_k(V_{n'})$ at neighboring coarse pixels, (ii) the correlation between class indicators $j_k(v_g)$ and $j_k(v_{g'})$ at neighboring informed fine pixels, and (iii) the correlation between a coarse resolution class fraction $a_k(V_n)$ and a neighboring fine resolution class indicator $j_k(v_g)$.

Since there are K classes, a set of K systems similar to that given in Equation (3.4) need to be solved, one per class k. The resulting ICK-based probabilities $\{\hat{p}_k(v_m), k = 1, \ldots, K\}$ derived from Equation (3.3) need to be corrected to ensure that they lie in the $[0, 1]$ interval and have a unit sum; typically, the magnitude of such corrections is small (Goovaerts, 1997). In addition, these K systems need to be solved M times, one for each fine pixel v_m. Because the number N of class fraction data can be large, the above systems are solved using a limited amount of such data, most often those falling in a neighborhood centered on the coarse pixel V_n containing the fine pixel v_m where ICK is performed. The extent of that neighborhood is typically linked to the range of the class fraction variograms (Goovaerts, 1997).

Since kriging is an exact interpolator, the ICK-derived probability $\hat{p}_k(v_g)$ at an informed fine pixel v_g yields back the class indicator datum value $j_k(v_g)$, no matter the indicator variogram models $2\gamma_k^v(\mathbf{h}; \boldsymbol{\theta})$ used to compute all the variogram values called for by the ICK system of Equation (3.4). Equally important for consistency, the average of the ICK-derived probabilities of class occurrence within any given coarse pixel V_n, provided they are calculated with the same neighborhood data, reproduces the corresponding class fraction:

$$\frac{1}{F}\sum_{m=1}^{F} \hat{p}_k(v_m) = a_k(V_n), \quad v_m \in V_n \tag{3.5}$$

see Mao and Journel (1998) for a proof, as well as Kyriakidis and Yoo (2005) for a generalization of that proof. The above reproduction holds only if the ICK-derived probabilities are not corrected to lie in the $[0, 1]$ interval and sum to one. If such a correction is performed, then Equation (3.5) becomes approximate, yet in practice very close to being true; see Mao and Journel (1998), as well as the case study in Section 3.4.

If no fine resolution class indicators are available ($G = 0$), the ICK estimate $\hat{p}_k(v_m)$ of Equation (3.3) simplifies to:

$$\hat{p}_k(v_m) = \boldsymbol{\eta}_k(v_m)^T \mathbf{a}_k + \pi_k \left[1 - \boldsymbol{\eta}_k(v_m)^T \mathbf{1}_N\right] \tag{3.6}$$

The corresponding ICK weights are simply found by reducing Equation (3.4) to:

$$\boldsymbol{\Gamma}_k^{VV} \boldsymbol{\eta}_k(v_m) = \boldsymbol{\gamma}_k^{vV} \tag{3.7}$$

and the reproduction of the coarse resolution class fraction by the resulting ICK probabilities is still guaranteed, i.e., Equation (3.5) still applies, no matter the fine resolution indicator variogram models $2\gamma_k^v(\mathbf{h};\boldsymbol{\theta})$ used to compute the entries of matrix $\boldsymbol{\Gamma}_k^{VV}$ and vector $\boldsymbol{\gamma}_k^{vV}$ in Equation (3.7).

The kriging component of the approach developed by Verhoeye (2002) can now be viewed within the proposed geostatistical framework. More precisely, Verhoeye estimates $\hat{p}_k(v_m)$, which was termed a spatial dependence parameter, by replacing the right hand side vector $\boldsymbol{\gamma}_k^{vV}$ in the system of Equation (3.7) by $\boldsymbol{\gamma}_k^{VV}$; this modification amounts to ignoring the resolution difference between the original coarse fractions and the target super-resolution map. As a consequence, any integration of coarse resolution fractions with fine resolution indicators would not be straightforward, as opposed to our method which readily allows for such a fusion.

On the computational side, if no fine resolution indicators are available, the weights vector $\boldsymbol{\eta}_k(v_m)$ can be found by solving much fewer ICK systems than the number M of fine resolution pixels (Kyriakidis and Yoo, 2005). This is a consequence of the fact that, for a given indicator variogram model $2\gamma_k^v(\mathbf{h};\boldsymbol{\theta})$, the ICK weights are only function of the spatial data configuration (the pixel layout), not of the actual data values. Since all fine pixels within a coarse pixel have the same coarse neighbors (ignoring edge effects) as any other set of fine pixels within another coarse pixel, that data configuration remains the same.

It is not recommended to turn the kriging-derived probabilities into a set of class labels by some form of, say, Maximum a Posteriori (MAP) allocation rule. Indeed, downscaling class label mapping is much more than per-pixel classification (Tarantola, 2005), in that it deals with *joint* classification under coarse fraction reproduction constraints; that is, with the task of creating class label maps consistent with all available information. An example of such classification is given in Section 3.4.2.

3.3 Sequential indicator simulation

One efficient simulation algorithm for generating synthetic categorical maps, given a sample set of known categories and a set of indicator variogram models, is sequential indicator simulation (Journel and Alabert, 1989; Goovaerts, 1997), abbreviated here as SIS. The key paradigm in SIS is the decomposition of the multivariate probability distribution $f(\mathbf{i}_k|\mathbf{d}_k)$ of M indicator RVs $\{I_k(v_m), m = 1, \ldots, M\}$ into a product or sequence of M univariate conditional probabilities as:

$$\begin{aligned} f(\mathbf{i}_k|\mathbf{d}_k) &= Prob\left\{I_k(v_1) = i_k(v_1), \ldots, I_k(v_M) = i_k(v_M)|\mathbf{d}_k\right\}, \; k = 1, \ldots, K \quad (3.8) \\ &= \prod_{m=1}^{M} Prob\left\{I_k(v_m) = 1|\mathbf{i}_k^{m-1}, \mathbf{d}_k\right\}, \quad k = 1, \ldots, K \end{aligned}$$

where $Prob\left\{I_k(v_m) = 1|\mathbf{i}_k^{m-1}, \mathbf{d}_k\right\}$ is the univariate conditional probability of the m-th indicator RV $I_k(v_m)$, given: (i) the $m-1$ previously simulated indicator values stored in a $((m-1) \times 1)$ vector $\mathbf{i}_k^{m-1} = [i_k(v_{m'}), m' = 1, \ldots, m-1]^T$, and (ii) the original data vector $\mathbf{d}_k$ comprised of the N coarse resolution fractions and possibly the G fine resolution sample indicators for the k-th class. In this chapter, any such univariate conditional distribution is determined by indicator cokriging (ICK); see Section 3.2.

Simulation proceeds along a, typically random, path which considers in sequence the M fine resolution pixels, and thus determines the order of the decomposition given in Equation (3.8). At any fine pixel, say v_m, along that path, a simulated class label $c^{(l)}(v_m)$ is generated from the corresponding K ICK-derived conditional probabilities at that pixel; here superscript (l) denotes the l-th simulated value, and there can be L such values. The associated K simulated indicators $\{i_k^{(l)}(v_m), k = 1, \ldots, K\}$ are then considered as data which constrain all subsequent univariate conditional probabilities, and hence all subsequent simulated class labels generated from such probabilities.

The end product of repeating the above steps at all M fine resolution pixels is a simulated realization of class labels denoted as $\mathbf{c}^{(l)} = [c^{(l)}(v_m), m = 1, \ldots, M]^T$. A new class realization $\mathbf{c}^{(l')}$ can be generated by repeating the above procedure with a different random path. Because any simulated class label conditions or constrains the generation of all subsequent labels along the simulation path, spatial continuity

is ensured. More precisely, the simulated class maps $\{\mathbf{c}^{(l)}, l = 1, \ldots, L\}$ reproduce in expected value the K fine resolution indicator variogram models $\{2\gamma_k^v(\mathbf{h}; \boldsymbol{\theta}), k = 1, \ldots, K\}$; for more details, the reader is referred to Goovaerts (1997).

This approach shares the same goal as the two-point histogram approach of Atkinson (2004). Instead of generating simulated class labels that reproduce these variogram maps by swapping fine spatial resolution class labels with an explicit objective function, this goal is achieved in this work directly and without iteration with the ICK system and the sequential simulation formalism.

3.4 Case Studies

The downscaling of class labels with ICK is demonstrated with two examples derived from Landsat satellites images. The first example involves three categories with parametric variogram models; a sensitivity analysis is performed on the spatial resolution of the coarse data. The second example has two categories, the texture of which is modeled by experimental covariance tables generated from an analog image. Sensitivity to availability of fine spatial resolution data is studied.

The reference map considered in these case studies was made available for illustrative and comparative purposes. The ICK approach can handle an arbitrary number of classes and arbitrarily complex indicator variogram models. But for a given resolution ratio F, the associated computational burden increases with the number of classes. Recall that the intent is the construction of multiple super-resolution maps from the available coarse and fine resolution information, not to make the most locally accurate super-resolution land cover map.

3.4.1 The Pearl River Delta example

This example considers a reference land cover classification derived from a Landsat TM scene of an area in the Pearl River Delta, South East China (Boucher et al., 2006); see also Seto et al. (2002) for more details. The reference land cover class map, shown in Figure 3.1, has dimension 15km×15km, and includes 500×500 fine resolution pixels. Each pixel has size $30m \times 30m$, and is considered as belonging to one of $K = 3$ broadly defined land cover classes: vegetation (white color, with

regional proportion 0.52), urban (gray color, with proportion 0.18), bare soil (black color, with proportion 0.30).

The reference land cover map of Figure 3.1 was upscaled into three coarse fraction maps using three different fine-to-coarse resolution ratios. For each coarse pixel, the corresponding class fractions are computed as the linear average of the fine pixel class indicators within that coarse pixel. More precisely, each coarse pixel is progressively comprised of $9 \times 9 = 81$, $15 \times 15 = 225$ and $25 \times 25 = 625$ fine pixels, with associated pixel size 270m $\times$ 270m, 450m $\times$ 450m and 750m $\times$ 750m; the resulting sets of coarse class fraction maps are shown in Figure 3.2. Since no observational errors are considered in this example, such synthetic fraction maps could be thought of as the outputs of a perfect spectral unmixing of data originating from sensors with coarser spatial resolution than that of Landsat TM.

The reference fine resolution indicator variograms for the three classes (not shown) were inferred from the land cover map of Figure 3.1. Each indicator variogram was modeled with a nugget effect contribution and two isotropic exponential functions (nested structures). The variogram model parameters, with partial sills given as proportions to their unit sum, are given in Table 3.1. The nugget effect contributions range from 0.10 to 0.14, indicating that the reference land cover map exhibits a small component of purely random spatial variability. The first exponential structure has ranges from 7 to 12 fine (TM) pixels, and shows similar partial sills across all classes (0.50 to 0.52). The second exponential structure has ranges from 65 to 70 fine pixels, and similar partial sills (0.36 to 0.39). The above variogram model parameters imply that: (i) more than 50% of the class spatial variability at the fine resolution occurs at scales smaller than the coarse pixel size, and (ii) for each upscaling scheme, all spatial variability occurs at scales smaller than eight, five and three coarse pixels, respectively, since $70/9 < 8$, $70/15 < 5$ and $70/25 < 3$.

In general, the larger the upscaling ratio, the less information is carried by the corresponding coarse resolution fractions. Such a loss of information due to averaging is a function of the variogram shape near the origin and its range: the larger that range or the smoother the underlying field, the smaller that information loss (Journel and Huijbregts, 1978). Since more than 50% of spatial variability in the reference land cover map of Figure 3.1 occurs within 7 to 12 TM pixels, one

should expect that the latter two upscaling schemes (15×15 and 25×25) would yield coarse resolution fractions with similar information content. A significant portion of the spatial information content in the reference land cover map is already lost going from the TM pixel size to the first coarse pixel size (9×9 upscaling).

Kriging

Using the coarse fraction maps of Figure 3.2, and the fine resolution indicator variogram models whose parameters are given in Table 3.1, the probabilities for class occurrence at any fine resolution pixel v_m^n that falls within any coarse pixel V_n are computed via ICK; see Equations (3.6) and (3.7). The data used for ICK at any such fine pixel v_m^n consist of a set of 21 coarse pixel neighbors defined as the 5×5 coarse template centered at V_n excluding the four corner pixels. That neighborhood geometry was chosen for computational efficiency reasons: the weights received by the four corner pixels are negligible with respect to those received by the remaining coarse pixels. For each upscaling scheme, the resulting set of three probability maps, one for each class, is shown in Figure 3.3. As expected from theory, these kriging-derived probability maps reflect the conditioning fraction data of Figure 3.2 without any block artifacts. The smooth transition between the ICK-derived probabilities computed at fine pixels within two adjacent coarse pixels V and V' is a consequence of considering the closest 21 neighboring coarse fraction data. If only the co-located coarse fractions $a_k(V)$ and $a_k(V')$ had been considered, strong discontinuities would be visible in the probability maps of Figure 3.3 near the coarse pixel boundaries.

However, the high frequency spatial variability seen on the reference land cover map of Figure 3.1 is not reproduced by the ICK-derived probability maps of Figure 3.3, especially for the larger upscaling ratio. This is as expected due to the smoothing effect of kriging and any interpolation algorithm in general. If one was to perform class allocation (using, say, a MAP criterion) by thresholding the ICK-derived probabilities of Figure 3.3, the spatial distribution of the resulting classes would have a much smoother spatial distribution than that seen in the reference land cover map of Figure 3.1. For a more detailed discussion of spatial prediction (kriging) versus simulation, the reader is referred to Goovaerts (1997).

To check the consistency of the kriging-based downscaling procedure in reproducing the coarse class fractions, the ICK-derived probabilities for the 25×25 upscaling scheme shown in Figure 3.3(g)-(i) were averaged within each coarse pixel, and compared to the corresponding coarse fractions of Figure 3.2(g)-(i). Figure 3.4 shows the resulting scatterplots: as expected from theory, see Equation (3.5), the coarse fraction data are reproduced; any mismatch is due to round-off errors.

Sequential simulation

Next, sequential indicator simulation (SIS) with the progressive servo-system defined in Equation (2.2) was performed, see Section 3.3. This yields multiple super-resolution realizations of land cover. The data used to generate these realizations were: (i) the different sets of coarse fraction images displayed in Figure 3.2, and (ii) the fine resolution indicator variogram models whose parameters are given in Table 1.

Three sets of such super-resolution realizations are shown in Figure 3.5; the realizations in each row were generated using the corresponding row of coarse fraction images shown in Figure 3.2 as conditioning data. Despite the significant difference in support provided by the coarse fraction images, the resulting super-resolution maps of Figure 3.5 exhibit similar spatial patterns although with some noise. One can accept, at least visually, that these simulated maps reproduce the low frequency spatial patterns of the reference land cover map of Figure 3.1. Bare soil (black color), for example, occurs in patches attached to the more elongated urban class (grey color). The high frequency spatial patterns (texture) found in the reference map of Figure 3.1 are well reproduced although with added noise, no matter the coarse fraction images used (i.e., across all upscaling ratios considered). This is not the case with the ICK-derived probability maps shown in Figure 3.3. As stated above, simulation aims at reproducing the prior model of spatial structure within the constraints of the available data. Since that fine resolution structural model is independent of the resolution of the coarse fractions, and these coarse fractions are consistent with the prior structural model (i.e., the reference indicator variograms were used for simulation), the resulting super-resolution realizations show realistic texture reproduction. Last, because the progressive correction algorithm was employed in generating the

super-resolution realizations of Figure 3.5, the corresponding coarse fractions are reproduced (up to round-off errors) when these fine resolution realizations are upscaled; see Figure 3.6.

The reproduction of the prior structural model at the target fine resolution is demonstrated in Figure 3.7, by comparing the indicator variograms of 25 SIS realizations with the input fine resolution indicator variogram models whose parameters are given in Table 3.1. These 25 realizations were conditioned to the coarse fraction data of Figure 3.2(g)-(i) corresponding to the 25×25 upscaling scheme. The indicator variograms shown in Figure 3.7 are not standardized to a unit sill, to better appreciate the different variances of land cover classes, which is equal to $\pi_k(1 - \pi_k)$ for a given class k. Overall, the indicator variograms computed from the simulated super-resolution maps reproduce the input model variograms; see Figure 3.7. These super-resolution realizations, however, exhibit a slightly higher degree of short-scale spatial variability; see the nugget contribution mismatch in Figure 3.7, particularly for the vegetation class. This mismatch is a consequence of the correction algorithm used to enforce the reproduction of the coarse class fractions of Figure 3.2(g)-(i).

Note that some non-repetitive (non-stationary) sub-pixel features, e.g., curvilinear roads, found in the reference land cover map of Figure 3.1 are not seen in the simulated maps of Figure 3.5. These features are lost during the upscaling procedure, and they cannot be retrieved from the coarse fractions alone without any additional fine resolution information. Such information is beyond that provided by the fine resolution indicator variograms, which can only characterize repetitive spatial patterns with elliptical or circular geometry. For a thorough discussion on the inadequacy of variograms to capture spatial patterns with complex geometrical characteristics, the reader is referred to Strebelle (2002) and to Chapter 4.

The procedure conducted above for checking the reproduction of coarse resolution information, was also repeated for the case of SIS without the the servo-system. Three such super-resolution realizations, generated using the coarse fraction data of Figure 3.2(g)-(i) corresponding to the 25×25 upscaling scheme, are shown in Figure 3.8. There is noticeably less speckles in the image when the servo-system is not used. However, these realizations do not reproduce the corresponding coarse fractions when upscaled; see Figure 3.9(a)-(c). Globally, and for all classes, the upscaled simulated

coarse fractions are unbiased; that is, the differences between simulated and observed coarse class fractions have a near zero mean. In addition, the super-resolution maps of Figure 3.8 reproduce the input indicator variogram models better than the realizations shown in Figure 3.5(g)-(i); compare Figure 3.9(d)-(f) to Figure 3.7(g)-(i). Note also that the indicator variograms of the simulated super-resolution maps fluctuate more when the correction algorithm is not used, because these simulated realizations are not as tightly constrained by the coarse fractions.

To illustrate the impact of different prior models of spatial structure on the resulting super-resolution maps, consider two new sets of fine resolution indicator variogram models; recall that the parameters of the reference indicator variogram models were given in Table 1. The first new set of variograms consists of pure nugget effect models, whereas the second set consists of exponential models with zero nugget contribution and large ranges equal to 500 fine pixels; that is, approximately 7 times larger than the largest range of the reference models. The former models correspond to an assumption of purely random spatial variation of classes, whereas the latter models correspond to an assumption of strong continuity of such classes.

To better appreciate the spatial structure associated with these different sets of indicator variogram models in the absence of conditioning data, three unconditional realizations are given in Figure 3.10(a)-(c); these were generated without accounting for any coarse fraction data or class labels at some informed fine resolution pixels, hence the term unconditional. Figure 3.10(a) exhibits the characteristic "salt and pepper" texture of a white noise image, whereas Figure 3.10(c) exhibits class labels that are spatially arranged in very large patches. The same sets of indicator variogram models were subsequently used to generate the three conditional realizations shown in Figure 3.10(d)-(f). Because these latter realizations are conditioned to the coarse fraction data shown in Figure 3.2(g)-(i) corresponding to the 25×25 upscaling scheme, they reproduce these coarse fraction values when upscaled. The differences between Figures 3.10(a)-(c) and 3.10(d)-(f) lie in the effect of the conditioning coarse fraction data. It is the local information carried by these coarse fractions that forces the spatial patterns of land cover classes to be located where they are in the latter set of realizations. Note that the simulated super-resolution map of Figure 3.10(f)

exhibits significantly different spatial patterns than those implied by the prior variogram model as shown in Figure 3.10(c). Indeed, with the ICK super-resolution mapping method, the data (i.e., the coarse fraction values in this case) exert a far greater influence on the spatial patterns of the resulting super-resolution maps than the prior model of spatial structure adopted. In the case of Figure 3.10(f), that prior model of spatial structure was inconsistent with the observed coarse fractions, since these fractions were generated from the reference land cover map of Figure 3.1 whose indicator variograms did not imply such strong spatial continuity. Consequently, one can conclude that exhaustive coarse fraction data (not the prior model) controls generation of the patterns found in the super-resolution land cover map of Figure 3.10(f).

Comparing the various conditional realizations, one can easily appreciate that Figure 3.10(d), which corresponds to the assumption of lack of spatial correlation, differs clearly by exhibiting more fragmented classes than Figures 3.10(e)-(f). Figures 3.10(e) and (f) differ slightly in the extent of class patches, the latter exhibiting larger patches (less fragmentation) than the former due to the assumption of stronger spatial continuity.

To corroborate the above statements, the reference land cover map of Figure 3.1 was first converted into three binary images of class indicators. Each binary image was then subjected to connected components labeling (using 8-point connectivity) to define objects comprised of groups of *connected* pixels with the same class indicator; see, for example, Gonzalez and Woods (2002). The natural logarithm of the areas of objects found in each binary image is recorded for the corresponding land cover class. The histogram of such object areas can be viewed as a measure of spatial structure going beyond that encapsulated by indicator variogram models. The same connected components labeling procedure was then applied to the simulated super-resolution maps of Figure 3.10(d)-(f). The distributions of the natural logarithm of object areas derived from the reference and from the three simulated land cover maps were then compared using the quantile-quantile (Q-Q) plots given in Figure 3.11. Quantiles for object areas derived from the reference land cover map pertain to the x-axes of these plots, whereas quantiles derived from object areas found in the simulated maps pertain to the y-axes of these plots. A perfect agreement between the distributions

	nugget	structure 1		structure 2	
class		sill	range	sill	range
#1	0.09	0.52	12	0.39	70
#2	0.14	0.50	7	0.36	70
#3	0.10	0.52	9	0.38	65

Table 3.1: Parameters of indicator variogram models for the reference land cover map of Figure 3.1. Both structures 1 and 2 refer to exponential variogram functions, partial sills are expressed as proportions of a unit total sill, and ranges are expressed in TM pixels.

of reference and simulated object areas would entail a Q-Q plot with bullets aligned on the diagonal solid line or first bisector. When a simulated super-resolution map exhibits more (or less) fragmentation than the reference land cover map, the bullets of the associated Q-Q plot are aligned below (above) the diagonal line.

Figure 3.11(a)-(c) corroborates that the simulated super-resolution map of Figure 3.10(d) exhibits much more class fragmentation than the reference land cover map of Figure 3.1: the Q-Q plots for all classes are all below the diagonal lines, indicating simulated objects with smaller areas than those found in the reference map. Figures 3.11(d)-(f) and 3.11(g)-(i) show a much better agreement between simulated and reference object areas than that found in Figure 3.11(a)-(c), indicating that a purely random model of spatial structure was inappropriate as expected. Comparing Figures 3.11(d)-(f) and 3.11(g)-(i), the two super-resolution maps of Figure 3.10(e) and (f) do not significantly differ with respect to the above fragmentation metric. Only some small differences can be detected in favor of the super-resolution map of Figure 3.10(e), in particular for the urban and bare soil classes; compare Figure 3.11(e)-(f) with Figure 3.11(g)-(i).

Note that the conditioning data control to some extent the type of fine scale patterns to be generated. Even if an inappropriate model of spatial structure is adopted the available coarse fractions did not allow the generation of simulated super-resolution maps with large patches of class labels; compare Figure 3.10(c) with Figure 3.10(f). The impact of the structural model is increased for patterns with spatial extent smaller than the size of a coarse pixel.

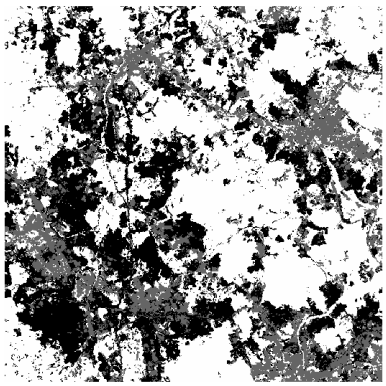
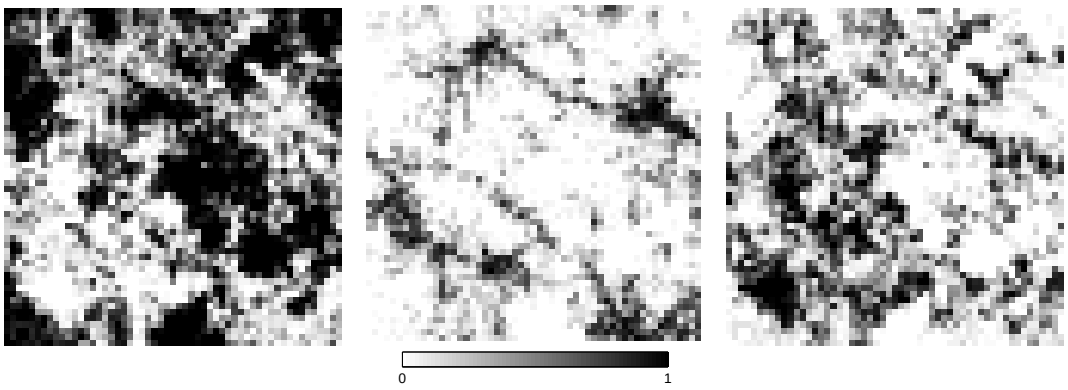
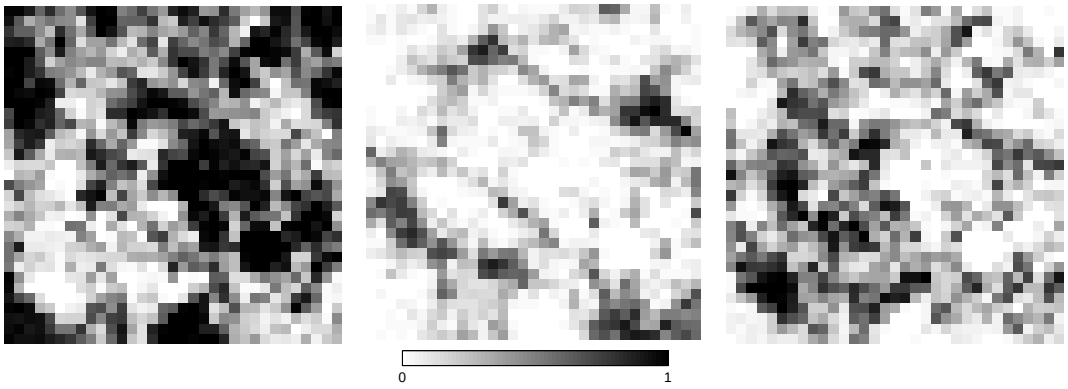


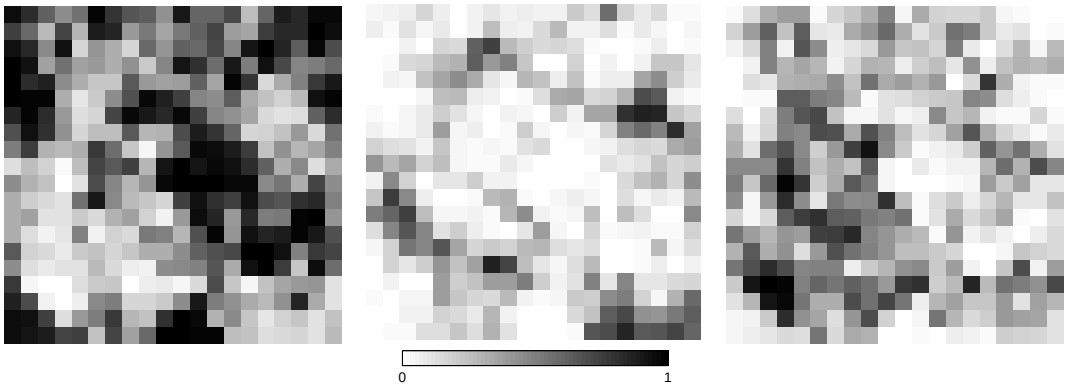
Figure 3.1: Reference land cover classification based on Landsat TM imagery of 500×500 pixels, each of size $30m \times 30m$. White color indicates vegetation, gray urban, black bare soil. The scene is 15km by 15km.



(a) Vegetation (b) Urban (c) Bare soil

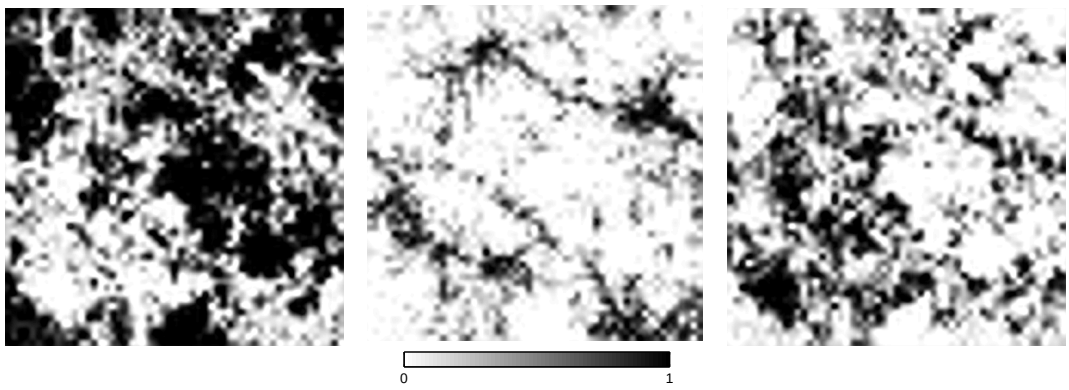


(d) Vegetation (e) Urban (f) Bare soil

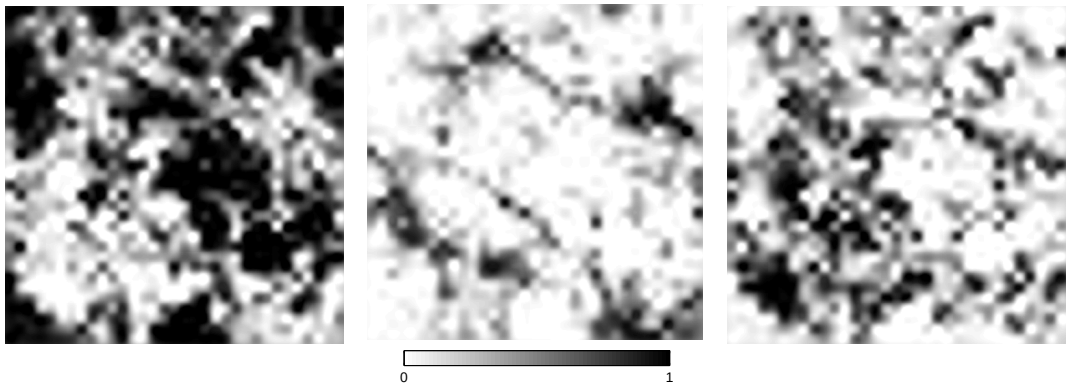


(g) Vegetation (h) Urban (i) Bare soil

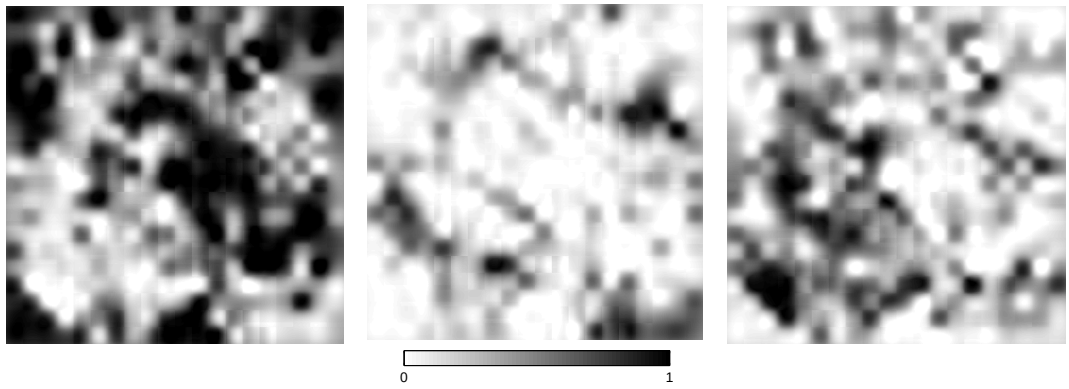
Figure 3.2: Maps of class fractions obtained at progressively coarser spatial resolutions. The fraction value at each coarse pixel is the average of the fine resolution class indicators within that coarse pixel derived from Figure 3.1. **Top row:** 9×9 upscaling scheme with pixel size 270m $\times$ 270m. **Middle row:** 15×15 upscaling scheme with pixel size 450m $\times$ 450m. **Bottom row:** 25×25 upscaling scheme with pixel size $750m \times 750m$. The scene is 15km by 15km.



(a) Vegetation (b) Urban (c) Bare soil

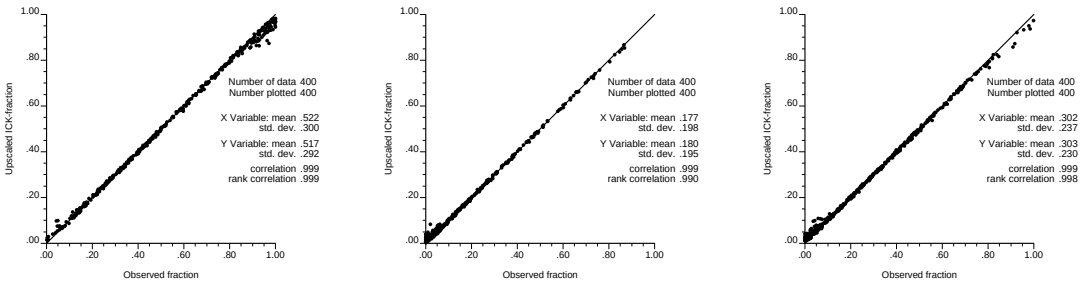


(d) Vegetation (e) Urban (f) Bare soil



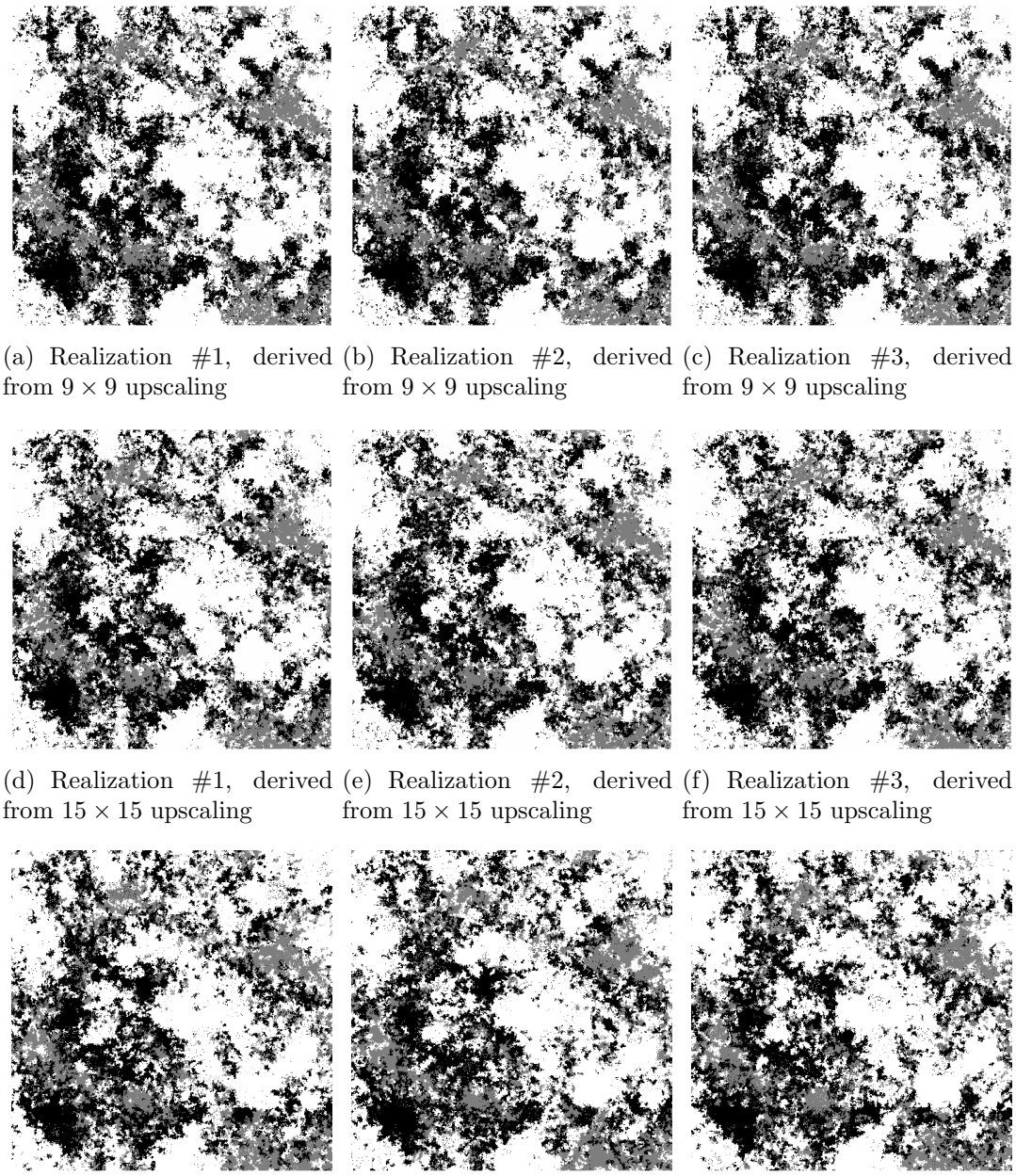
(g) Vegetation (h) Urban (i) Bare soil

Figure 3.3: Conditional probabilities of fine resolution class occurrence computed via simple indicator cokriging (ICK). For each row, the conditioning information consists only of the coarse resolution fractions of the corresponding row of Figure 3.2. **Top row:** 9×9 upscaling scheme. **Middle row:** 15×15 upscaling scheme. **Bottom row:** 25×25 upscaling scheme. The scene is 15km by 15km.



(a) Vegetation (b) Urban (c) Bare soil

Figure 3.4: Reproduction of coarse class fractions by the corresponding upscaled ICK-derived probabilities for the 25×25 upscaling scheme of Figure 3.3(g)-(i).



(a) Realization #1, derived from 9×9 upscaling

(b) Realization #2, derived from 9×9 upscaling

(c) Realization #3, derived from 9×9 upscaling

(d) Realization #1, derived from 15×15 upscaling

(e) Realization #2, derived from 15×15 upscaling

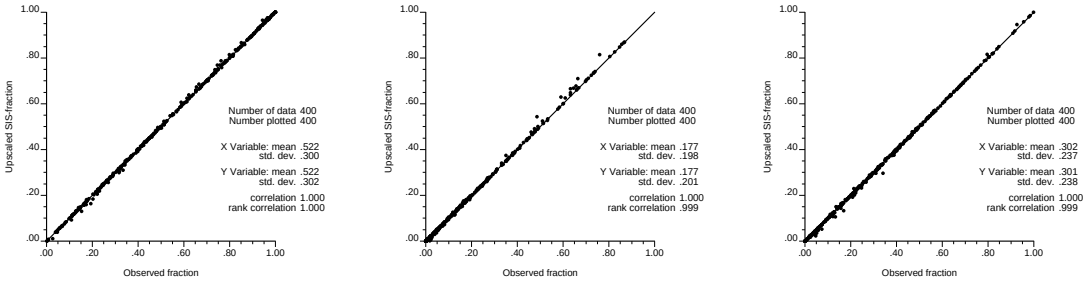
(f) Realization #3, derived from 15×15 upscaling

(g) Realization #1, derived from 25×25 upscaling

(h) Realization #2, derived from 25×25 upscaling

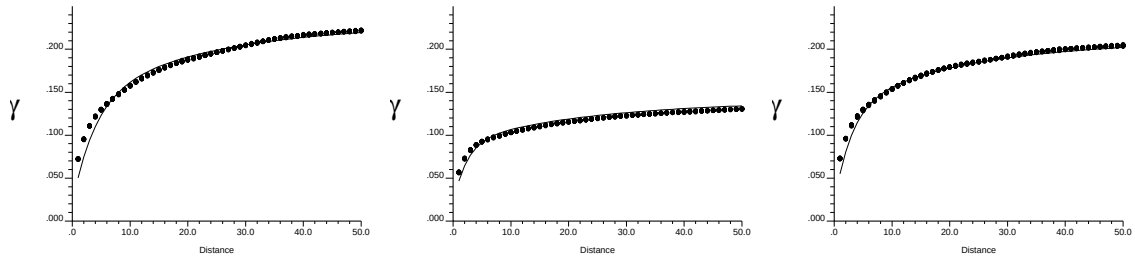
(i) Realization #3, derived from 25×25 upscaling

Figure 3.5: Simulated super-resolution land cover maps, generated using SIS with the progressive correction algorithm; see text for details. White color indicates vegetation, gray urban, black bare soil. For each row, the conditioning information consists only of the coarse resolution fractions of the corresponding row of Figure 3.2. The scene is 15km by 15km.

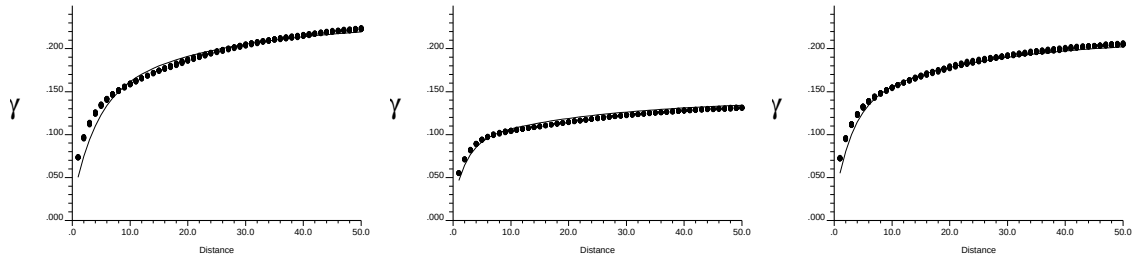


(a) Vegetation (b) Urban (c) Bare soil

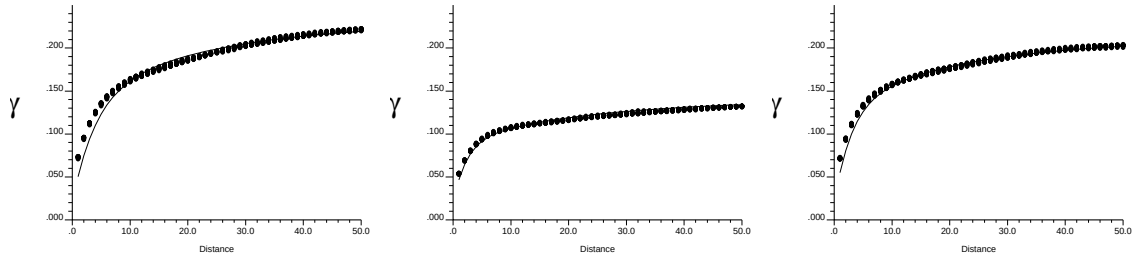
Figure 3.6: Reproduction of the coarse class fractions for the 25×25 upscaling scheme shown in Figure 3.2(g)-(i) by the corresponding upscaled simulated super-resolution realizations of Figure 3.5(g)-(i).



(a) Vegetation, 9×9 upscaling (b) Urban, 9×9 upscaling (c) Bare soil, 9×9 upscaling

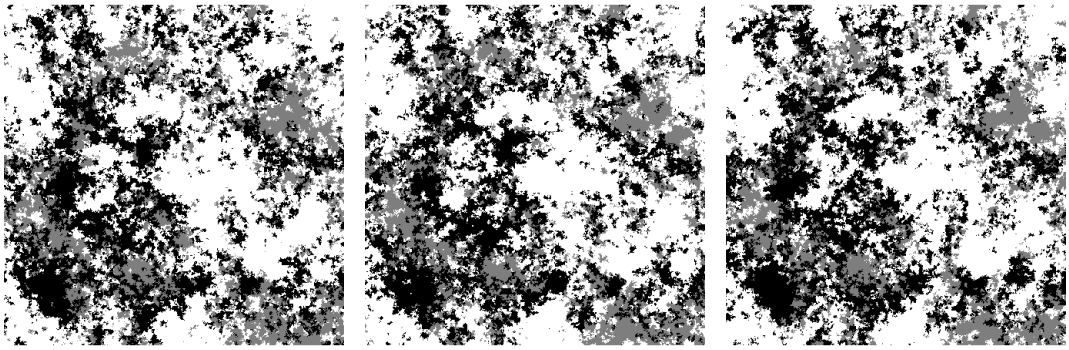


(d) Vegetation, 15×15 upscaling (e) Urban, 15×15 upscaling (f) Bare soil, 15×15 upscaling



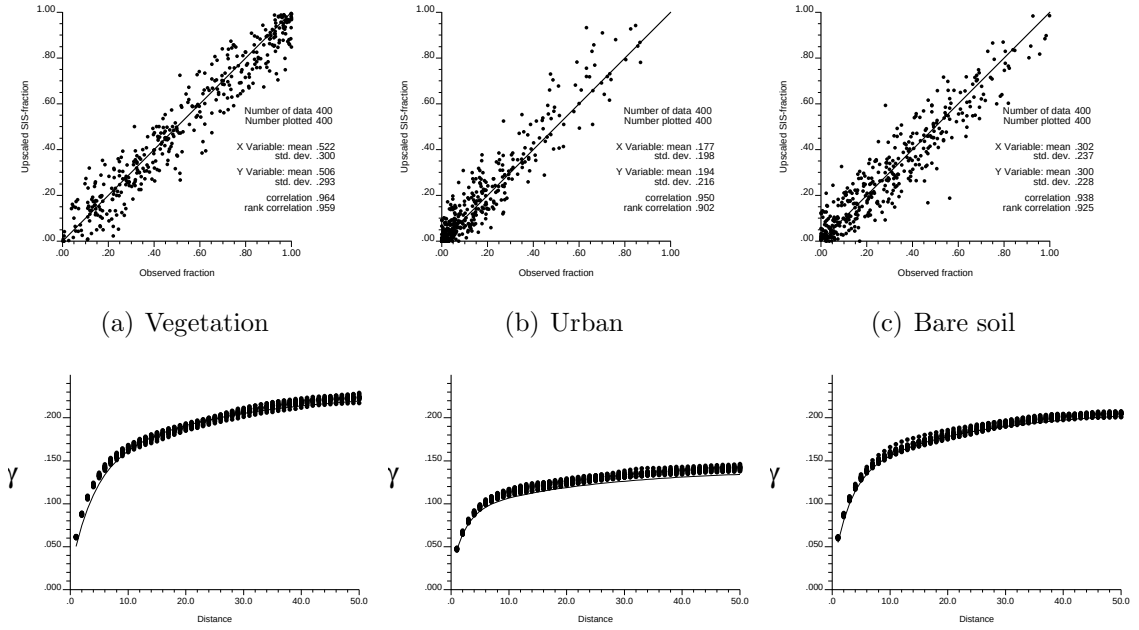
(g) Vegetation, 25×25 upscaling (h) Urban, 25×25 upscaling (i) Bare soil, 25×25 upscaling

Figure 3.7: Indicator variogram reproduction for 25 super-resolution realizations of land cover generated conditional to progressively coarser class fraction data; see text for details. Solid lines indicate the reference isotropic variogram models, dots correspond to variograms of simulated super-resolution land cover maps.



(a) Realization #1 (b) Realization #2 (c) Realization #3

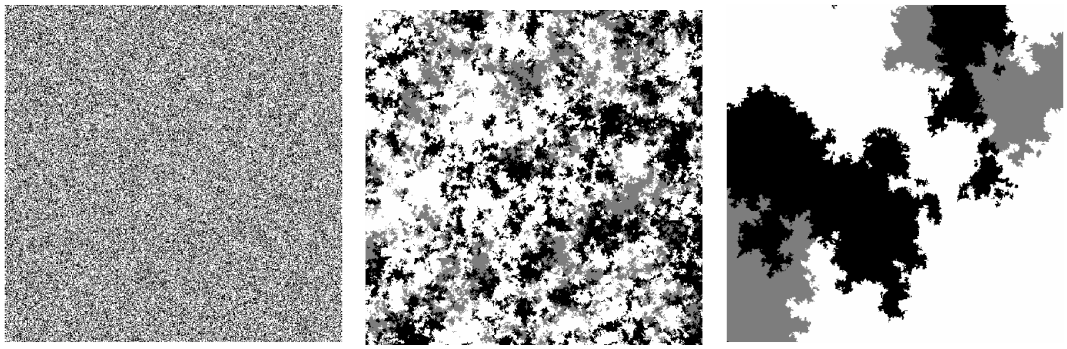
Figure 3.8: Three super-resolution realizations of land cover generated using SIS without servo-system; see text for details. White color indicates vegetation, gray urban, black bare soil. The conditioning information consists of the coarse resolution fractions of Figure 3.2(g)-(i) corresponding to the 25×25 upscaling scheme. The scene is 15km by 15km.



(a) Vegetation (b) Urban (c) Bare soil

(d) Vegetation (e) Urban (f) Bare soil

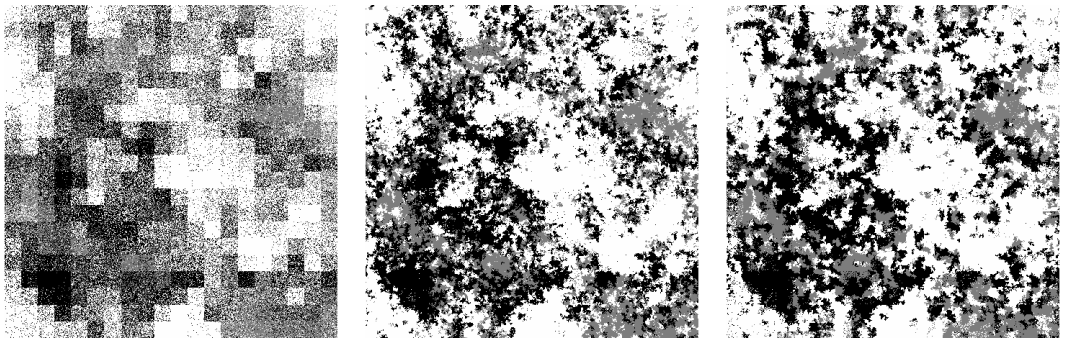
Figure 3.9: **Top row:** Reproduction of the coarse class fractions for the 25×25 upscaling scheme shown in Figure 3.2(g)-(i) by the upscaled simulated super-resolution realizations of Figure 3.8. **Bottom row:** Indicator variogram reproduction for 25 super-resolution realizations of land cover from the 25×25 upscaling scheme without the progressive correction algorithm; see text for details. Solid lines pertain to the reference isotropic indicator variogram models, whereas dots pertain to indicator variograms of simulated super-resolution land cover maps.



(a) Purely random; no coarse fraction data

(b) Reference variogram; no coarse fraction data

(c) Strong continuity; no coarse fraction data

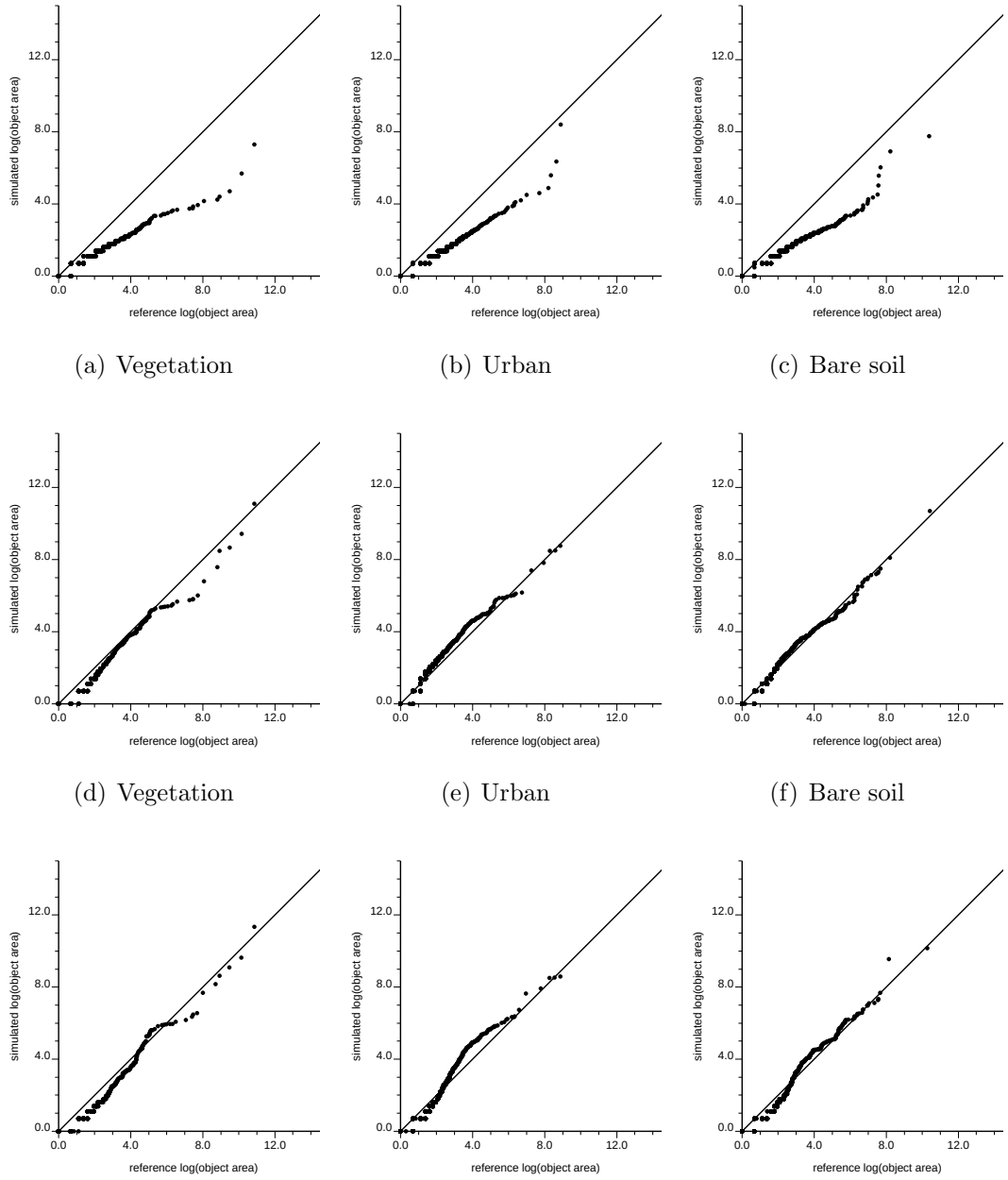


(d) Purely random

(e) Reference variogram

(f) Strong continuity

Figure 3.10: **Top row:** Three unconditional super-resolution realizations of land cover, generated using different indicator variogram models; see text for details. **Bottom row:** Three super-resolution realizations of land cover conditioned to the coarse fractions of Figure 3.2(g)-(i) corresponding to the 25×25 upscaling scheme, generated using the same indicator variogram models used for Figure 3.10(a)-(c); see text for details. White color indicates vegetation, gray urban, black bare soil. The scene is 15km by 15km.



(a) Vegetation (b) Urban (c) Bare soil

(d) Vegetation (e) Urban (f) Bare soil

(g) Vegetation (h) Urban (i) Bare soil

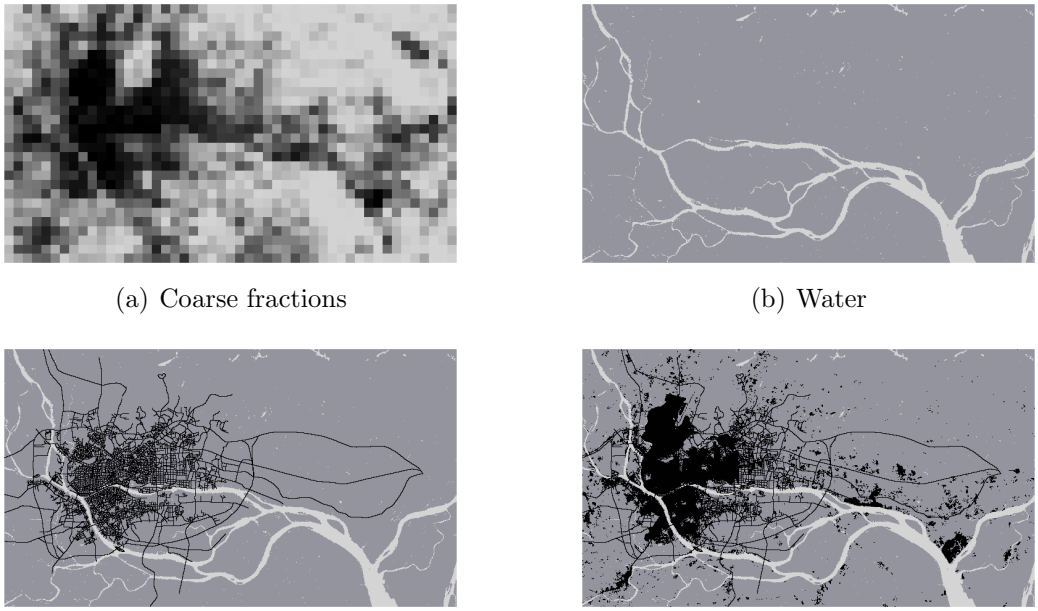
Figure 3.11: Quantile-quantile plots between distributions of object areas derived from the reference land cover map of Figure 3.1 for each class, and the super-resolution realizations from: Figure 3.10(d) (random model) – **top row**, Figure 3.10(e) (reference model)– **middle row**, and Figure 3.10(f) (maximum continuity model) – **bottom row**.

3.4.2 The Guangzhou example

The integration of fine spatial resolution data into sub-pixel class mapping is illustrated with a data set of impervious surfaces derived from Seto and Fragkias (2005) and Fragkias and Seto (2007). This data set relates to the city of Guangzhou, capital of the Guangdong province in China. Figure 3.12 shows the binary image of impervious versus non-impervious surfaces, corresponding to a scene of size 735×420 fine pixels with nominal size $60m \times 60m$. That reference class map is upscaled to generate the coarse class fraction data shown in Figure 3.13(a); each coarse pixel contains 15×15 fine pixels, and the upscaled image is comprised of 49×28 coarse pixels.

Figure 3.12: Reference binary image of impervious (black) versus non-impervious (light grey) surfaces for the city of Guangzhou, China. Image extent: 735×420 fine pixels.

In addition to the coarse class fractions of Figure 3.13(a), three sets of fine spatial resolution data are made available, each with an increasing amount of information: (i) a map of a water body in Figure 3.13(b), (ii) a road network map in Figure 3.13(c), and (iii) a map of previously mapped impervious surfaces in Figure 3.13(d). This fine spatial resolution information is assumed to have been derived from pertinent databases in a geographic information system (GIS). The pixel counts for Figure 3.13(b) to (d) are shown in Table 3.2; fine spatial resolution data cover ranges from 6.5% to 22% of the entire image. Figure 3.13(b) provides only information about non-impervious surfaces while Figure 3.13(b) and (d) contain both impervious and non-impervious data.



(a) Coarse fractions

(b) Water

(c) Road network and water

(d) Previously mapped impervious areas + (b) and (c)

Figure 3.13: Available data: (a) coarse fractions for the impervious class obtained by upscaling the reference image of Figure 3.12, (b) river and known water bodies providing only information about the absence of the black class; light grey pixels indicate non-impervious surfaces, whereas dark grey pixels indicate background, (c) road network and water, providing additional information on impervious surfaces denoted by black pixels, and (d) information in (b) and (c) along with previously mapped impervious surfaces.

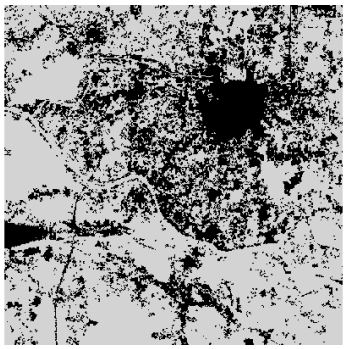
	Figure 3.13(b)	Figure 3.13(c)	Figure 3.13(d)
Impervious	0 (0%)	25118 (8.1%)	48517 (15.7%)
Non-Impervious	20096 (6.5%)	19457 (6.3%)	19457 (6.3%)
Total	20096 (6.5%)	44575 (14.4%)	67974 (22.0%)

Table 3.2: Number of pixels (and percentage of full image) with known class labels for each of the three fine spatial resolution data schemes.

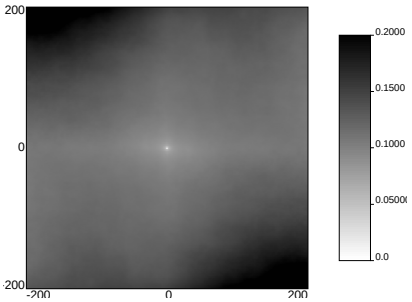
Structural model

Figure 3.14(a) shows a map of impervious areas around the city of Foshan also in the Guangdong province. This map is chosen as representative of the class patterns expected to occur at the fine spatial resolution in the Guangzhou area. In other words, the binary class map of Figure 3.14(a) for Foshan is used as an analog image to infer a prior model of spatial structure for the unknown class labels in the Guangzhou area. Note that the proportion of impervious surface in the reference image is similar to that in the analog image: 35.5% for Guangzhou versus 32.5% for Foshan.

The indicator semivariogram map calculated from the analog image of Figure 3.14(a) using the FFT method described in the Section 3.1.2 is displayed in Figure 3.14(b). This indicator semivariogram map is a limited summary statistics of the spatial patterns seen in Figure 3.14(a). The semivariogram map has been calculated on a table of size 401×401 to provide structural information between all data retained within a search neighborhood up to 100×100. Using the map of Figure 3.14(b) as a prior model of fine spatial resolution spatial structure for the Guangzhou area, implies that the 2-point statistical texture in the analog image is assumed representative of that in the reference image.



(a) Analog scene for impervious surfaces (black). Size: 500×500



(b) Indicator semivariogram map. Size: 401×401 fine pixels

Figure 3.14: Analog binary image and corresponding indicator semivariogram map for the impervious class.

Downscaling coarse fractions into class occurrence probabilities

The coarse fraction data of Figure 3.13(a) are downscaled into probabilities of impervious surface occurrence using ICK and the indicator semivariogram map of Figure 3.14(b). A different probability map is produced for each level of fine spatial resolution data shown in Figure 3.13(b) to (d), and the resulting four ICK-derived probability maps are shown in Figure 3.15. These ICK-derived probabilities do not directly classify pixels as impervious or not, but instead represent estimates of the likelihood that a particular pixel be impervious, given the coarse fraction data and the available fine spatial resolution information.

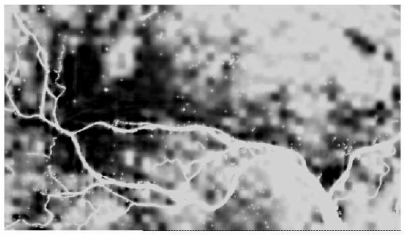
When fine spatial resolution information, such as the river in Figure 3.15(b), is included in super-resolution mapping, the kriging system of equations ensures that the resulting probabilities reproduce the known class labels of the fine spatial resolution map. In our case, this means that ICK will yield a probability of 1 for impervious surface at a fine pixel that has been flagged as impervious, and 0 otherwise. Such extreme probability values indicate no uncertainty at those pixels. Precisely due to this exactitude kriging property, the road networks are clearly reproduced on the ICK map of Figure 3.15(c) and (d). The addition of the previously mapped impervious data in Figure 3.15(d) does not add details for the inner part of the city where the road network from Figure 3.13(c) and the coarse fractions already provide significant information. It does, however, produce more compact impervious surfaces with crisper boundaries. The number of uninformed pixels, for each fine-resolution data scheme, is shown in Table 3.2.

It is rather tempting to apply a maximum a posterior (MAP) allocation rule for classifying the ICK-derived probability maps shown in Figure 3.15. Indeed, the MAP allocation is simple: at any fine pixel, the label that corresponds to the most likely class, i.e., to the one with the highest probability of occurrence, is assigned at that pixel:

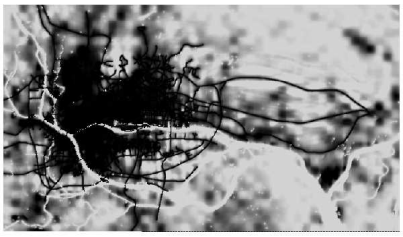
$$\hat{c}_k(v_m^n) = \begin{cases} 1 \;\; \text{(impervious)} \;\; \text{if} \;\; p_k^{IK}(v_m^n) > 0.5 \\ 0 \;\; \text{(non-impervious)} \;\; \text{otherwise} \end{cases}$$

Such MAP-derived super-resolution class maps are shown in Figure 3.16(a) through (d). The corresponding scatter plots between the upscaled fractions computed from

(a) Coarse fractions only



(b) Coarse fractions and water

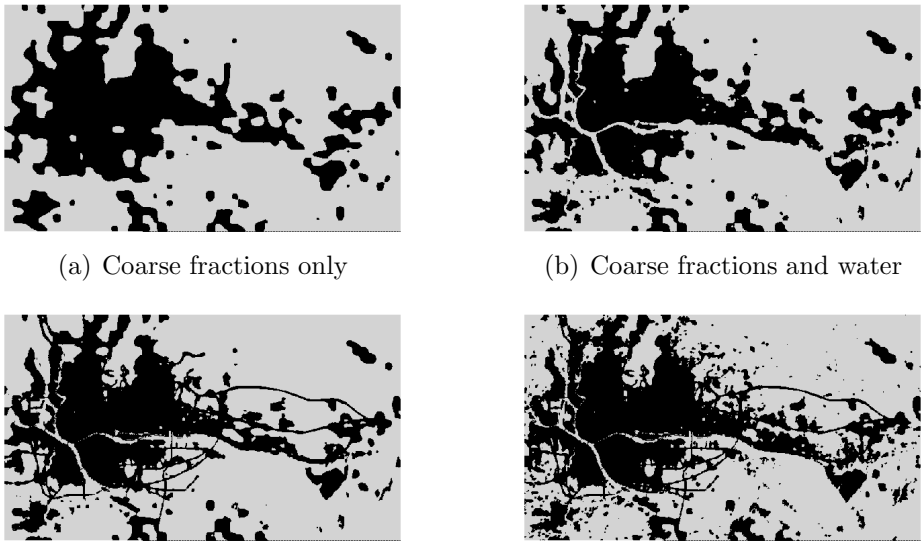


(c) Coarse fractions, river and road network

(d) Coarse fractions, river, road and previously mapped impervious area

Figure 3.15: Downscaled probabilities of occurrence for impervious surfaces conditional to (a) the coarse fraction data of Figure 3.13(a), and (b) through (d) additional fine spatial resolution information.

these maps and the input coarse fractions of Figure 3.13(a) are shown in Figure 3.17(a) to (d). As expected from theory, this per-pixel allocation rule does not guarantee coarse fraction reproduction, which is a multi-pixel constraint. Moreover, the patterns displayed in the MAP-derived maps are too smooth compared to both the analog image of Figure 3.14(a) and the reference image of Figure 3.12.



(a) Coarse fractions only

(b) Coarse fractions and water

(c) Coarse fractions, water and road network

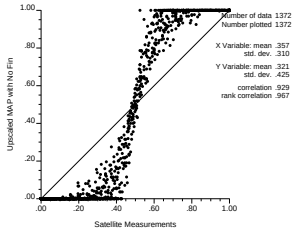
(d) Coarse fractions, water, road and previously mapped impervious area

Figure 3.16: Maximum a posteriori (MAP) classification based on the probability maps of Figure 3.15.

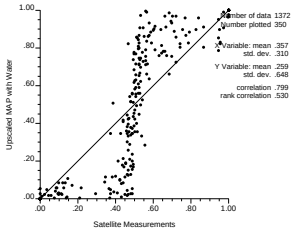
Simulating super-resolution class maps

The above results illustrate that the per-pixel MAP allocation rule applied to ICK-derived probabilities of class occurrence does not reproduce the coarse fraction data or the spatial patterns. The sequential simulation formalism with ICK-derived probabilities, however, can be used to fulfill both these objectives. Recall that stochastic simulation does not aim at achieving local accuracy, but aims at spatial pattern accuracy or reproduction.

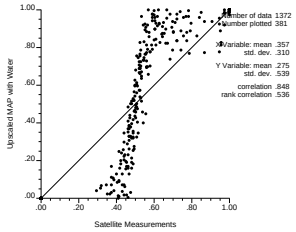
One simulated super-resolution class map, among many possible ones, is shown in Figure 3.18 for each of the four sets of conditioning data given in Figure 3.13. All



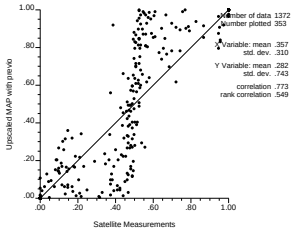
(a) Coarse fractions only



(b) Coarse fractions and water



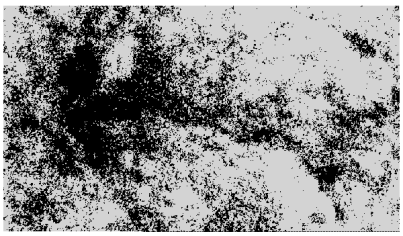
(c) Coarse fractions, river and road network



(d) Coarse fractions, river, road and previously mapped impervious area

Figure 3.17: Coarse fraction reproduction when the MAP allocation rule is applied to the probability maps of Figure 3.16.

simulations reproduce by construction both the fine spatial resolution class labels and the coarse class fractions. Note the difference between these super-resolution class maps and the smooth MAP-derived maps of Figure 3.16. Note also that the patterns displayed by the SIS-generated maps are not only function of the amount of conditioning information, but also (and most importantly) of the indicator semivariogram map of Figure 3.14(b). Contrast that to the MAP-derived maps of Figure 3.16, where the smooth spatial patterns are predominantly dictated by the number of fine spatial resolution data.



(a) Simulation #1 Coarse fraction only

(b) Simulation #1 Water data

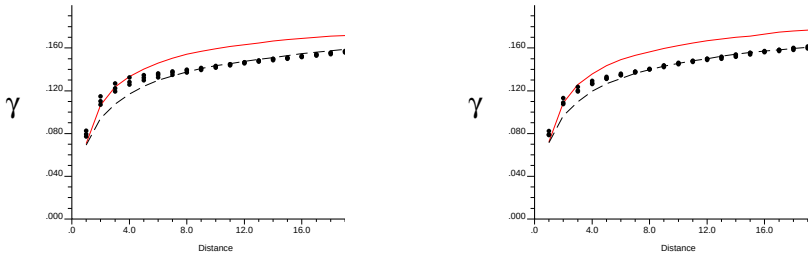
(c) Simulation #1 Road data

(d) Simulation #1 Past data

Figure 3.18: Simulated super-resolution class maps conditioned to the different data sets shown in Figure 3.13.

The approximate reproduction of the indicator semivariogram map of Figure 3.14(b), along directions North-South and East-West, by the simulated class maps of Figure 3.18 is corroborated in Figure 3.19. Note that the super-resolution maps are not expected to reproduce the indicator semivariograms of the reference image of Figure 3.12, since these were never given to the simulation algorithm. It is expected, however, that the simulated class maps reproduce the indicator semivariogram map of Figure 3.14(b), as shown by Figure 3.19. In addition, all the SIS-generated maps

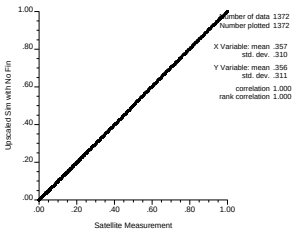
reproduce exactly the coarse fraction data, as checked in Figure 3.20.



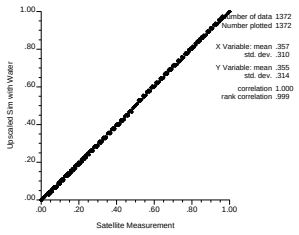
(a) E-W direction (b) N-S direction

Figure 3.19: Indicator semivariogram reproduction along directions E-W and N-S. Solid lines correspond to indicator semivariogram values computed from the analog image of Figure 3.14(a), dashed lines correspond to such values computed from the reference image of Figure 3.12, and dots correspond to such values computed from the simulated super-resolution maps of Figure 3.18.

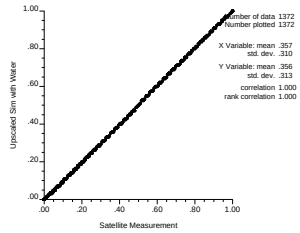
Recall that the ICK-derived probability maps of Figure 3.15 provide a model of uncertainty regarding the unknown class label at any fine pixel. Stochastic simulation provides an even richer uncertainty model through multiple super-resolution maps, all compatible with the prior structural model within the constraints imposed by the conditioning data.



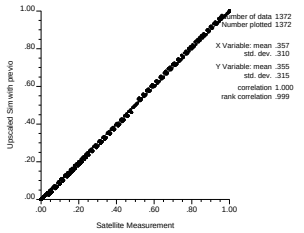
(a) Coarse fractions only



(b) Coarse fractions and water



(c) Coarse fractions, river and road network



(d) Coarse fractions, river, road and previously mapped impervious area

Figure 3.20: Coarse fraction reproduction from the SIS-generated class maps of Figure 3.18.

Chapter 4

Downscaling with training image

An alternative to characterizing spatial patterns via indicator variograms is to consider a training image (TI); see for example, Strebelle (2002). A TI is an image depicting the patterns of class labels expected to be seen on the landscape at the target spatial resolution. Indicator variograms are analytically formulated but cannot distinguish spatial patterns beyond their two-point statistics. A TI is not analytical but can depict complex spatial patterns. Note that a TI is not some initial classification based on the coarse fractions, if only because it carries no local accuracy. It is a mere conceptual representation of the spatial patterns (texture) of class labels expected to be seen at the target resolution. Note that, as with variogram-based geostatistics, one needs to invoke stationarity when using TIs. Stationarity in this case entails that a TI must contain enough repetitive patterns, so that their frequency of occurrence can be reliably estimated.

Simulation from a TI can be performed using the single normal equation simulation (SNESIM) algorithm (Strebelle, 2002). In this framework, the structural information provided by the indicator variogram models is replaced by a TI. The local CPMF $\Pr\left\{I_k(v_m)=1|\Omega_{(m-1)}^{W(v_m)}\right\}$ of Equation 2.3 is no longer computed by solving an ICK system of equations, see Equation 3.4. Instead that CPMF is estimated by: (i) defining a fixed neighborhood template W, (ii) scanning the TI with that template, and (iii) extracting the frequency of the k-th class occurrence at the central pixel of that template given the conditioning data event $\Omega_{(m-1)}^{W(v_m)}$. A repetitive re-scanning of the TI for each data event at each fine pixel v_m along the sequential simulation

random path would not be efficient. Instead, the TI is scanned only once and all patterns within the template W are stored in a search tree structure for quick and repetitive retrieval.

The classical SNESIM algorithm, discussed in detail below, cannot condition to data defined on larger support such as coarse fraction data. This chapter offers two methods to extend SNESIM to accept such conditioning.

4.1 The SNESIM algorithm

4.1.1 Building a search tree

Consider the schematic training image and the four-pixel search template given in Figure 4.1, the corresponding search tree is shown in Figure 4.2. The search tree records all the patterns found in the TI and their frequency of occurrence given the template geometry. Each tree level corresponds to one of the five pixels shown in Figure 4.1.

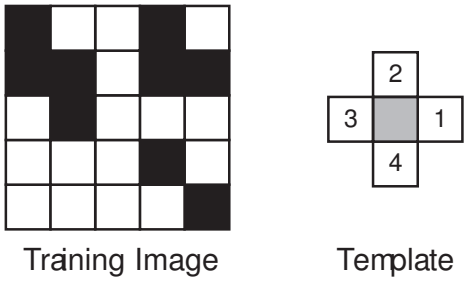


Figure 4.1: Left: Training image with two classes: white and black. Right: Search template of five pixels arranged on a cross: the central grey pixel corresponds to the pixel with unknown class label, the four white pixels noted 1 to 4 correspond to pixels where class labels can be known, i.e., to a data event.

The starting point for building a binary search tree is the root (level 0), which corresponds to an empty data event. The frequency associated with the root reflects the global proportion of black and white pixels in an "eroded" TI. The eroded TI is the biggest subset of the original TI, within which the search template is always fully contained; this eliminates edge effects. In Figure 4.1, the eroded TI corresponds to

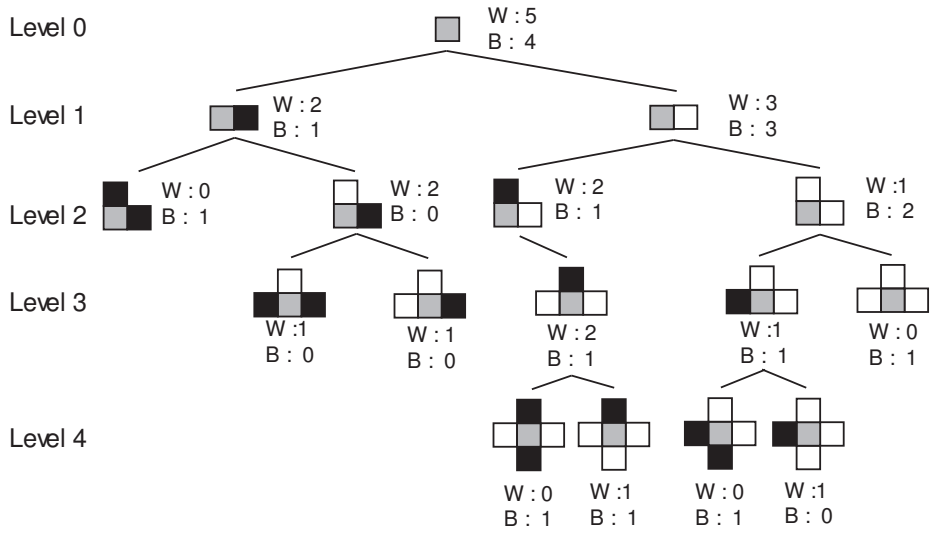


Figure 4.2: Search tree built from the synthetic TI and template of Figure 1. The frequency for each pattern is computed from the eroded TI, with W and B indicating the number of white and black class occurrence at the central pixel given the data event.

the inner 3x3 grid. In that eroded TI, there are 5 white and 4 black pixels, which complete the information at level 0 of the search tree shown in Figure 4.2.

Level 1 of the search tree in Figure 4.2 involves class labels both at the central template pixel and its neighboring pixel #1; see Figure 1. Consider the case of pixel #1 being black, count how many times the central template pixel is black or white; this procedure generates the left branch at level 1 of the tree structure in Figure 4.2. There are three occurrences of a black template pixel #1 in the eroded TI. Two of them are associated with a white central pixel, and one with a black central pixel. Similarly the right branch at level 1 of the tree structure corresponds to the template pixel #1 being white and counting how many times the central pixel appears as black or white. In this case, the navigation rules in the search tree dictate that a white pixel leads right branches, and a black pixel left branches. The same procedure is applied for template pixels #3 and #4 to complete the remaining levels of the search tree. Search trees are not limited to two classes, although the enumeration of pattern combinations associated with too many classes may lead to very large trees and increased computational cost. Note also that a search tree is specific to a given

template size and geometry.

The search tree data structure allows, in addition to storing the actual TI patterns, to rapidly retrieve their frequency of occurrence. The local CPMF at any pixel along the random path can be directly read from the search tree instead of having to scan anew the TI.

4.1.2 Retrieving probabilities from a search tree

Retrieving the conditional probability associated with a spatial pattern in a fully informed search template is straightforward; it can be read directly by moving along branches on the left or right until the final level is reached. The task is slightly more difficult when the data event is incomplete. When an uninformed template pixel forbids the passage to the next tree level, that pixel is assigned all possible classes (here two) then the search keeps branching out.

Consider the incomplete data event shown in Figure 4.3. Template pixels #2 and #3 are informed but not pixels #1 and #4. It is not possible to access the last level 3 to retrieve the CPMF without going first to level 1 which is associated with an unknown class label. Instead, level 3 is reached by informing template pixel #1 first with a black then with a white category, going down the tree and weighting the two resulting CPMFs. There is a probability of $6/9 = 2/3$ that template pixel #1 is white and $1/3$ that it is black. At that point, the search is divided in two branches; one given that pixel #1 is white, the other given that pixel #1 is black. From level 1, each branch needs to access its right children since pixel #2 is white, then its left children to get to the last level 3. From the white pixel #1 branch, there is a probability of $1/2$ that the central pixel is white; the black pixel #1 branch yields a probability of 1. Weighting these two probabilities as a function of the likelihood of each branch gives a probability of $2/3 \times 1/3 + 1/3 \times 1 = 2/3$ for a white central pixel and of $2/3 \times 1/2 + 1/3 \times 0 = 1/3$ for a black central pixel.

When a data event is not found in the search tree because the TI does not contain it, the farthest away template pixel is ignored until the reduced data event is found in the tree and the CPMF can be retrieved.

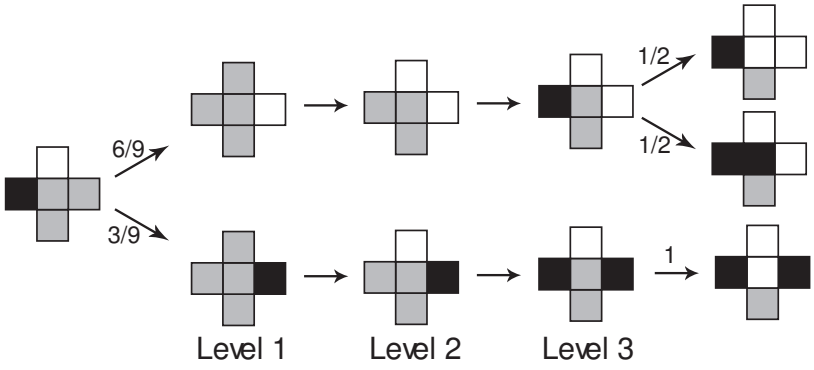


Figure 4.3: Retrieving the conditional probability for class occurrence at the central template pixel given that pixel #2 is white and #3 is black; class labels at pixels #1 and #4 are unknown. The pixel numbering is the same as in Figure 4.1.

4.1.3 The multi-grid approach

The size of the search tree rapidly becomes prohibitive when a large and pattern-rich TI, or a TI with many classes, is scanned with a large template. To alleviate this problem a multi-grid or cascaded simulation approach is typically implemented.

The multi-grid approach consists of expanding the search template, as shown in Figure 4.4, building a new search tree and simulating only at those nodes that belong to that expanded template; the j-th template expansion is performed by considering only 2^{j-1} nodes. Figure 4.4 shows (from right to left) search template expansions for three grid levels, with an example of simulated classes at their respective grid nodes: for the coarsest grid #3 simulated classes are generated at every $2^2 = 4$ nodes, for grid #2 at every second node, and for grid #1 at all remaining nodes. In this way, the multi-grid approach allows simulating large scale patterns that cannot be reproduced by the small template of grid #1.

4.1.4 Improving spatial continuity with multiple passes

The spatial continuity of the realizations can be improved by iterating over the various grids involved in the multi-grid simulation approach. These extra iterations are fast since the search template is full once the first pass is completed. Being full, no branching is required for reading the conditional probabilities from the search tree; see Section 4.1.2.

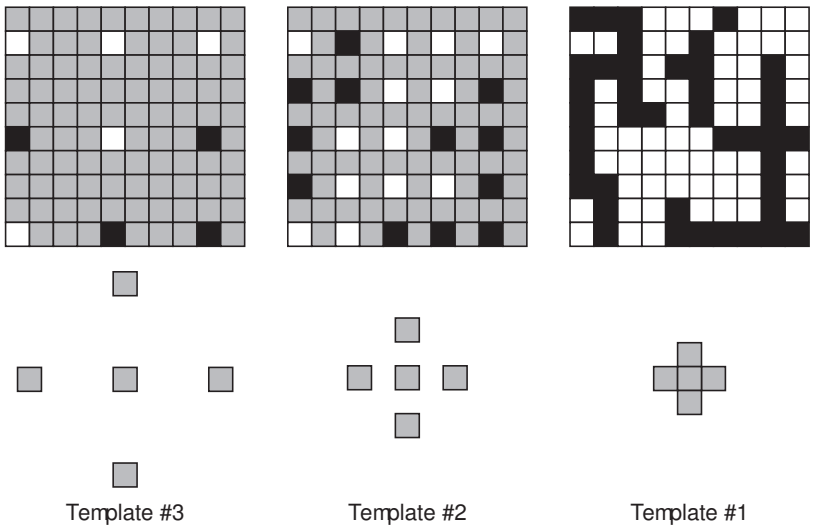


Figure 4.4: Example of simulation using the multi-grid approach.

4.2 Accounting for coarse resolution class fractions

As opposed to the sequential indicator simulation with regularized indicator variograms (Section 3), realizations from the current SNESIM algorithm code could not be conditioned to coarse fraction data. Two non-exclusive methods are thus proposed to integrate these coarse fractions into the SNESIM algorithm. The first method consist of downscaling the coarse fractions using ICK into fine resolution probabilities of class occurrence and then merging these ICK-derived probabilities with the TI-derived local CPMFs. The second method relies on upscaling to the resolution of the sensor measurements the training image into a proportion map, then identify which patterns is associated with which classes of coarse proportions. The first method relies on the commonly used tau model (Journel, 2002) for integration of probabilities. The second method is completely novel.

4.2.1 Method 1 : Downscaling proportion into probability

Regularized variograms offer a convenient and rigorous method to handle information defined on large support, an approach that the current training image-based simulation algorithms do not provide. However, indicator variogram tables computed from the training image can be used to downscale the coarse fractions into fine scale

probability map as shown in Section 3.1.2. But such downscaling does not take full advantage of the patterns found in the training image, since the variogram captures only their two-point statistics which may not be critical in reproducing patterns geometry.

Before starting the SNESIM simulation, the coarse fractions are downscaled into fine scale probabilities. Consider the ICK-derived local CPMF of k-th class occurrence, denoted as $p_k^{IK}(v_m)$, obtained by a direct application of Equation 3.3 and Equation 3.4 given in Chapter 3. The probability map is obtained by building the ICK system of Equation 3.6 with the coarse fraction data as described in Section 3.2. Through that ICK system the coarse proportions are downscaled into a series of fine scale probabilities. For any given variogram model, whether parametric or not, that probability map is unique.

During simulation, at every fine pixel along the random path, a fused local CPMF $p_k^F(v_m)$ is obtained by combining the ICK-derived CMPF $p_k^{IK}(v_m)$ and the TI-derived CPMF $p_k^{TI}(v_m)$ using the tau-model (Journel, 2002; Strebelle, 2002):

$$p_k^F(v) = \left[1 + \left(\frac{1-p_k^{TI}(v_m)}{p_k^{TI}(v_m)}\right)^{\tau_{TI}} \left(\frac{1-p_k^{IK}(v_m)}{p_k^{IK}(v_m)}\right)^{\tau_{IK}} \cdot \left(\frac{1-\pi_k}{\pi_k}\right)^{1-\tau_{TI}-\tau_{IK}}\right]^{-1} \tag{4.1}$$

where the tau exponent τ_{TI} controls integration of the structural information from the training image, and the exponent τ_{IK} that of the ICK-derived probability carrying the information of the coarse fraction classes. A larger exponent for a particular source of information would tend to give it more importance, hence a larger impact on the final results.

These tau exponents can be made to vary from one nested grid to another in the multi-grid simulation approach. For instance at the coarser mulitple grid there is little redundancy between $p_k^{TI}(v_m)$ and $p_k^{IK}(v_m)$. But as simulated data are passed down to the next grid level, these data have already been simulated conditional to the the ICK-derived probability, hence TI-derived probability ($p_k^{TI}(v_m)$) coming from these data is partially redundant with the ICK-derived probability ($p_k^{IK}(v_m)$). These exponents

should be chosen such that to improve reproduction of the prior structural model. Low values $\tau_{IK} < 1$ for the finest grid in the multi-grid simulation approach (see Section 4.1.3) would change less the TI-derived probability, hence providing better structural integrity.

Since the probability $p_k^{IK}(v_m)$ has been derived from the variogram of the training image it may be partially redundant with the training image-derived probability $p_k^{TI}(v_m)$. The tau-exponents in Equation 4.1 aim at correcting such redundancy; for a heuristic determination of the tau parameters see Krishnan et al. (2005).

The resulting downscaled realizations of class labels will only reproduce the coarse fractions on average (in expectation) over many realizations; this is true for both the ICK and SNESIM approaches. If such coarse fractions are themselves uncertain, such an approximate reproduction may be sufficient; otherwise the servo-system of Section 2.2 needs to be implemented to enforce the coarse fractions reproduction on each realization.

Recall that the servo-system merges $p_k^R(v_m)$, the saturation probability of the k-th class occurrence, with the probability $p_k^F(v_m)$ obtained with Equation 4.1. That running probability is function of the number of previously simulated labels of the same class at fine pixels within V_n, and can be viewed as the probability of saturation of coarse pixel V_n in the k-th class.

4.2.2 Method 2 : Partitioning the search tree

Instead on relying on the ICK-derived probabilities for downscaling with SNESIM, a novel approach is proposed to downscale directly from the training image without calling for variograms.

A coarse fraction pixel with high proportion of category k is more likely to have similar patterns with another high coarse proportion pixel than with one with low proportion. Thus, the coarse fraction controls in part the type and extent of the fine scale patterns. Knowing the coarse fraction considerably narrows the type of patterns that should be present at the fine scale. The second method of conditioning SNESIM realization to coarse fraction exploits that relationship between coarse fractions and fine scale patterns. The algorithm is summarized in Algorithm 4.1.

First, the training image is upscaled to the same resolution as the coarse fractions,

Algorithm 4.1 SNESIM algorithm with embricated trees

1: Upscale the training image to the resolution of the observed coarse fraction
2: Partition the upscaled training image into D partition classses $\Theta_d, d = 1, ..., D$ with a clustering algorithm $\psi()$
3: **for** each simulation **do**
4: **for** each multiple grid **do**
5: Build the search tree $T_d, d = 1, ..., D$ for each partition class $\Theta_d, d = 1, ..., D$
6: **for** each pixel v_i within the current multiple-grid **do**
7: Retrieve the data event associated with location v_i
8: Get the partition class $\theta^a(v_i)$ for location v_i
9: Retrieve the conditional probability $p^{TI}(v_i)$ associated with the conditioning data event from the corresponding tree $T_{\theta^a(v_i)}$
10: Merge $p^{TI}(v_i)$ with the ICK-derived probability if necessary
11: Update $p^{TI}(v_i)$ with the servo-system
12: Draw a class label from $p^{TI}(v_i)$
13: Add the simulated class label to the hard data set
14: **end for**
15: **end for**
16: **end for**

see Equation 1.2. At this point we have a dual training image made of the original fine scale TI and of the k-th coarse fraction map, one for each category, derived from that upscaling. That coarse fraction map is partitioned with the algorithm $\psi(\mathbf{a}(V))$ into a set of partition classes $\Theta_d, d = 1, ..., D$, with each class having similar coarse fraction values. Consider that the partition class $\theta^{TI}(v_i)$ of fine scale pixel v_i in the training image is given by

$$\theta^{TI}(v_i) = \psi(\mathbf{a}(V)), \forall v_i \in V \tag{4.2}$$

where $\psi()$ is a clustering algorithm that partition the coarse fractions into D classes. The k-mean algorithm (Hastie et al., 2001) or simply a set of thresholds on the coarse fractions are possible partition algorithms.

It is through these partition classes $\Theta_d, d = 1, ..., D$ that the fine scale patterns are linked to the coarse fraction data. The patterns found within each partition class d are recorded with a specific search tree T_d corresponding to the partition class Θ_d. There is one search T_d tree by partition class Θ_d and $d = 1, ..., D$. Specifically, each

data event centered at location v_i is recorded in its corresponding search tree $T_{\theta^{TI}(v_i)}$. The original large search tree T is therefore replaced by a vector of smaller search trees $T_d, d = 1, ..., D$. Note that any data event found in the TI is still recorded.

Prior to the simulation stage the coarse fractions from the sensor are also partitioned into the same set of partition classes $\Theta_d, d = 1, ..., D$ using the same algorithm. The underlying requirement is that the partition classes of the coarse fraction data corresponds to the partition classes of the training image. Then given a partition class Θ_d the variety of fine scale patterns available is narrowed to those recorded in the smaller search tree T_d.

The simulation is still performed with multiple grids and with a random path. However, at any location v along the path, the partition class $\theta^a(v_i)$ of that location is retrieved to select which search tree $T_{\theta^a(v_i)}$ to use to get the local CPMF. A class label is drawn from that CPMF, then the next node along the path is simulated with its corresponding search tree until all nodes have been visited. Given a partition class, the fine scale patterns provided by the corresponding search tree are those found in the TI for that class of coarse fractions.

When building the D search trees and when retrieving the CPMF from them, it is essential to let the data event extend over the neighboring partition classes. This overlap ensures that all data events found in the training image are recorded. This overlap also ensures a smooth transition between partition classes when simulating. Given that no single partition tree contains all the patterns, the full variety of patterns found in the TI can only by restituted by spatially linking the trees together. Hence, the spatial distribution of the partition classes over the simulation domain and over the training image must be similar.

In addition to allowing conditioning of patterns to coarse fractions this search tree partitioning brings two major improvements to the classical SNESIM algorithm :

- It provides robustness with regard to departure from stationarity, see hereafter
- It considerably increases the simulation speed.

Take a large analog (training) image such as the Foshan city of Figure 3.14(a), that image is non-stationary with trend and spatially varying anisotropy and proportions. Were that entire image used as TI, the SNESIM algorithm would build a single search

tree T pooling together all patterns ignoring their relative locations. In addition to being large, hence slow to build and read, such search tree will generate simulations with patterns that may be strikingly different from those found in the training image.

To demonstrate the robustness brought by partitioning as to departure from the assumption of stationarity, take the city of Foshan as a training image. Two unconditional simulations with a single global tree are shown in Figure 4.5. The realizations from the single tree do not reproduce the features of the training image. By pooling all patterns together into a single search tree, the characteristics of the training image are lost, such as the transitioning from dense urban core to lighter texture.

Use now the partitioning approach. First, the training image is upscaled to coarse pixels each of size $15\times15=225$ fine scale pixels. The upscaled training image of Foshan is shown in Figure 4.6(a). The partitioning is performed with a k-mean algorithm into the 5 partition classes displayed in Figure 4.6(b). Each of the corresponding search tree by itself is not meaningful; one unconditional simulation for each tree is generated and shown in Figure 4.7. None of these simulated images are similar to the full TI of Figure 3.14(a). However each contains some patterns from it.

Two fields of partition class with the correct autocorrelation are simulated with the classical SNESIM algorithm using Figure 4.6(b) as training image. Two partition class realizations are shown in Figure 4.8(a) and (b). From each of these simulated partition classes and using the Foshan TI, one realization is generated and shown in Figure 4.8(c) and (d). These realizations generated with search tree partioning are more similar in texture to the training image than the realizations obtained from a global search tree, for the latter see Figure 4.5. They display the dense cores of urban area that were not present when simulating with the global tree.

If the simulation is repeated with partition classes randomly distributed in space, the texture of the training image is not reproduced. Figure 4.9 shows two realizations usin random partition classes and the resulting fine scale realizations. The unstructured partitions do not generate texture similar to those of Figure 4.8 or those of the training image of Figure 3.14(a). The success of proposed algorithm requires that the partition classes over the simulation field and over the training image have similar spatial distribution. Only then it is possible to patch together the patterns of Figure 4.7 into the texture of Figure 3.14(a) and not ending with the random texture of

Figure 4.9.

Although the training image is the most critical input specifying the patterns to be reproduced, other simulation parameters mostly the size of the search template and the number of multiple grids greatly affect the final output.

Finally, simulating with smaller trees is faster. With a smaller tree there is less branching to visit when data events are incomplete, see Section 4.1.2. The simulation of Figure 4.8(c) and (d) using the partition class have been generated up to 4 times faster than the one using a single global tree in Figure 4.5 (approximately 5 minutes versus 20 minutes on an IBM laptop with 1.7 GHz processor).

In summary, partitioning the global search tree based on coarse fraction provides a conditioning on large support data which is robust from stationarity departure, yet a faster algorithm.

4.3 Conditioning to fine scale data

The multiple-grid concept requires special consideration when local fine scale data are present. The standard implementation of data conditioning in SNESIM in presence of multiple-grid calls for moving the conditioning points onto the current grid being simulated. While this re-allocation may be satisfactory when few data are available, it generates artifacts in presence of a large number of conditioning data or when these data have specific geometry such as lineation, which may be lost through re-allocation. With coarse proportion data, the re-allocation approach may cause conflicts when a hard data is moved into a neighboring coarse pixel. For instance, a fine scale hard data $j_k(v) = 1 \in V$ that is reallocated within coarse pixel V' where $a_k(V') = 0$ creates a conflict.

To reflect the geometry of data and accomodate a large number of fine scale hard data, it is suggested not to use the re-allocation scheme but to condition to the hard data through a probability field $p_k^{IK}(v_m)$ generated by ICK, see Section 3.2. The fine scale hard data remain at their actual locations and if they happen to be on the current multiple grid they become part of the conditioning data event. When the fine scale hard data is outside the current multiple grid the ICK-derived probability accounts for the proximity of that hard data. The integration of probability $p_k^{IK}(v_m)$

is done with the tau model as shown in Equation 4.1.

4.4 Case studies

Consider the previous examples used for the variogram-based algorithm of Section 3.4. The difference now is that the structural model is given through a training image instead of variogram models. The training image for the Pearl River Delta (PRD) example remains the image given in Figure 3.1. The training image for the Guangzhou (GZ) example is the analog city of Foshan given in Figure 4.14.

The SNESIM algorithm is applied for downscaling on these two case studies. The focus is first on the downscaling of the coarse fractions for each case study using the full algorithm, that is with the ICK-derived probability (Section 4.2.1), the partitioning of the search tree (Section 4.2.2) and the servo-system (Section 2.2). In the remainder of the section the efficiency of the two proposed methods at integrating the coarse fractions is investigated.

4.4.1 The Pearl River Delta example

Refer to Section 3.4.1 for a detailed explanation of the PRD data set. As with the variogram-based case study, the true fine scale map is used as training image to allow focusing only on the algorithm performance. The same three coarse spatial resolutions of Section 3.4.1 are considered and evaluated in this section.

With a k-mean algorithm the upscaled training images of Figure 3.2 are processed into 8 partition classes for the 9×9 coarse resolution and 6 partition classes for the 15×15 and 25×25 coarse resolutions. The distribution of the resulting partition classes is shown in Figure 4.10. The k-mean algorithm is an unsupervised algorithm that classifies similar obervations into classes. Recall that the patterns within each partition class is recorded in a specific search tree, hence the patterns of the training image (Figure 3.1) are recorded with six (or eight) search trees, one for each partition class. Each search tree contain different patterns that can be visualized by performing unconditional simulations. For the 15×15 upscaling case the structural model implicit to each partition class is displayed in Figure 4.11 through an unconditional simulation. The patterns seen in each of these realizations while taken from the full training image

(Figure 3.1) are not, by themselves, representative of it. However it is shown that each partition class narrows the variety of patterns to be generated. The patterns found in the partition class #1 (Figure 4.11(a)) which are a mix of the three land covers are different from those of partition class #4 (Figure 4.11(d)) which mostly model the dense vegetation, also different from those of partition class #6 (Figure 4.11(f)) which contains the dense urban patterns. The patterns seen in the full training image can only be reconstituted by taking into consideration the spatial autocorrelation of the coarse fractions or, similarly of the partition classes, respectively shown in Figure 3.2 and Figure 4.10.

The vector of search trees built from the partition classes in Figure 4.10 is now used in conjunction with the ICK-derived probability map shown in Figure 3.3, and with the servo-system of Section 2.2 to downscale the coarse fractions. This is done for all three upscaled resolutions of Figure 3.2. The algorithm used three multiple grids and a search template of 60 pixels. The tau exponent τ_{IK} in Equation 4.1 for the integration of the ICK-derived probabilities and the tau exponent τ_R in Equation 2.6 used by the servo-system vary with the multiple grid, they are given in Table 4.1. For the coarse multiple grid the servo-system is kept to a minimum with a tau exponenent τ_R set to 0.01 while the ICK-derived probability is given more weight with $\tau_{IK} = 1$. As the simulation progresses, more weight is given to the coarse fraction constraint, that is τ_R increases. At the same time the tau-exponent τ_{IK} for the ICK-derived probability decreases and is ultimately ignored at the fine grid where $\tau_{IK} = 0$. At this point the information carried by the ICK-derived probability is highly redundant with the one given by the training image since half the grid has already been simulated conditional to the ICK probability. The structural information carried by the ICK-derived probability is then carried by the simulated class labels. Recall that whenever the servo-system is used, i.e. where $\tau_R \neq 0$, the coarse fraction is exactly reproduced for all simulations, whatever the other tau-exponents may be.

Two realizations for each coarse spatial resolution are shown in Figure 4.12. As with the variogram-based simulations in Figure 3.18, and compared to the estimation map in Figure 3.3, the generated patterns do not significantly differs when the coarse fraction resolution changes. There is no major discontinuities across coarse pixel boundaries, although some block artifacts are visible for the 25×25 upscaling

resolution (Figure 4.12(c) and (f)). There is, however, evident speckles (noise) in the images, more so than with the variogram-based algorithm. This suggests that the integration of the ICK-derived probability and the partitioning is not as efficient than the direct and explicit integration of the coarse fraction through regularized variograms. The ICK approach has the advantage to directly integrate the surrounding coarse pixel data into the conditional probability. This probability changes as the simulation progresses and class labels are being simulated. With the SNESIM approach, the neighboring coarse fraction are integrated through the ICK-derived probabilities. However, these probabilities are calculated prior to the simulation and remain the same throughout the sequential simulation, regardless of the simulated class label.

Keeping the same parameters with the servo-system disabled, two simulations are generated for each upscaling scheme and are shown in Figure 4.13. With the servo-system disabled, the speckle artifacts have disappeared, but the coarse fractions are reproduced only approximately. The coefficient of correlations between the simulated and the actual coarse fractions are 0.91, 0.90 and 0.87 for the 9×9, the 15×15 and the 25×25 resolution respectively. The reproduction slightly improves with smaller coarse fraction pixels. Recall that with the servo-system the coefficient of correlation is always one.

4.4.2 The Guangzhou example

The training image approach for downscaling is next tested on the Guangzhou (GZ) example, see Section 3.4.2 for details about the data set. The analog image of Foshan (Figure 3.14(a)) is used as training image for the downscaling of the coarse fractions of Figure 3.13(a). This data set provides different types of conditioning data at the fine spatial resolution, these data are the river and roads networks and the previously mapped impervious areas. As discussed in Section 4.3, these hard data are not re-allocated on the multiple-grid as with the classical SNESIM algorithm, instead they are integrated using the ICK-derived probabilities of Figure 3.15 obtained with the variogram table calculated from the Foshan analog, see Section 3.4.2.

The upscaling and partitioning of the Foshan training image has already been done in Section 4.2.2 and remain the same here. The upscaled training image and

its five partition classes are shown in Figure 4.6(a) and Figure 4.6(b). The same partitioning rule is applied to the coarse fraction map of Figure 3.13(a) yielding the partition classes shown in Figure 4.14(a). Note the similarity between the spatial distribution of the partitions between the training image of Figure 4.6(b) and those of the observation data in Figure 4.14(a). Both consist of a core of the same partition class surrounded by the same other partition classes. As shown in Section 4.2.2, that similarity is needed to yield realizations with the correct spatial distribution.

For each of the four fine scale hard data set (no hard data, water only (Figure 3.13(b)), water and road (Figure 3.13(c)) and previously mapped impervious area (Figure 3.13(d))) two downscaling realizations are generated and displayed in Figure 4.15. The coarse fraction are integrated with the ICK-derived probability shown in Figure 3.15, the partition classes of Figure 4.14(a) and the servo-system.

Note how the impervious shapes become crisper when more fine scale data are available. The conditioning through the ICK-derived probability, explained in Section 4.3 is efficient in that it has reproduced the hard data without discontinuity. For instance the road can clearly be seen on Figure 4.15(e) to (h) and there is no artifact around these fine scale hard data. When compared to the downscaling of the PRD coarse fractions (Figure 4.12), the downscaled maps of Figure 4.15 are less noisy. The speckles come from the servo-system imposing labels in order to enforce exact reproduction of the coarse fractions. The GZ data set being only binary (impervious vs non-impervious) as compared to three categories for the PRD data set, the servo-system has fewer categories to enforce hence less opportunity to create discontinuities. Note that all the simulations of Figure 4.15 exactly reproduce the coarse fractions per the servo-system.

The downscaling is repeated for each hard data scenarios but now without the servo-system. The realizations are shown in Figure 4.16. These realizations have a significantly lower noise level than those in Figure 4.12 where the servo-system is in service. The upscaled coarse fractions from the realizations show a coefficient of correlation of 0.94 with the true coarse fractions for all fine scale data scenarios. This is surprising since the fine scale data set of Figure 3.13(d) covers more than 28% of the domain, including all the dense urban center, but generates the same fraction reproduction accuracy than when no fine scale data are available. Adding fine scale

data improves the texture of the realization but cannot replace the servo-system to ensure coarse fraction reproduction.

4.4.3 Sensitivity analysis

The remaining of this chapter investigates the effectiveness and sensitivity of the ICK-derived probability (Section 4.2.1) and the search tree partitioning (Section 4.2.2), two algorithms proposed to integrate coarse fraction data. The goal is to look simulataneously at the coarse fraction reproduction and at the reproduction of training image texture when using either algorithm. The servo-system remains disabled.

First the effect of the number of partition classes is investigated. The PRD coarse fraction map for the 15×15 upscaling resolution (Section 4.4.1) is partitioned six times into partition classes, each time with an increasing number of partition classes. The partitioning is done with the k-mean algorithm, All partitioning are shown in Figure 4.17. The first partition, shown in Figure 4.17(a) has three partition classes, the second in Figure 4.17(b) four partition classes, and the last one in Figure 4.17(f) has 10 partition classes. By increasing number of classes from 3 to 10, the partition classes get more precise in identifying specific texture, such as the dense urban core or the transition between urban and vegetation.

Figure 4.18 shows one realization for each of the six partition schemes of Figure 4.17. Only these partition classes are used for downscaling; the ICK-derived probability and the servo-system are not used. Without the servo-system most speckles disappear. The coarse fraction reproduction is validated with a scatter plot between the upscaled proportion from the simulation and the true coarse fraction data derived from the sensor (shown in Figure 3.2(d)). Figure 4.19 shows the coarse fraction reproduction for the vegetation land cover only. A perfect reproduction would generate points on the 45° line; the higher the coefficient of correlation the better the reproduction. A consistent and significant improvement of the coefficient of correlation from 0.72 to 0.90 is observed going from the 3-class partition to the 10-class partition.

The time taken to compute the simulations consistently decreases when the number of partition classes increases. One realization with the 3-partition classes took 16 minutes while with the 10-partition classes it tooks only 8 minutes. More partition classes produces smaller trees which are faster to search than a single global tree.

Overall, increasing the number of partition class increase the accuracy and the speed of the simulations.

An alternative algorithm is to consider only the ICK-derived probability without partitioning the search tree. Figure 4.20(a) shows a simulation where partitioning and the servo-system are not used, the coarse fractions are integrated only through the ICK-derived probability. The coarse fraction reproduction, shown in Figure 4.20(b), is similar to that obtained with the 10-partition classes with a coefficient of correlation of 0.91. However that simulation takes 35 minutes of processing time, four time slower than when using the 10-class partition.

Comparing realizations from the 10-class partition of Figure 4.18(f) realization generated with only the ICK-derived probability of Figure 4.20(a), the reference image in Figure 3.1, it appears that the partitioning provides more accurate texture. Conditioning with the ICK-derived probability alone generates fine scale maps that are too smooth without the fragmentation seen in the training image. The accurate reproduction of spatial patterns is also clearly seen with experimental variograms. Figure 4.21 shows the vegetation label experimental variograms for the reference image, for the 10-partition classes and the ICK-derived probability realizations. The experimental variograms for the search tree partitioning match very closely the reference ones while the experimental variograms from the ICK-derived probability is too continuous.

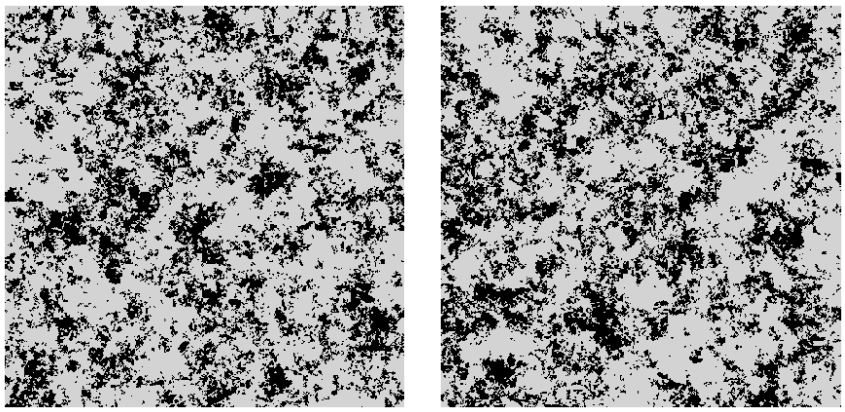
The comparison between conditioning with the ICK-derived probability and with the search tree partitioning is repeated for the GZ case study (Section 4.4.2). Figure 4.22(a) and (b) show one realization without the servo-system, using respectively the partition of Figure 4.14(a) and the ICK-derived probability of Figure 3.15 to condition to the coarse fraction. As with the PRD case study, the realization generated with the ICK-derived probability only is too smooth and lacks the texture fragmentation seen in the training image (Figure 3.14(a)) and in the reference image (Figure 3.12). The partitioning approach provides better texture reproduction.

The coarse fraction reproduction of Figure 4.22(a) and (b) is analyzed in Figure 4.23. In this case both methods deliver a coefficient of correlation of 0.92, as shown in Figure 4.23(b) and (d). However, the partitioning option, with its smaller search trees, cuts the simulation time by a factor of three from 35 to 12 minutes. By

conditioning both with the ICK-derived probability (using unit tau-exponent value) and with the partition classes, shown in Figure 4.16(a), the coarse fraction reproduction increases to a coefficient of correlation of 0.96 (Figure 4.23(f)) and only takes 12 minutes of computing time. Using both the search tree partitioning and the ICK-derived probability methods provides a faster and more accurate algorithm.

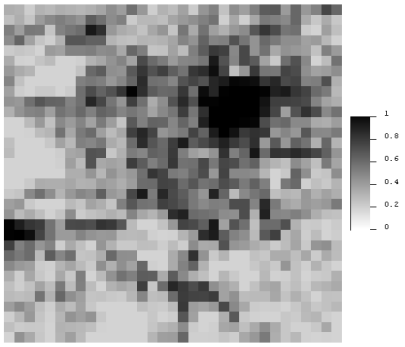
Finally, the spatial location of the errors on the coarse fraction reproduction of Figure 4.22(a) and (b) and (c) can be seen in Figure 4.23(a), (c) and (e). Notice that despite reproducing the coarse fraction equally well, the errors generated by the partitioning approach (Figure 4.23(a)) are not spatially correlated as opposed with ICK-derived probability ((Figure 4.23(a))). The latter approach generates errors mostly in the transition between the core urban center and its surroundings.

In summary, it is recommended to use simultaneously the partition approach and the ICK-derived probability to integrate coarse fraction data. Combined together, they generate realizations which are more accurate and faster than when only one of the method is used. The ICK-derived probability is especially useful in region with very high or low coarse fraction value, which yield ICK-derived probability close to 0 ot 1, where it provides strong conditioning. The partitioning always increases the simulation speed and is also significant in increasing the coarse fraction reproduction.

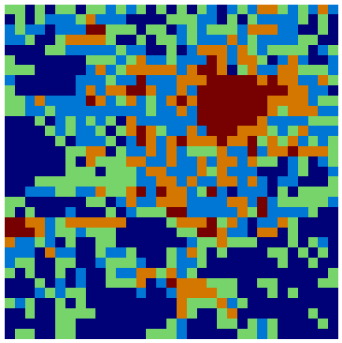


(a) Realization #1 (b) Realization #2

Figure 4.5: Unconditional simulations from a single search tree using Figure 3.14(a) for training image. The scene is 495 × 495 pixels (≈ 30km × 30km). Black color indicates impervious, white non-impervious.

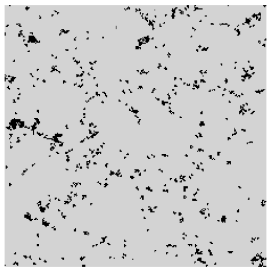


(a) Upscaling of the training image. Black indicates higher proportion of impervious area.

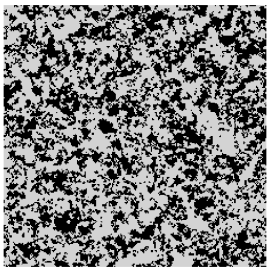


(b) Partition classes derived from (a) with a k-mean algorithm

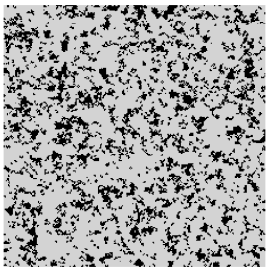
Figure 4.6: Upscaling and partitioning of the Foshan training image of Figure 4.14. A coarse pixel contains $15 \times 15 = 225$ fine scale pixels. The scene is 33×33 pixels ($\approx$ 30km $\times$ 30km).



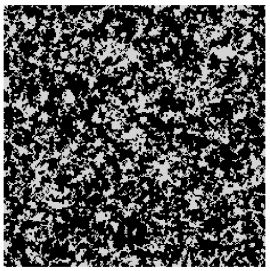
(a) Partition #1



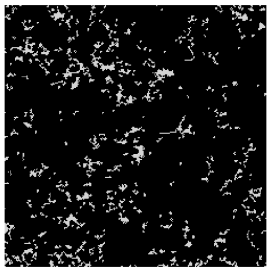
(b) Partition #2



(c) Partition #3

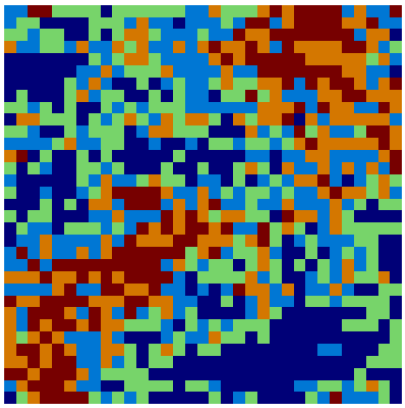


(d) Partition #4

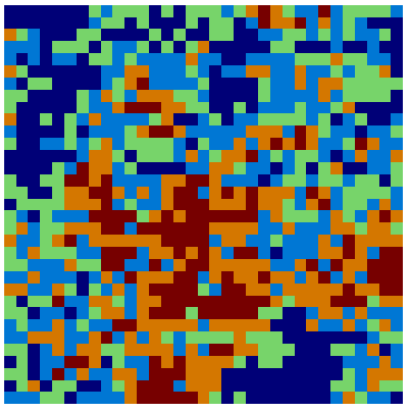


(e) Partition #5

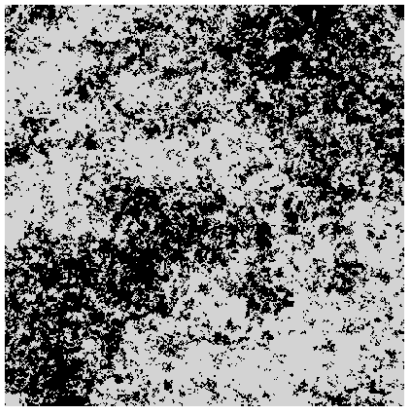
Figure 4.7: Display of the structural model implicit to each search tree derived from the k-mean partition of Figure 4.6(b). The scene is 495×495 pixels ($\approx$ 30km $\times$ 30km). Black color indicates impervious, white non-impervious.



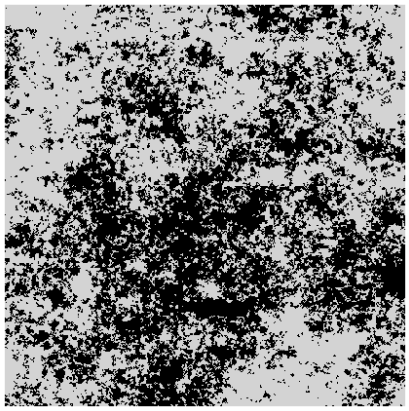
(a) Simulated partition class, realization #1. Each color represent a partition class



(b) Simulated partition class, realization #2. Each color represent a partition class

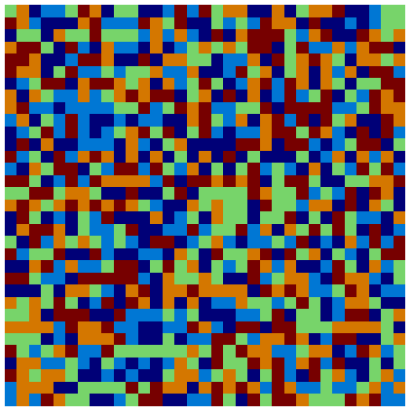


(c) SNESIM simulation from partition class in (a)

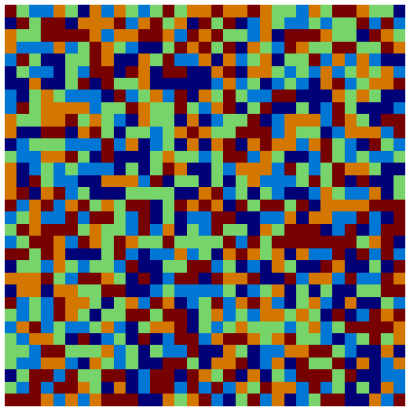


(d) SNESIM simulation from partition class in (b)

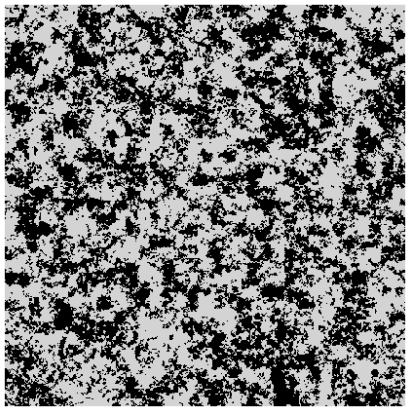
Figure 4.8: Unconditional simulations with partition of the search tree. The partition classes are simulated using SNESIM in (a) and (b) using Figure 4.6(b) as training image. These simulated classes are then used for a fine scale simulations shown in (c) and (d) using Figure 3.14(a) as training image. The scene is 495×495 pixels ($\approx$ 30km $\times$ 30km). Black color indicates impervious, white non-impervious.



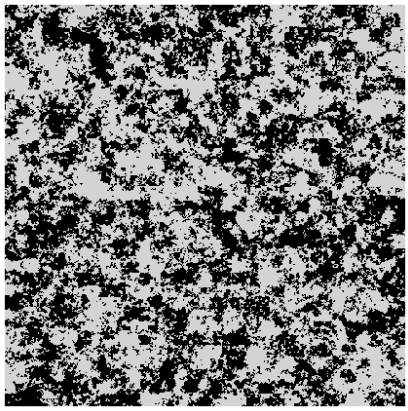
(a) Random partition class, realization #1



(b) Random partition class, realization #2



(c) SNESIM simulation from partition class in (a)

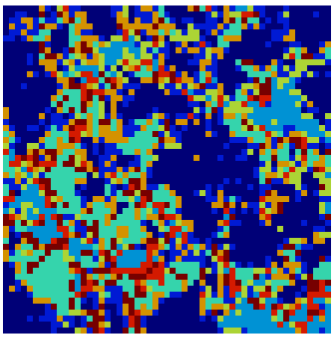


(d) SNESIM simulation from partition class in (b)

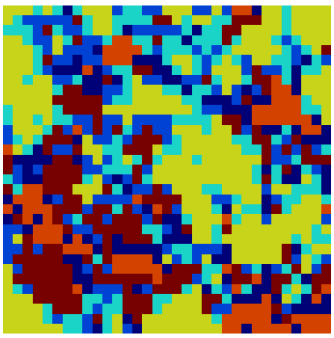
Figure 4.9: Unconditional simulations with search tree partitioning. The partition classes (a) and (b) are randomly distributed in space. These simulated classes are used for fine scale simulation in (c) and (d) using Figure 3.14(a) as training image. The scene is 495 $\times$ 495 pixels ($\approx$ 30km $\times$ 30km). Black color indicates impervious, white non-impervious.

Nested grid	τ_{TI}	τ_{IK}	τ_R
# 3 (coarse)	1	1	0.01
# 2	1	0.5	0.2
# 3 (fine).	1	0	0.5

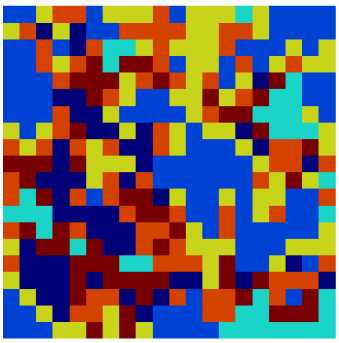
Table 4.1: Tau exponent values for each nested grid.



(a) K-mean partition for 9×9 resolution. There are 8 partition classes. The scene is 55×55 pixels.

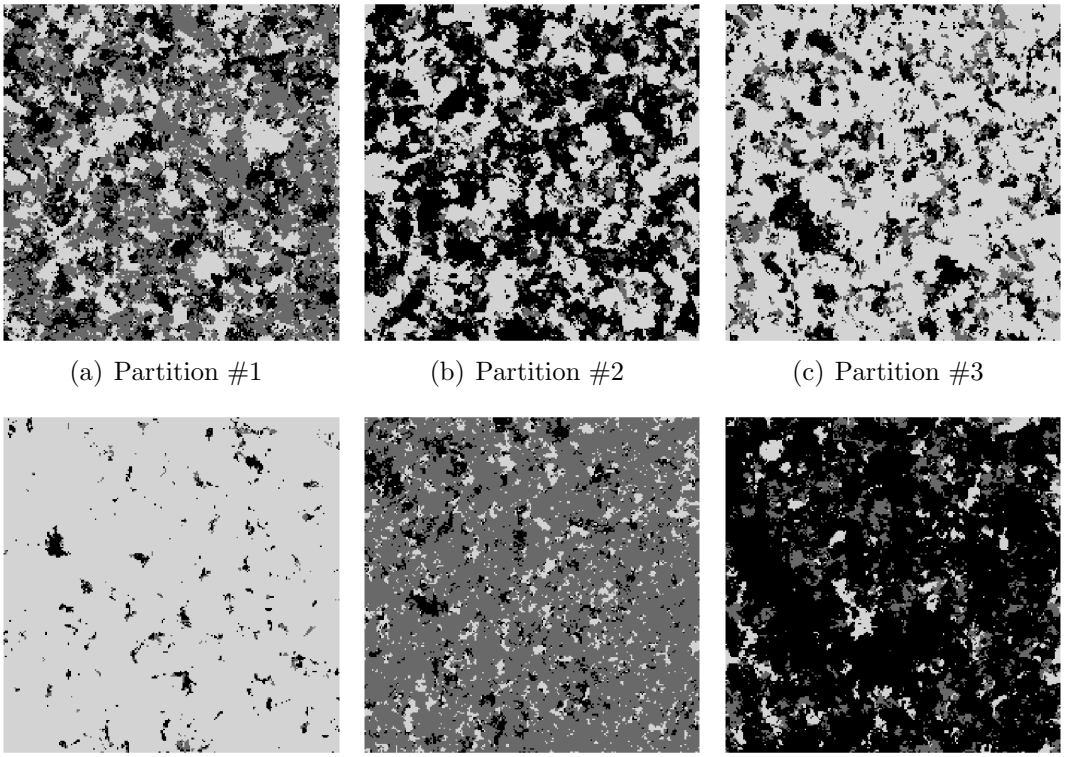


(b) K-mean partition for 15×15 resolution. There are 6 partition classes. The scene is 33×33 pixels.



(c) K-mean partition for 25×25 resolution. There are 6 partition classes. The scene is 20×20 pixels.

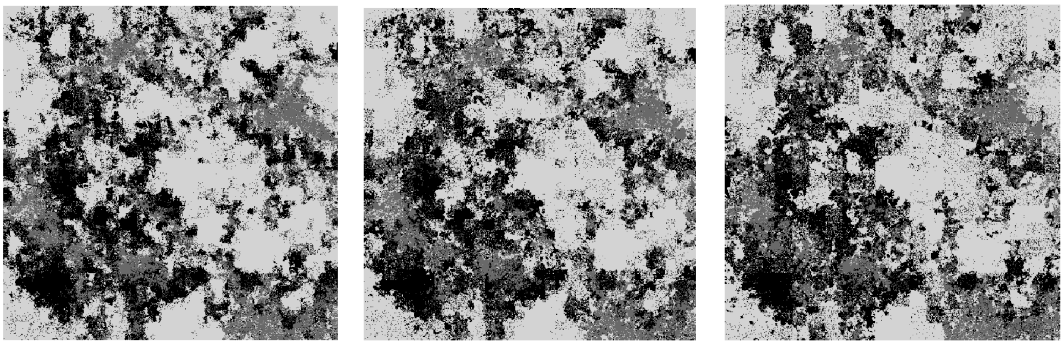
Figure 4.10: K-mean partition from the coarse fractions of the PRD data set (Figure 3.2). The scene is $\approx$ 15km $\times$ 15km.



(a) Partition #1 (b) Partition #2 (c) Partition #3

(d) Partition #4 (e) Partition #5 (f) Partition #6

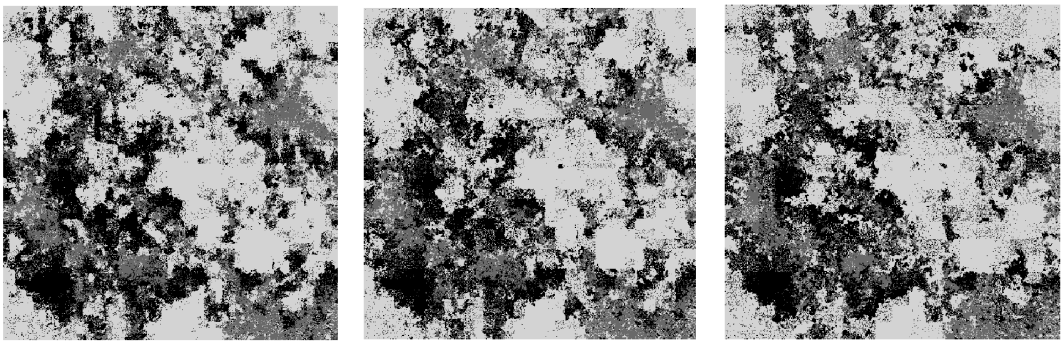
Figure 4.11: Implicit structural model for each partition shown with an unconditional simulation on a 225×225 grid. White color indicates vegetation, gray urban, black bare soil.



(a) Realization #1; 9×9 resolution.

(b) Realization #1; 15×15 resolution.

(c) Realization #1; 25×25 resolution.

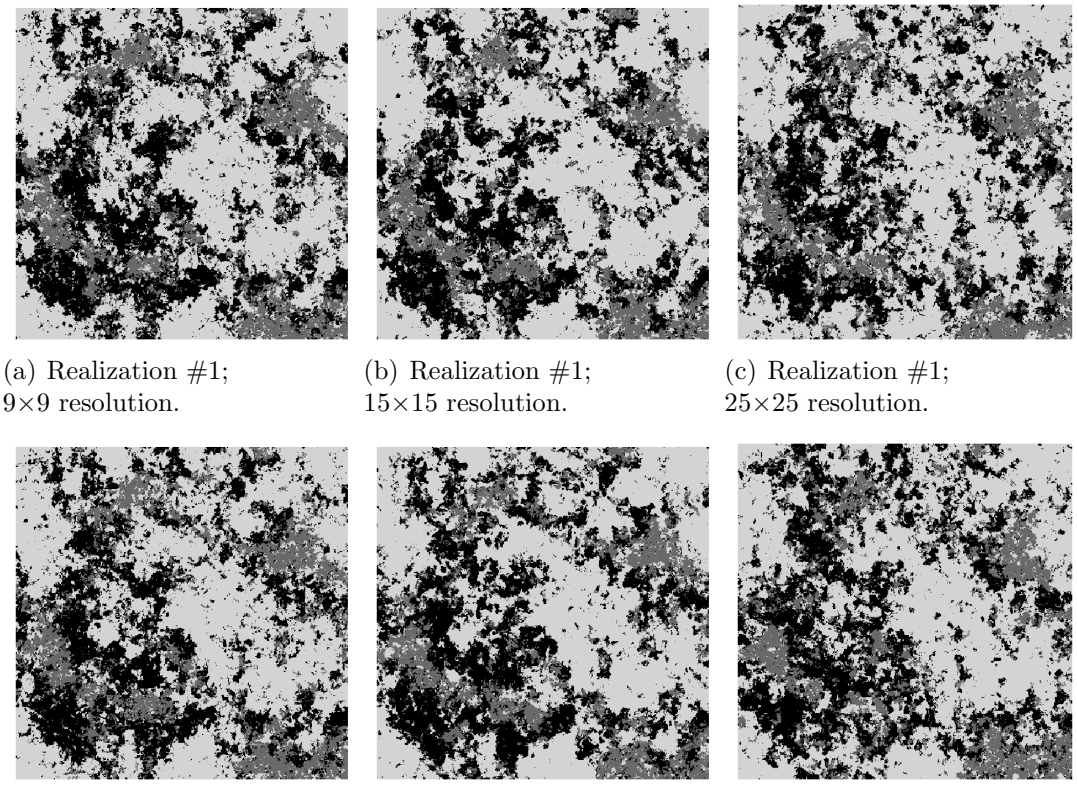


(d) Realization #2; 9×9 resolution.

(e) Realization #2; 15×15 resolution.

(f) Realization #2; 25×25 resolution.

Figure 4.12: Downscaling of the PRD coarse fractions (Figure 3.2) with search tree partitioning using the ICK-derived probability and the servo-system. The scene is 495×495 pixels ($\approx$ 15km $\times$ 15km). White color indicates vegetation, gray urban, black bare soil.



(a) Realization #1; 9×9 resolution.

(b) Realization #1; 15×15 resolution.

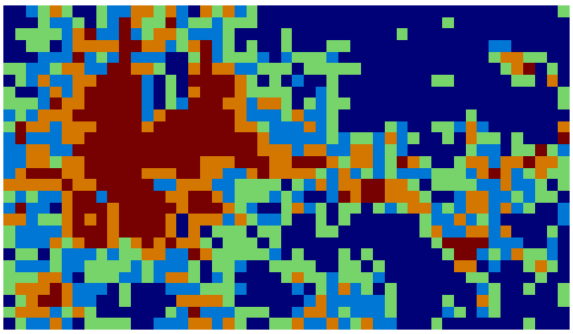
(c) Realization #1; 25×25 resolution.

(d) Realization #2; 9×9 resolution.

(e) Realization #2; 15×15 resolution.

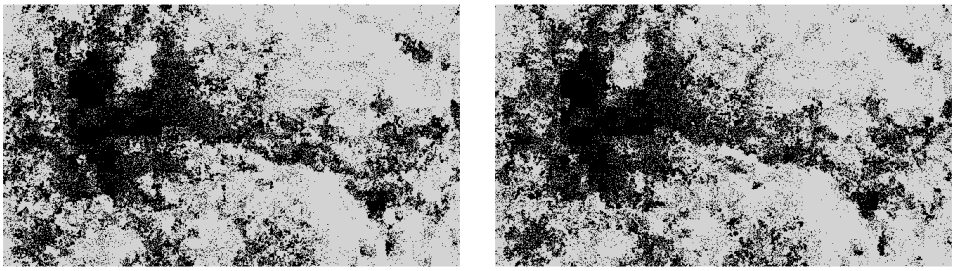
(f) Realization #2; 25×25 resolution.

Figure 4.13: Downscaling of the PRD coarse fractions (Figure 3.2) with search tree partitioning using the ICK-derived probability but without the servo-system. The scene is 495 × 495 pixels (≈ 15km × 15km). White color indicates vegetation, gray urban, black bare soil.



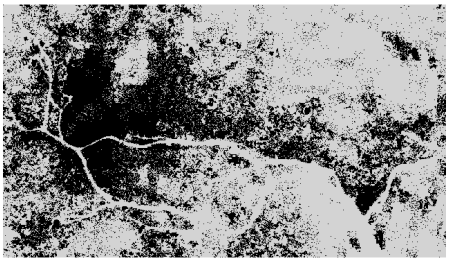
(a) Partitioning of the coarse fraction map

Figure 4.14: Partitioning of the coarse fraction map for the GZ case study. Note the similarity in the partition classes spatial distribution between this image and the training image of Figure 4.6(b). The scene is 49×28 pixels ($\approx$ 44km $\times$ 25km).



(a) Simulation #1 No hard data

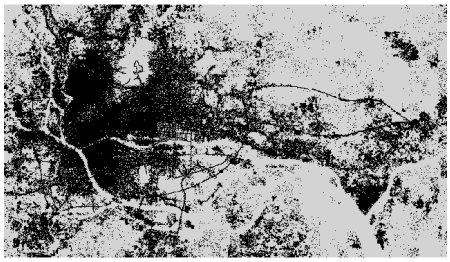
(b) Simulation #2 No hard data



(c) Simulation #1 Water data

(d) Simulation #2 Water data

(e) Simulation #1 Road data

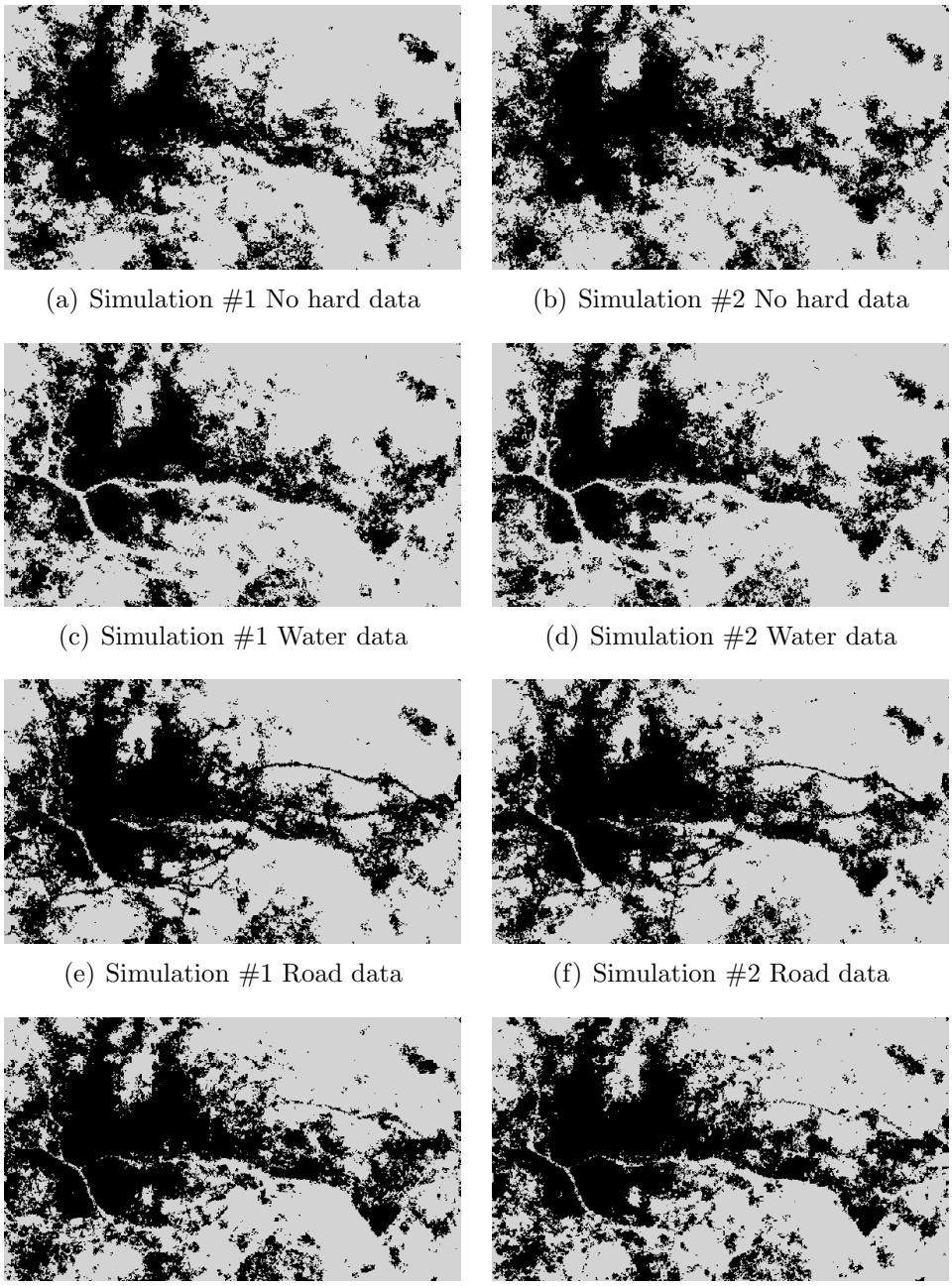


(f) Simulation #2 Road data

(g) Simulation #1 Past data

(h) Simulation #2 Past data

Figure 4.15: Downscaling of of the GZ coarse fraction of Figure 3.13(a) conditioned to the different data sets shown in Figure 3.13. Partition classes and ICK-derived probability and servo-system were used. The scene is 735×420 pixels ($\approx$ 44km $\times$ 25km). Black color indicates impervious, white non-impervious.



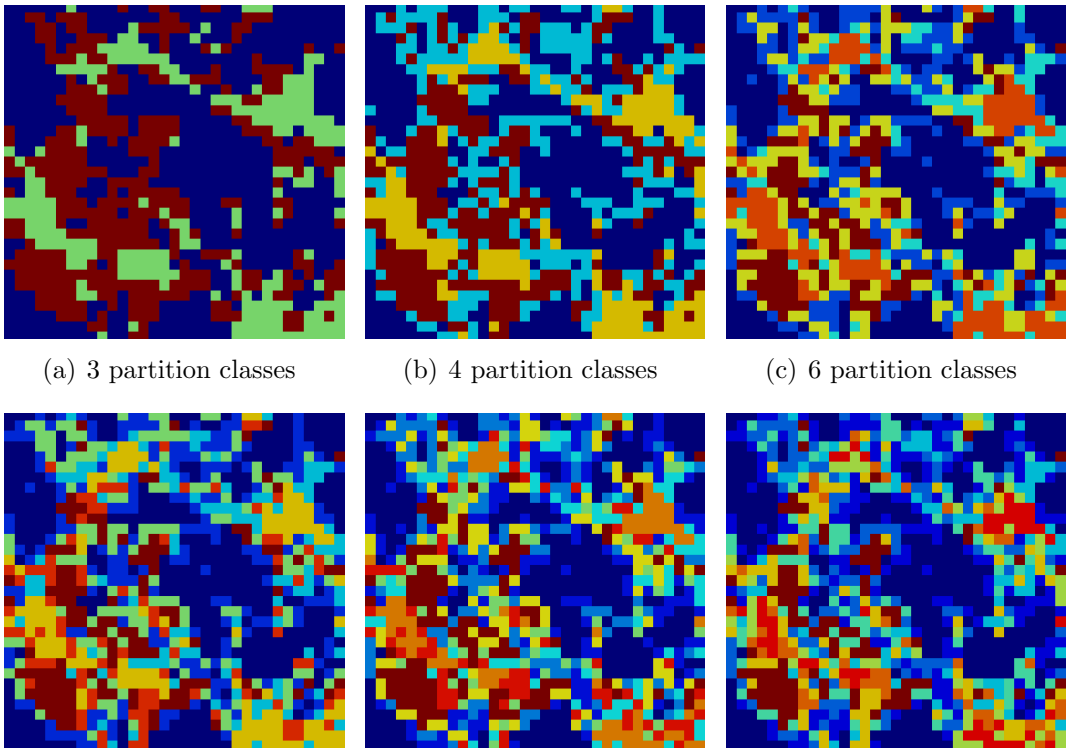
(a) Simulation #1 No hard data (b) Simulation #2 No hard data

(c) Simulation #1 Water data (d) Simulation #2 Water data

(e) Simulation #1 Road data (f) Simulation #2 Road data

(g) Simulation #1 Past data (h) Simulation #2 Past data

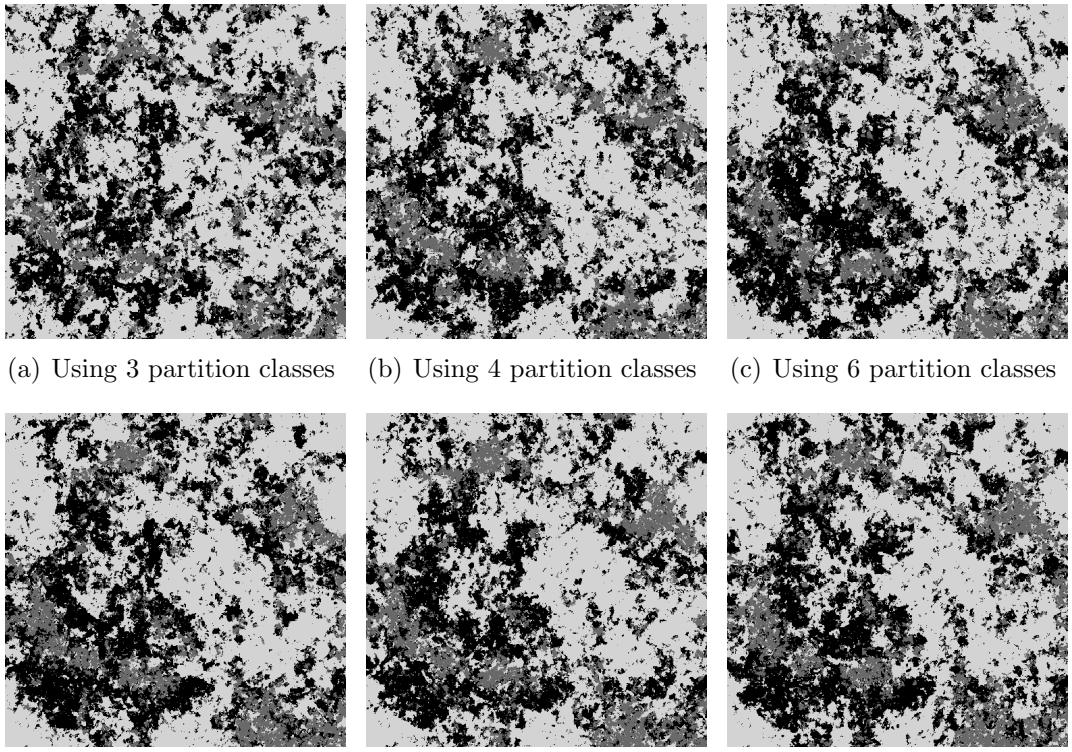
Figure 4.16: Downscaling of of the GZ coarse fraction of Figure 3.13(a) conditioned to the different data sets shown in Figure 3.13. Partition classes and ICK-derived probability were used but without the servo-system. The scene is 735×420 pixels ($\approx$ 44km $\times$ 25km). Black color indicates impervious, white non-impervious.



(a) 3 partition classes (b) 4 partition classes (c) 6 partition classes

(d) 7 partition classes (e) 9 partition classes (f) 10 partition classes

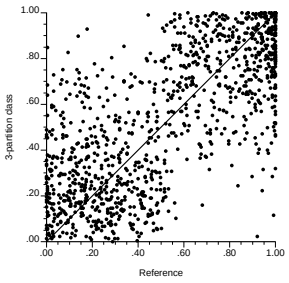
Figure 4.17: Six partitioning of the 15×15 upscaling resolution of PRD TI (Figure 3.1) with the k-mean algorithm The scene is 33×33 pixels ($\approx$ 15km $\times$ 15km).



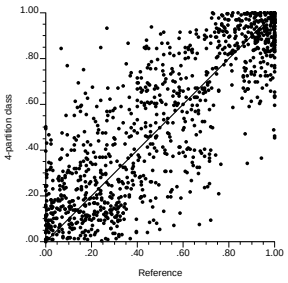
(a) Using 3 partition classes (b) Using 4 partition classes (c) Using 6 partition classes

(d) Using 7 partition classes (e) Using 9 partition classes (f) Using 10 partition classes

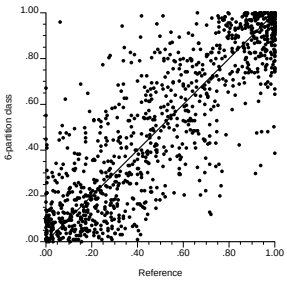
Figure 4.18: Downscaling with each of the six set of partition classes of Figure 4.17. The scene is 495×495 pixels ($\approx$ 15km $\times$ 15km). White color indicates vegetation, gray urban, black bare soil.



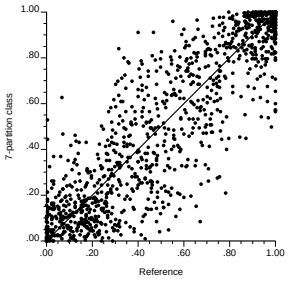
(a) 3 classes: $\rho = 0.72$



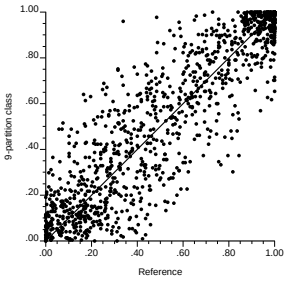
(b) 4 classes: $\rho = 0.83$



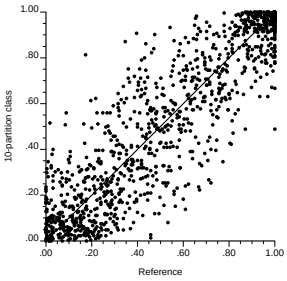
(c) 6 classes: $\rho = 0.86$



(d) 7 classes: $\rho = 0.89$

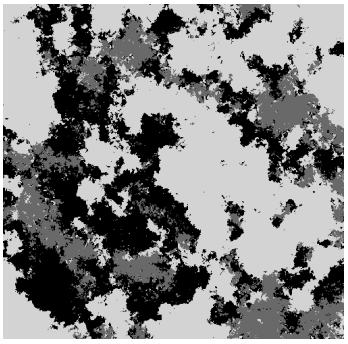


(e) 9 classes: $\rho = 0.90$

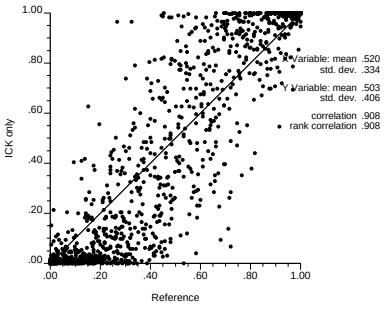


(f) 10 classes: $\rho = 0.90$

Figure 4.19: Coarse fraction reproduction when using partition classes



(a) Downscaling with the ICK-derived probability (no partition)



(b) Coarse fraction reproduction: $\rho = 0.91$

Figure 4.20: Downscaling using only the ICK-derived probability. Corresponding coarse fraction reproduction. The scene is 495×495 pixels ($\approx$ 15km $\times$ 15km). White color indicates vegetation, gray urban, black bare soil.

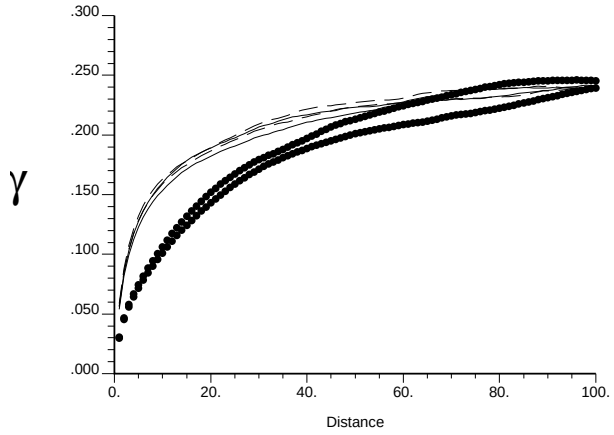
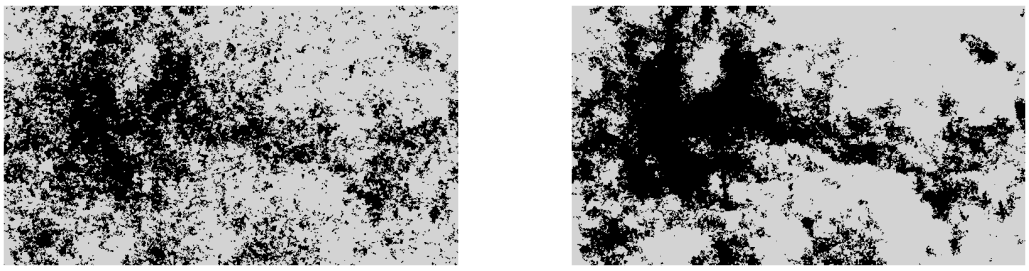


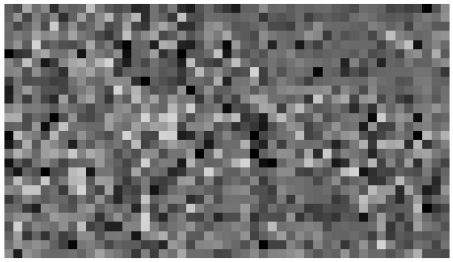
Figure 4.21: Experimental variograms for the vegetation land cover with the solid lines for the reference image of Figure 3.1, the dashed lines for the 10-partition classes of Figure 4.18(f) and the dots for the ICK-derived probability of Figure 4.20(a).



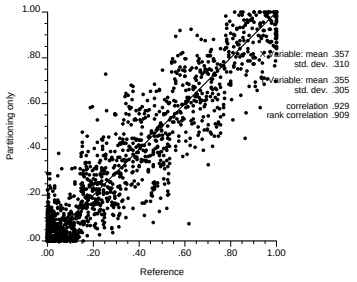
(a) Simulation using partition classes

(b) Simulation using ICK-derived probability

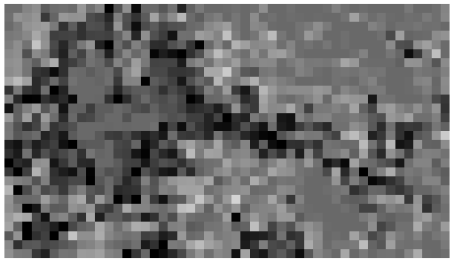
Figure 4.22: Simulation of the GZ case without servo-system. In (a) the conditioning is done with the partition classes while in (b) it is done with the ICK-derived probability. The scene is 735 × 420 pixels (≈ 44km × 25km). Black color indicates impervious, white non-impervious.



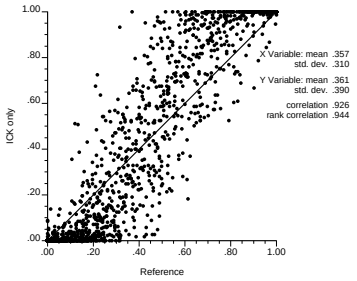
(a) Coarse fraction errors when simulating only with partition classes



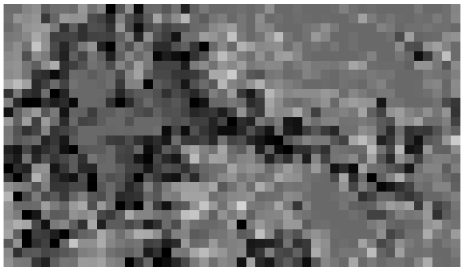
(b) Coarse fraction reproduction when simulating only with partition classes



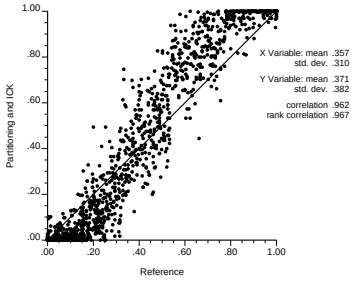
(c) Coarse fraction errors when simulating only with the ICK-derived probability



(d) Coarse fraction reproduction when simulating only with the ICK-derived probability



(e) Coarse fraction errors when simulating with both the ICK-derived probability and partition classes



(f) Coarse fraction reproduction when simulating with both the ICK-derived probability and partition classes

Figure 4.23: Spatial distribution of errors when simulating without the servo-system. The scale goes from -0.3 to 0.3. The scene is 49×28 pixels ($\approx$ 44km $\times$ 25km).

Chapter 5

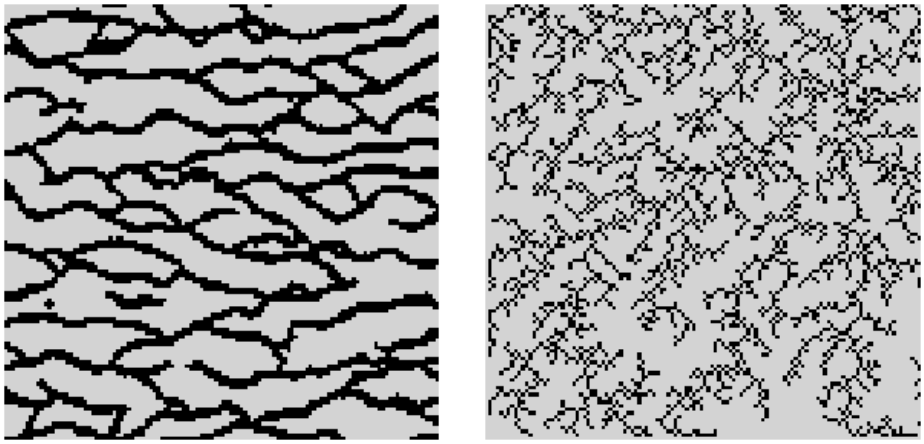
On the importance of structural models

Chapter 2 presented a generic algorithm, sequential simulation, for generating fine scale categorical maps from coarse fraction data. The key to the algorithm is the specification of the structural model defining the type of patterns expected to be found at the fine scale. Two modes of encapsulating this prior structural information have been suggested: the variograms in Chapter 3 and the training image in Chapter 4.

This chapter looks at the difference between these two types of structural models. The choice of structural model leads to downscaled class label maps that exhibit a variety of spatial patterns ranging from simple to complex. The objective is to illustrate, via four new examples, the impact of the prior structural model on the patterns exhibited by the resulting fine scale maps.

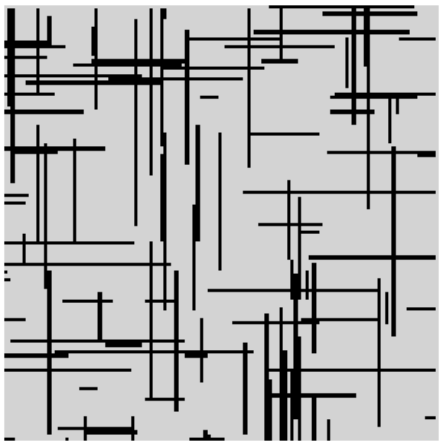
Consider the four synthetic reference binary images shown in Figure 5.1 (a) to (d), each of size 120 $\times$ 120 fine pixels. Theses images are hereafter called Case #1 to Case #4, respectively. Case #1 shows elongated curvilinear shapes; Case #2 shows dendritic patterns generated by a Diffusion Limited Aggregation (DLA, a fractal growth model) algorithm (Viczek, 1989); Case #3 shows intersecting vertical and horizontal lines of varying length; Case #4 shows dense patches aligned along a 45 degree azimuth. The corresponding coarse fraction maps, computed from each reference image, are shown in Figure 5.2 (a) to (d). These coarse fraction maps were

computed by upscaling (via linear averaging) the reference maps into blocks of 15 × 15 pixels each, creating images of 8 × 8 coarse pixels.

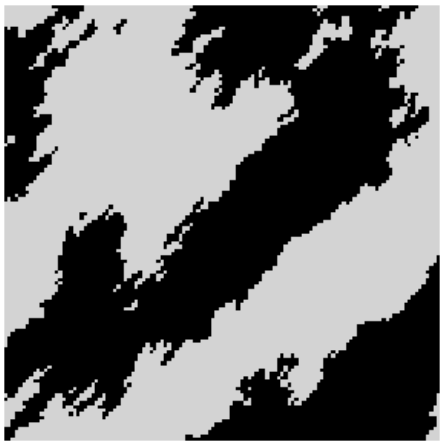


(a) Case #1

(b) Case #2



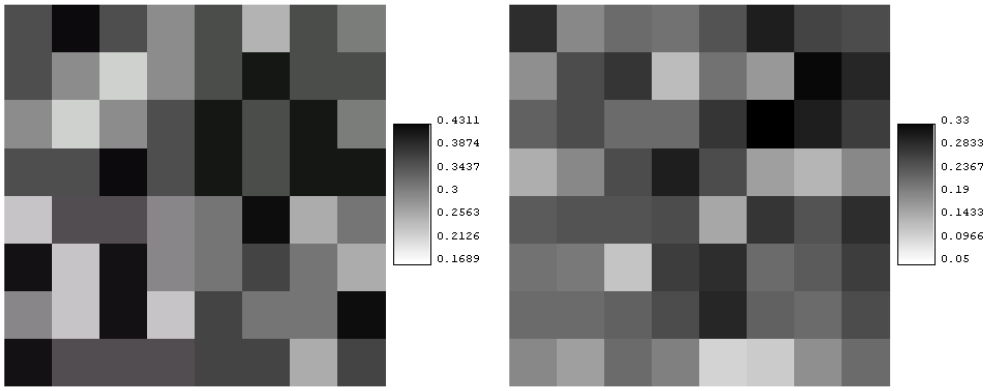
(c) Case #3



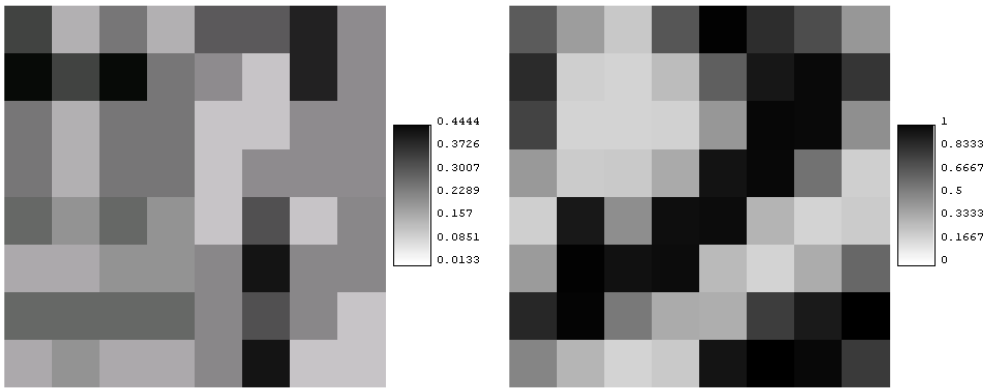
(d) Case #4

Figure 5.1: Reference images of size 120 × 120 fine pixels, labeled Case #1 to Case #4.

Four prior structural models, shown in Figure 5.3, are assigned to the four cases of Figure 5.1. The structural models for Cases #1, #2 and #3 are given as training images, while a variogram model is given for Case #4. Note that the training and reference images are not identical but were generated with the same process; they only share the same patterns. The training images are also larger than the reference



(a) Case #1 (b) Case #2



(c) Case #3 (d) Case #4

Figure 5.2: Coarse fractions for the black class computed from the reference images in Figure 5.1. Each coarse pixel contains 15×15 fine pixels.

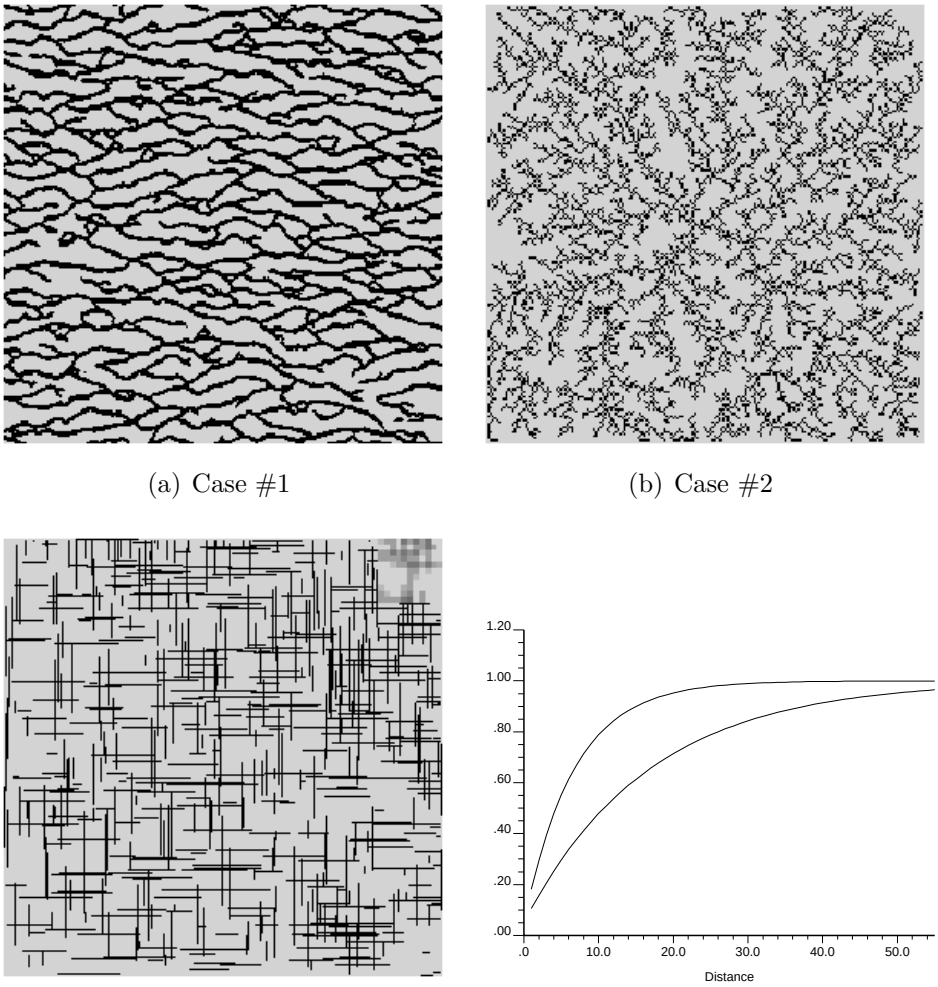
maps. Case #4 was generated with an anisotropic indicator variogram model with maximum range 50 fine pixels along the N45E direction and minimum range 20 fine pixels along the perpendicular direction. This variogram model includes a 5% nugget effect. Note that for a binary case, the indicator variograms of both classes are the same.

Two downscaled class label realizations are generated for each scenario and shown in Figures 5.4 and 5.5; each of these maps reproduces almost exactly the corresponding coarse fraction map in Figure 5.1. Fine resolution maps for Cases #1 to #3 were generated using a multi-grid approach with three nested grids without the partitioning, see Chapter 4. Two additional passes are performed on nested grids #3 and #2. For the realizations generated with a training image, the tau-exponent values τ_{Ti}, τ_R and τ_{IK} in Equation 2.6 and Equation 4.1 are given in Table 5.1. The coarse fraction reproduction is checked in Figure 5.6, where the class proportions derived by upscaling the fine resolution realizations are plotted against the original coarse fraction data; a 45° line indicates perfect reproduction. The spatial pattern information carried by the prior structural model, a training image for Cases #1 to #3 and a set of indicator variogram models for Case #4, are, at least visually, well reproduced in the fine resolution maps. A shortcoming of the downscaled class label maps generated from a training image, and to a lesser extent from a variogram model, is the presence of isolated black pixels (noise) that are not seen on the training image. This is caused by the interaction between the random path and the servo-system: as the simulation progresses it may reach a point where the correct number of white class labels have been simulated within a specific coarse pixel. The remaining fine pixels are then assigned a black label independently of their neighbor pixel values. Note that the same artifact was also observed in the previous case study in Sections 3.4 and 4.4, see Figures 3.5, 3.18, 4.12 and 4.15 for examples.

Nested grid	τ_{TI}	τ_{IK}	τ_R
# 3 (coarse)	1	1	0.01
# 2	1	1	0.7
# 3 (fine).	1	0	1

Table 5.1: Exponent values for each nested grid.

A measure of (two-point) spatial statistics reproduction is given in Figure 5.7,



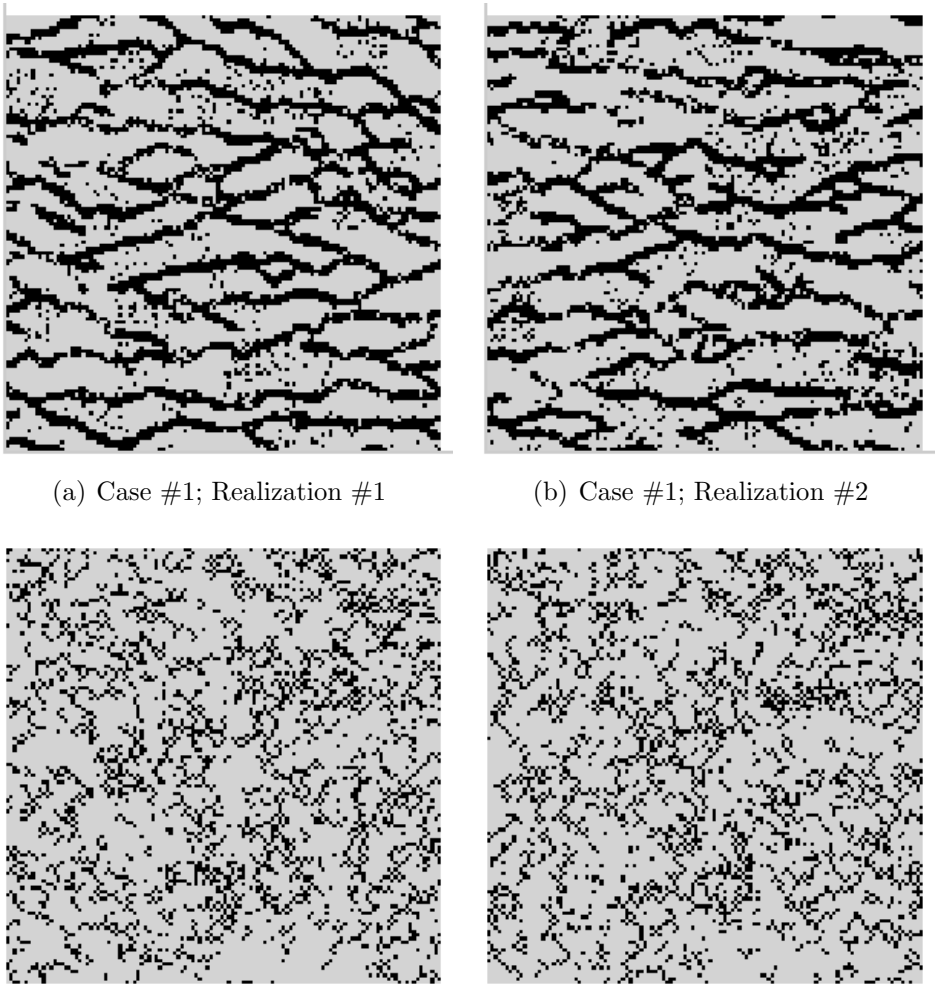
(a) Case #1

(b) Case #2

(c) Case #3

(d) Case #4

Figure 5.3: Structural models for Case #1 to Case #4. The training images are of size 250 $\times$ 250 fine pixels. Right : Structural model for case #4 is an exponential variogram model with anisotropy; long range is 50 fine pixels, short range is 20 fine pixels.



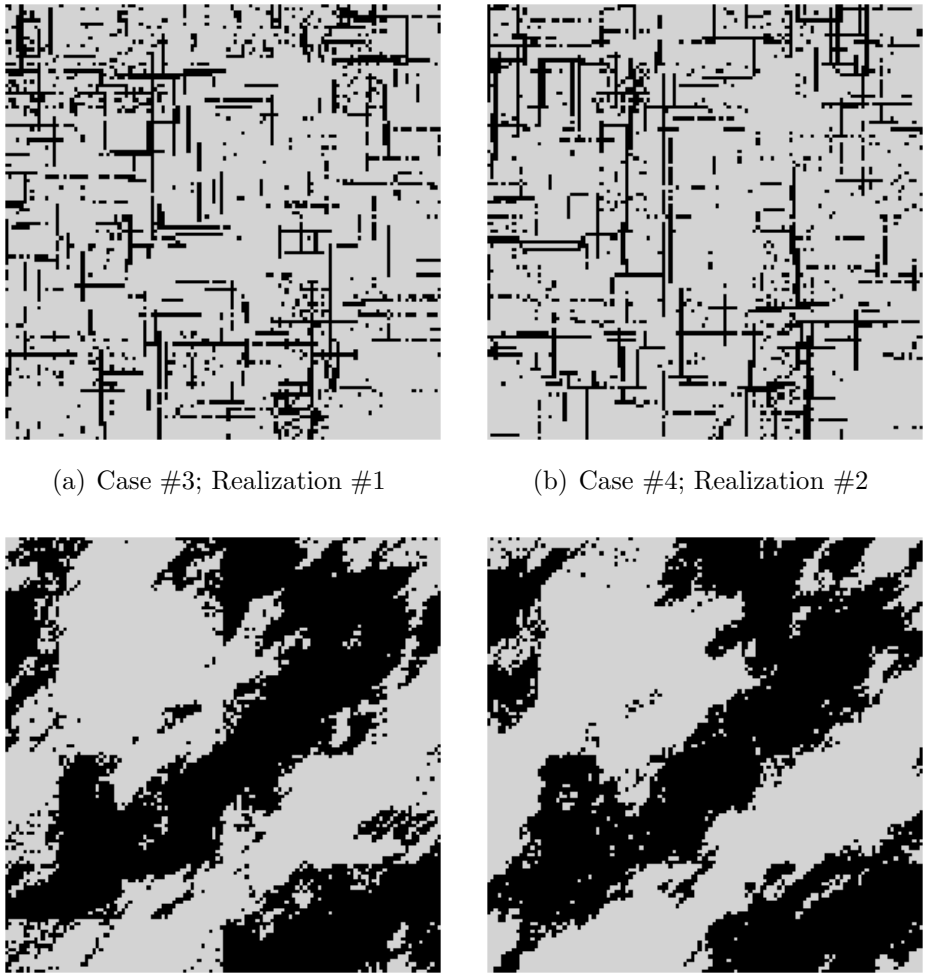
(a) Case #1; Realization #1

(b) Case #1; Realization #2

(c) Case #2; Realization #1

(d) Case #2; Realization #2

Figure 5.4: Downscaled class label realizations for reference Case #1 and Case #2 when using the correct structural model.



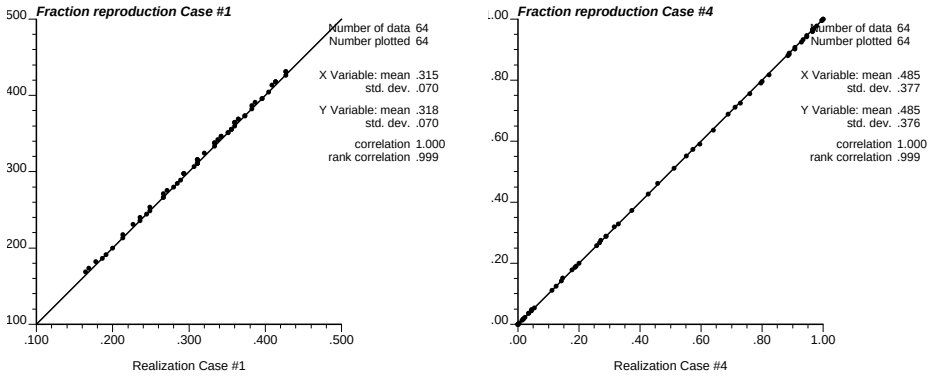
(a) Case #3; Realization #1

(b) Case #4; Realization #2

(c) Case #4; Realization #1

(d) Case #4; Realization #2

Figure 5.5: Downscaled class label realizations for reference Case #3 and Case #4 when using the correct structural model.

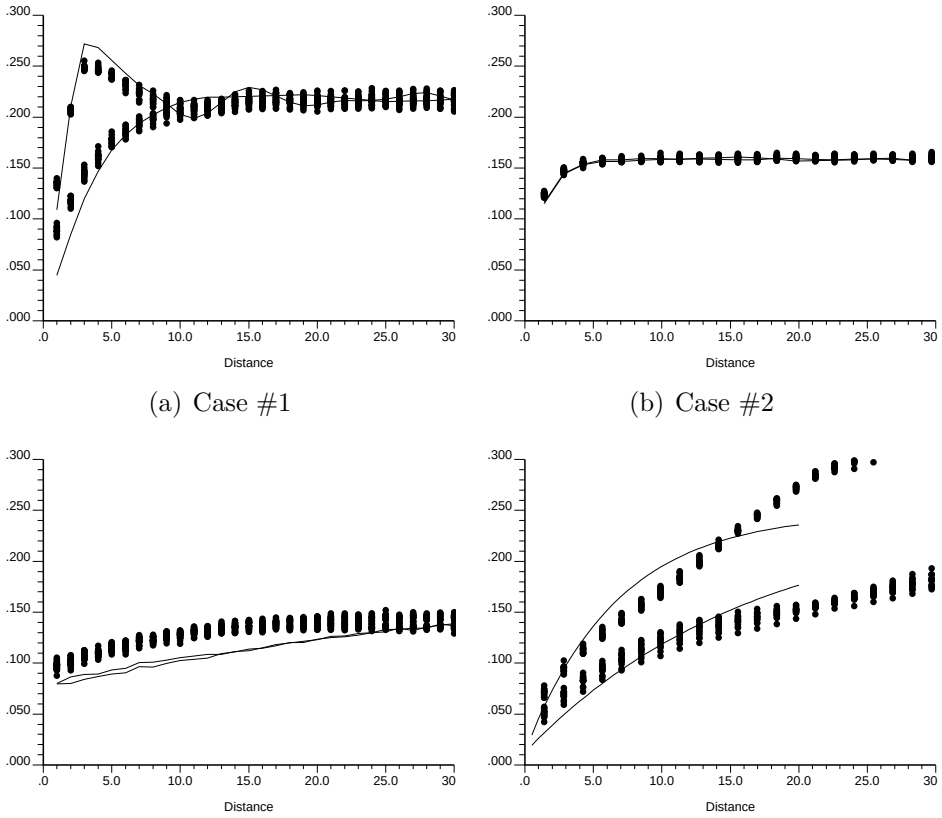


(a) Case #1; Realization #1 (b) Case #4; Realization #1

Figure 5.6: Coarse fraction reproduction for the downscaled class label maps of Figure 5.4(a) and Figure 5.5(c); a 45 degree line indicates a perfect reproduction.

where the experimental variograms calculated from 15 fine resolution realizations for each case are compared with the variograms implicit to Cases #1, #2 and #3, and the explicit variogram model for Case #4. All variograms except perhaps for Case #3 are reasonably well reproduced. Note that variograms for Case #1 and Case #2 are well reproduced although they never have been given explicitly to the simulation algorithm; two-point statistics or variograms are contained in the training image.

Consider now rotating the structural models for Cases #1 and #4 of Figure 5.3 by 90 degrees clockwise. The curvilinear features of Case #1 are now running N-S and the maximum range of the variogram in Case #4 now has direction 135 degrees. Figure 5.8 shows the downscaled class label realizations generated for each scenario. Although the coarse fraction data have not been rotated, the realizations reproduces the continuity of the structural model with its new orientation. The level of conditioning to coarse fractions greatly differs in these two cases. For the Case #1 the coarse fraction information is not enough to control the orientation of the curvilinear features. That low level of coarse information gives all latitude to generate curvilinear features with N-S orientation. On the other hand, the coarse fractions of Case #4 are very informative (values close to 0 or 1) and override the wrong input direction of maximum continuity at 315°, Figure 5.8 (b) still shows elongated wide bodies at 45°. The impact of rotating the variogram model can only be seen on the border of the wide bodies. Recall that, per the servo-system, all fine scale maps reproduce exactly

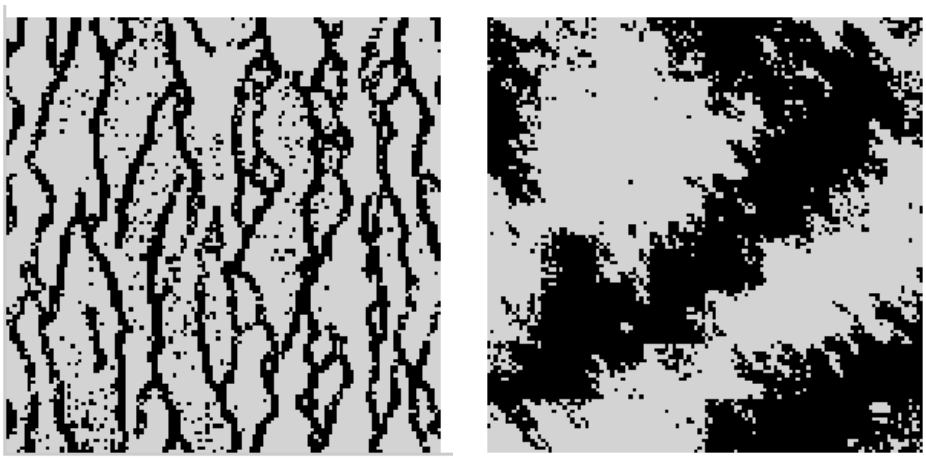


(a) Case #1 (b) Case #2

(c) Case #3 (d) Case #4

Figure 5.7: Indicator variogram reproduction for each scenario. Directional variograms of 15 downscaled class label realizations are shown with dots. For Cases #1, #2 and #3, solid lines show directional sample variograms computed from the corresponding training image; the solid line in Case #4 displays the input directional variogram models.

the coarse fractions; see Figure 5.6.



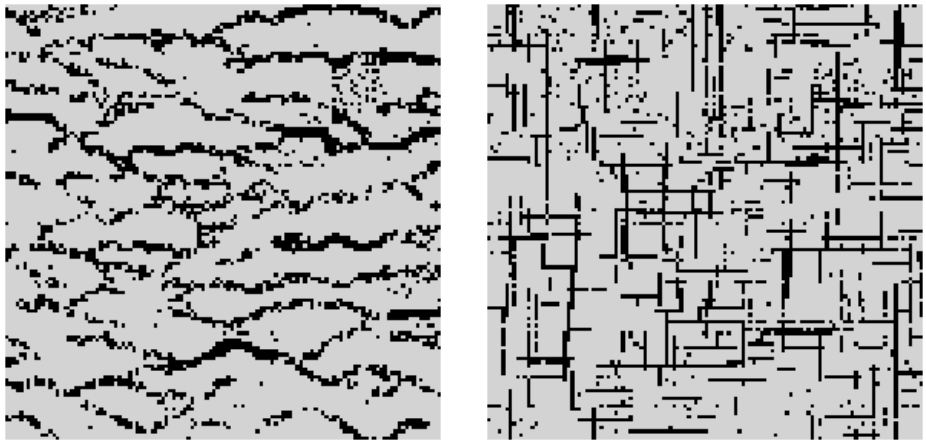
(a) Case #1 with the TI rotated by 90 degrees; Realization #1

(b) Case #4 with the variogram rotated by 90 degrees; Realization #1

Figure 5.8: Downscaled class label realizations, using the coarse fraction data of Figure 5.1, but with the structural model of Figure 5.3 rotated by 90 degrees clockwise.

Next, the impact of a wrong prior structural model on the resulting downscaled realizations is investigated. Figure 5.9 (a) shows a realization generated using the coarse fractions of Case #2 as data (Figure 5.2 (b)) together with the Case #1 structural model (Figure 5.3 (a)). The SNESIM algorithm is not able to reproduce correctly the continuous curvilinear channels of the training image retained. The difference in the class proportions of Case #1 and Case #2 means that the model cannot fit simultaneously the training image and the coarse fractions data. Figure 5.9 (b) shows a realization generated with the coarse fractions of Case #2 as data (Figure 5.2 (b)) together with the structural model of Case #3 (Figure 5.3 (c)): the interconnected horizontal and vertical lines can still be seen on the resulting fine scale map. In this situation, the structural model remains consistent with the coarse fraction data. The similar class proportions for Cases #2 and #3, and the latitude given by the short scale variation of the linear patterns allow this reproduction.

Case #4 is the example best informed by the coarse fraction data. The density and continuity of the coarse fraction pixels make the length, form, and orientation of the reference image main bodies easily discernable, so the effects of downscaling are seen mainly at the borders of such bodies. Even in such a case, the structural



(a) Coarse fraction from Case #2 and TI from Case #1

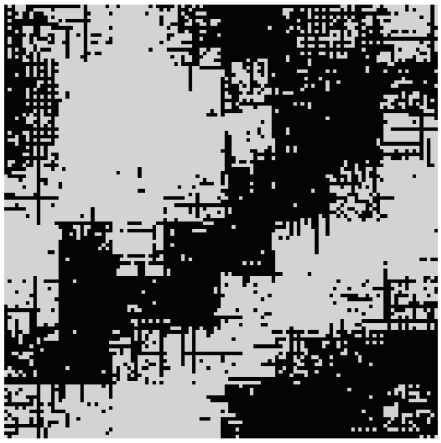
(b) Coarse fraction from Case #2 and TI from Case #3

Figure 5.9: Downscaled class label realizations conditioned to different coarse fraction data and generated using various structural models. Left: Coarse fraction data from Case #2 with the curvilinear features training image. Middle: Coarse fractions from Case #2 with the structural model of Case #3.

model matters; Figure 5.10 shows two fine resolution realizations generated using the coarse fractions of Case #4 as data (see Figure 5.2 (d)) and two "wrong" structural models: the curvilinear TI (Case #1) and the orthogonal lines TI (Case #3). In both cases, the coarse fraction data are not compatible with the structural model. Even so, features from the training images can be seen on the resulting downscaled class label maps. In Figure 5.10(a), simulation using a structural model depicting curvilinear features manages to connect two large bodies. Once again recall that all simulated fine resolution maps reproduce exactly the coarse fractions.

Last, Figure 5.11 indicates that it is possible to represent a variogram with a training image. From the training image for Case #1 in Figure 5.3, a variogram map is tabulated and used with the indicator cokriging algorithm to simulate a fine scale map using the Case #1 coarse fraction data (Figure 5.2 (a)) for conditioning. No variogram modeling is needed here; instead, variogram values are computed directly from the training image via fast Fourier transform ensuring a valid variogram look-up table (Section 3.1.2). Conversely, a training image is generated with the variogram model of Case #4 (Figure 5.3 (d)) by performing an unconditional simulation using

(a) Coarse fraction from Case #4 and TI from Case #1



(b) Coarse fraction from Case #4 and TI from Case #2

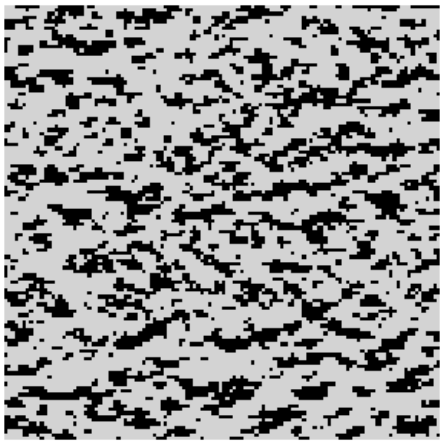
Figure 5.10: Downscaled class label realizations generated using the coarse fractions of Case #4 as data (see bottom row of Figure 5.1) and the structural models for Case #1 (left) and Case #3 (right).

the indicator block cokriging algorithm. That training image is then used with the SNESIM algorithm to generate a fine resolution map using the coarse fraction data of Case #4 (Figure 5.2 (d)) for conditioning. Figures 5.11 (a) and (c) should be compared with Figure 5.4 (a) and Figure 5.5 (c), respectively. Figure 5.11 (b) and (d) compare the variograms of the realizations generated from this experiment with the 15 variograms obtained from corresponding realizations generated using the original structural models and already shown in Figure 5.7. No curvilinear features are reproduced in Figure 5.11 (a) despite the good reproduction of the variogram model by the SIS approach using the TI-derived variogram table. Instead, the simulation displayed a fragmented texture with a weak N-S periodicity. That periodicity is also seen as a hump on the experimental variogram in Figure 5.11 (b). A variogram model is not sufficient to reproduce the patterns of a training image, whereas a training image includes the information carried by a variogram.

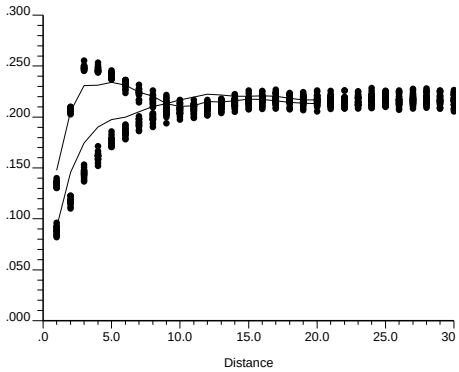
The first step in deciding which structural model to choose is to identify the critical fine scale patterns that must be reproduced. Failure of non conditional realizations to reproduce these critical patterns eliminates the model. Only once a model has been deemed satisfactory by its unconditional simulations should it be used with data. See

Appendix B for a general discussion on choosing a structural model. If the critical patterns are equally well reproduced by a variogram or a training image, then the simpler variogram-based approach should be used. The drawback of training image-based approaches is the higher level of noise artifact in the resulting simulation, compare Figure 5.11 (c) to 5.4 (d). The same observation was made for the case studies of Chapter 4.

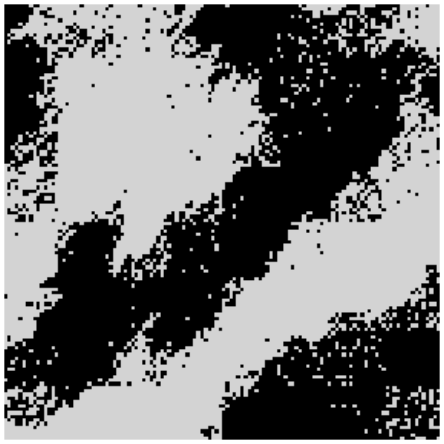
The above experiments demonstrate the complex relationship between coarse fractions and structural model. When the coarse fraction data conflict with the structural model, the latter is not reproduced; conditioning to data taking precedence to pattern reproduction. Such scenario is indicative of a poorly chosen structural model that should be re-considered.



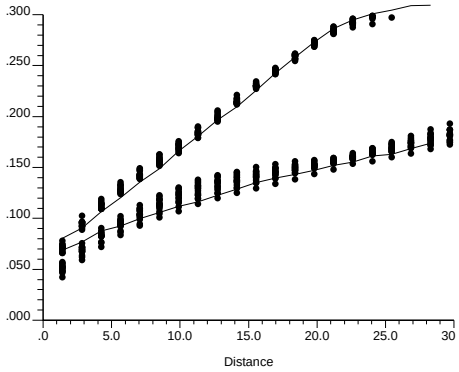
(a) Coarse fraction from Case #1 and variogram obtained from the TI of Case #1



(b) Variogram reproduction of (a)



(c) Coarse fraction from Case #1 and TI obtained from the variogram of Case #4



(d) Variogram reproduction of (d)

Figure 5.11: (a): Downscaled class label realizations generated using the coarse fraction data of Case #1(see bottom row of Figure 5.1 (a)) and indicator variograms calculated from the corresponding training image of Figure 5.3 (a); (b): Directional indicator variograms for (a) (solid lines), along with the 15 corresponding variograms (dots) shown in Figure 5.7 (a). (c): Downscaled class label realization generated using the coarse fraction data of Case #4 (see bottom row of Figure 5.1(d)) and a training image generated from the variogram model of that case (see Figure 5.3 (d)) using SIS. (d): Directional indicator variograms of (c) (solid lines), along with the corresponding variograms (dots) shown in Figure 5.7 (d).

Chapter 6

Discussion and Conclusions

This thesis contributes to two fields of study, satellite remote sensing and geostatistics. A novel approach to downscaling coarse satellite image was presented wherein downscaling is viewed as an under-determined inverse problem, that of constructing fine resolution class label maps from coarse class fraction data. The focus is on the pattern reproduction instead of the traditional local accuracy criterion. From a geostatistical perspective, this thesis offers two major contributions with potential wide applicability: a probabilistic servo-system to match target statistics and an operational multi-scale simulation anchored on a training image.

This last chapter discusses the methods and results presented in this thesis from both remote sensing and geostatistical perspectives. Then, avenues for future work are presented.

6.1 Downscaling remote sensing data

The major novelty of this work from a remote sensing perspective is the emphasis on patterns reproduction instead of the classical focus on local accuracy. Since downscaling is an ill-posed inverse problem, to resolve the inherent ambiguity it is necessary to have an additional prior model of spatial structures deemed to exist at the fine (target) resolution. A mapping algorithm cannot exist without such prior structural model be it implicit or explicit. The downscaling techniques presented in this thesis allow the user to control the patterns generated in the fine scale maps to produce

structurally accurate representations of the landscape.

Local and structural accuracy are two conflicting goals in mapping. The aim of the proposed algorithms is structural accuracy, that is the reproduction of the patterns expected to be seen at the fine scale. Local accuracy refers to the comparison pixel-per-pixel (no patterns involved) with some validation class labels. The two goals are mutually exclusive since local accuracy calls for a conservative mapping, with high frequency patterns smoothed out and extremes patterns omitted. The decision of whether to aim for local or structural accuracy depends on the intended use for the resulting map(s). If the application relies on patterns, i.e. the texture, of the map, then one must aim at structurally accurate maps. Note that in either case the available local data are honored whatever their support.

Downscaling coarse fraction data into fine scale labels is often not the end-goal per se; instead, downscaled maps typically serve as inputs to flow simulators, detailed spatial analysis operations, coupled environmental models, or decision support systems, all requiring input data at that fine resolution. The consequences of the uncertainty associated with downscaling on the outcomes of the above operations need to be investigated within an uncertainty propagation context. The methods provided in this thesis for generating synthetic fine scale numerical representation can be used towards such an uncertainty assessment endeavor in a Monte Carlo simulation framework.

Two types of structural models are accepted, variogram-based and the training image-based models. The variogram only models linear correlation between any two points in space; variograms can range from a pure nugget effect model, which is indicative of completely random spatial variation of class labels, to models with extremely large range and no nugget contribution, indicative of strong continuity in the spatial patterns of class labels. The training image on the other hand is a fully explicit depiction of the patterns expected to be found at the fine scale.

Any structural model calls for some assumption of stationarity. The concept of stationarity is at the basis of all statistical inference, it allows to predict an unknown value from an environment somewhat different from that of the outcomes of the variable being estimated. For example, in one of the case studies, a decision was made that the patterns in one city can be used to map the urban forms of another city. Such critical decision (stationarity) is somewhat necessarily subjective, yet it

defines the probabilistic model and thus impacts critically the resulting map and estimates.

Regardless of the source of information about spatial structure, any super-resolution mapping attempt is based, either explicitly or implicitly, on the availability of such a model. Consider, for example, the case of simple contouring of coarse fractions to derive a map of fine resolution fractions, or better stated a map of probabilities of class occurrence. Any per-pixel thresholding of such fine resolution fractions into class labels will result in a particular (albeit not explicitly defined) spatial pattern of land cover. The thresholding procedure, although very simple, invokes implicitly a hidden model of fine resolution spatial structure. That model is embedded in the weights used for contouring the coarse fractions, and in the constraints of contiguity (if any) adopted for the final class allocation step. It is far better to state upfront the prior fine scale structural model guiding the construction of the fine resolution maps, rather than hiding such prior model under some assumption, such as maximum spatial continuity, whose consequence is not always well understood.

Once a prior model of spatial structure is defined, a sequential simulation algorithm can be used to generate alternative plausible solutions to the downscaling problem. A plausible solution is any fine scale map that simultaneously reproduces (i) the structural model adopted (variogram or training image), (ii) the fine scale hard data (if available), and (iii) the corresponding coarse fraction data. Multiple plausible solutions exist to this ill-defined downscaling inverse problem; consequently, the mapping process must be stochastic involving the task of generating or "exploring" such equally plausible solutions.

Two novel, non-iterative, geostatistical simulation algorithms for generating such alternate, sub-pixel maps have been developed. Being non-iterative the downscaling is fast enough to generate many (10s or 1000s) alternative solutions. Furthermore, these algorithms are flexible; they can handle different geometric patterns simply by inputing a different structural model. This a major advantage over the classical maximum continuity algorithm. By separating the mapping/simulation algorithm from the structural model, the two proposed algorithms can downscale both fragmented and continuous landscapes. One only needs to change the input structural model to change the patterns displayed in the realizations.

For a given prior structural model all the resulting maps are equally likely. None of these plausible solution maps is expected to be more locally accurate than any other. As long as these maps reproduce the available data, they are all equally likely to yield similar accuracy scores in a validation test. This statement involves an expected value concept: a particular realization might yield a higher accuracy score than another. But it is expected that all solutions generated from the same prior structural information, and accounting for the same data via the same algorithm, have similar degrees of per-pixel mismatch with the true (but unknown) land cover map. Note that the accuracy scores of a structurally accurate map is necessarily lower than that of a fine scale map specifically designed for local accuracy. Recall, however, that any input hard data will be reproduced exactly by all the algorithms proposed.

From this viewpoint, an accurate map in downscaling is cast not as the most per-pixel accurate map but as a map that accurately reproduces the prior structural model in addition to identifying the local data (coarse and possibly fine scale data).

There remains the issue of choosing a prior model of spatial structure that accurately displays the expected fine scale spatial patterns. The inference or choice of a prior structural model remains a complex and all-important task. The choice of a structural model can be guided by interpretation of the coarse-fraction data, and by any prior knowledge regarding the fine scale spatial patterns. Not all structural models may fit a particular coarse fraction map; some coarse fraction data may impose tighter constraints on the possible patterns of class labels at the fine resolution.

The issue of choosing an appropriate analog image is still an open yet critical question. This thesis did not address this all important issue, but instead provided a means for using any such prior model in the downscaling process. Chapter 5 demonstrated the importance of the structural model used in downscaling class label. The reproduction of the patterns contained in the structural model depends on several key factors:

- The difference in class proportions (marginal distribution) between the structural model and the coarse fraction data
- The spatial heterogeneity of the coarse fractions versus that of the fine labels
- The size, shape, and orientation of the spatial patterns of the prior structural

model

A training image has the advantage over a variogram model of being explicit in the sense that one can see which patterns are considered. Any expert who is not familiar with geostatistics theory can more easily validate a training image than a variogram. The former can be visualized and accepted or discarded based on the expert expectations and experiences. Choosing a variogram model is more complex. In its parametric form it is simply an equation. It is difficult to intuitively relate that equation to specific patterns of land cover/land use and their texture. The patterns implicitely encapsulated by any specific variogram model can be visualized by generating a few unconditional simulations on a large grid, this would be helpful in determining the relevance of the patterns implicit in the variogram formulation, see Figure 5.11 for an example.

The two models, the variogram and the training image, are not equivalent. A variogram model only represents the two-point statistics of a training image. Many very different training images may share a common variogram. Whereas a training image taken as whole can relate the probability of having a specific class label at an unknown location to the multi-pixel data event (multiple-point statistics) informning that location. The variogram only relates the unknown to one known pixel at a time, hence it is a two point-statistics. The variogram models the correlation in space between two points while the training image provides the correlation between a point and a multiple-point configuration of data values. The statistics carried by the variogram are part of the training image. There are patterns that cannot be summarized by these two-point statistics and require a fuller use of the training image, see Figure 5.11 for example.

Uncertainty about the choice of a structural model can be added into the analysis by considering several candidate structural models. Based on different analog data and expert judgements, these alternate models could arise from a sensitivity analysis or a scenario building exercise. For each alternate scenario several downscaled fine spatial resolution maps can be generated. The overall uncertainty about the location of the class labels is given by the set of all realizations generated from all the structural models retained.

Although only hard (either 0 or 1) indicator data were used in the case studies

presented, it is possible to incorporate soft information probability data into the analysis. Such probability data can be directly handled by indicator kriging; see, for example, Goovaerts (1997). Auxiliary data, such as terrain elevation, could provide an estimate of the probability of class occurrence at the fine resolution, through, say, a logistic regression model. This additional information can be accounted for in the both proposed approaches, in lieu of the currently used 0/1 class indicator coding.

Some downscaling applications may call for structural models much more complex than the ones considered in this work. Experience has shown that the classical SNESIM algorithm can work successfully with training images containing up to five categories for simple cases, and up to three categories for cases with complex patterns. A hierarchical approach, by embedding classes, is traditionally used for such multi-categories problems. For example, consider the case of five land cover classes; high-rise urban, low-rise urban, forest, agriculture and bare soil. First these five land covers would be split into two super-classes: urban (high- and low-rise) vs non-urban (forest, agriculture and bare soil). A binary downscaled map for these two super-classes would be generated with any of the proposed algorithms. Then each simulated super-classes will be further processed to obtain the sought after five land cover categories. Instead of requiring a potentially large five categories structural model, one would need to specify two binary models (urban vs non-urban and high- and low-rise) and one three-category models (forest, agriculture and bare soil). Note that the partitioning approach presented in Section 4.2.2 by generating smaller trees will further ease the computational efficiency of applications involving several nested categories.

The proposed sub-pixel mapping methods were illustrated via two case studies derived from two Landsat TM scenes. These case studies are from the Pearl River Delta (PRD) and from the city of Guangzhou (GZ). The proposed methods have been shown to be efficient for mapping land cover using coarse resolution data, the only requirements being that: (i) the classes are mutually exclusive and collectively exhaustive, (ii) the classification scheme is valid at both coarse and fine resolutions, and (iii) one has available a prior model of spatial structure for the fine (target) resolution. Requirement (i) entails that there are no fuzzy but only crisply-defined classes, and that these classes cover all possible states that are expected to be found

at the fine spatial resolution. However, soft labels data (e.g., defined by probability intervals) can be accounted for by soft indicator Kriging (Goovaerts, 1997). When the classes are not deemed collectively exhaustive, one could artificially define an additional class which encompasses labels not classified in any other class. Requirement (ii) implies that the adopted classes are meaningful, and satisfies requirement (i) at both coarse and fine resolutions. Requirement (iii) is, typically, the most difficult piece of information to obtain in practice.

Ideally all available information should be used in the downscaling process. In that regard, this thesis also addresses such data integration including : (i) delineated water bodies, (ii) road networks, and (iii) previously mapped impervious surfaces. Providing more and consistent information results in more realistic class maps that those obtained based only on coarse fraction data.

The applicability of the method to high-definition satellite imagery (sub-meter spatial resolution) is not trivial. At that scale, the structure is essentially made of defined crisp objects (buildings, cars, roads, outcrops, trees, etc.) that may not be suitable for these algorithms. The proposed methods seem better adapted to lower spatial resolution with more diffused patterns.

On a more conceptual note, the proposed algorithms are not traditional statistical algorithms. As explained in Chapter 1, they are stochastic solutions to the downscaling problem. Traditionally, the modeling is done by first choosing an analytical random function then estimating its parameters from the data and developing an algorithm for simulating that random function. This is not the route chosen in this thesis. The proposed algorithms, either variogram or training image-based, do not have analytical definitions. The underlying random function is defined by the algorithm itself and its parameters; such random functions are called algorithm-driven random functions (ADRF), see Appendix B for more details. For example, the random function modeling the GZ data set in Figure 4.15 is constituted by the snesim algorithm, the Foshan training image with six partition classes, a template with 60 pixels, three multiple grids and enabling the servo-system. To study the properties of that random function one can generate several unconditional realizations and retrieve their statistical properties.

6.2 Contribution to geostatistics

This thesis offers some major contributions and advances to the field of geostatistics. First, there is the challenge of applying existing techniques to a new application field. The practice of multiple-point geostatistics practice has been mostly limited to petroleum reservoir characterization. These applications have few hard data and the training image is typically of low entropy with crisp, well defined patterns. The case studies in this thesis involved dense conditioning data and higher entropy training images. This calls for non trivial modifications of the standard algorithms. Hard data conditioning is a case in point: the SNESIM procedure whereby hard data are re-allocated to grid nodes is not appropriate anymore and had to be replaced with a probability field approach.

Other advances on the algorithmic side pertain to the servo-system to reproduce target statistics, block indicator kriging, and a proposed partitioning of the global search tree. Each contribution is critically reviewed below.

6.2.1 Probabilistic servo system

Reproduction of coarse fraction data is a necessary and desired feature in downscaling categorical data. In the proposed algorithms, exact reproduction of the coarse fraction data is enforced through a tau model based servo-system. The coarse fraction reproduction is obtained by integrating the usual data event-derived probability with a probability that conveys the mismatch between the target and simulated proportions.

Through the diverse case studies presented this new servo-system was proven to be an effective mechanism to ensure coarse fraction reproduction. It has however the drawback of generating speckles (noise) especially when downscaling is done with a training image-based algorithm. Such noise is generated toward the end of the simulation path when probabilities with values of 0 and 1 are passed to the tau model; such hard probabilities overwrite all other sources of information. The reason is that the function that turns the mismatch into a probability acts as a greedy optimizer. It does not try to look forward in the simulation if discontinuities are likely to be created. If needed, the speckles in the images can be removed through

an iterative post-processing algorithm, such as the one presented in Appendix A, or possibly through a series of backtracking while simulating. Both options are time consuming.

Even with this noise drawback, the servo-system remains efficient compared to the alternatives. Most of the alternate techniques for reproducing coarse fractions are iterative; they perturb a fine scale image until the noise or other unwanted features are removed. First, such iterative techniques cannot be used on a large image, since it would not be feasible to generate many high quality (noise free) realizations within a reasonable amount of time. The option then is to stop the iterative process before convergence, hence, leaving some noise in the fine scale map. If both exact coarse fraction reproduction and a noise free image are required, then the map resulting from approaches proposed in this thesis could be used as a starting image for further processing with an iterative technique.

The coarse fraction data may contain some measurement errors that should be accounted for in the coarse fraction reproduction. The function that turns the mismatch into a probability can be adapted to allow a tolerance based on the measurement errors. This tolerance would prevent generating hard 0 and 1 mismatch probabilities.

6.2.2 Downscaling proportion data with indicator kriging

Downscaling of continuous linear averaged data defined over large support volumes is well understood theoretically (Journel and Huijbregts, 1978). A kriging system with regularized variogram automatically ensures the reproduction of these linear average data. Downscaling categorical attributes uses the same basic kriging equation as in the case of a continuous attribute. However, unlike the continuous case, these equations only ensures coarse fraction reproduction in expected value, that is in average over many simulated realizations. The servo-system solution proposed here overcomes that limitation. The examples shown in this thesis demonstrate strict reproduction of proportion data in each realization.

Theoretically, spatially autocorrelated error measurements for coarse fraction data could be taken into account directly in the ICK system by adding a noise term to the left-hand side matrix under the assumption that the measurement errors are homoscedastic (independent of the actual data values). Even if such an assumption is

met and because the ICK system alone fails to reproduce the coarse fraction, the additional tolerance for the measurement errors could not be distinguished from ergodic deviations from the target proportion. Therefore, the ICK is suited for downscaling in expected value the coarse fraction data, but an exact reproduction of these fractions requires using the servo-system.

6.2.3 Multi-scale SNESIM algorithm

A major proposal of this thesis, the search tree partitioning, addresses the important issue of varying data support in the training image-based algorithm SNESIM. It allows to condition a training-image based realization directly to proportion data defined on a support larger than that of the fine scale pixels of the training image. The proposed solution is to pre-classify all fine scale patterns of the training image into a few partition classes based on their coarse fractions. All patterns within a partition class are recorded by a search tree; there is one tree per partition class. At each fine scale pixel along the simulation path, the coarse fraction data is retrieved and used to select the appropriate search tree. That search tree contains the patterns relevant to that coarse fraction data.

The proposed search tree partitioning compares positively to the alternative of using the traditional SNESIM algorithm with the ICK-derived probabilities as soft data. The search tree partitioning is more robust with regard to departure from stationarity, it provides better coarse fraction reproduction and is faster. These three points are elaborated next.

Stationarity

Using a real analog as training image for the original SNESIM algorithm results in poor simulation because analog images are typically non-stationary. By pooling all the patterns in a single search tree but not recording the pattern relative locations, some critical features of that analog get lost.

The search tree partitioning subdivides the large training image into imbricated, stationary, smaller images, called partition classes. Each of these smaller images have a corresponding search tree that can be utilized by the SNESIM algorithm.

Furthermore, these search trees are necessarily consistent one with another since they were built from the same training image. The result of this partitioning is a series of search trees where any recorded fine scale patterns is associated with a partition class defined on a coarse scale. The partition classes account for coarse scale patterns, such as trends, that could not be captured at the fine scale. The smaller, more stationary, training images associated with the partition classes permit the use of large non-stationary analogs as training image.

Take the example of an analog city which typically has a dense central business district, surrounded by a ring of commercial and residential developments and some rural settlements farther away. The distinct patterns and density of these different areas will get lost if pooled together into a single search tree since their relative locations are not recorded. From a single search tree it is impossible to know that the patterns associated with the business district are at the center of the image and that the patterns associated to commercial and residential areas surround that center. The coarse fraction data derived from the satellite sensor provide an indication about these large scale relationships. The smaller-scale patterns within each different district are provided by the smaller search trees.

Coarse fraction reproduction

In addition to providing better pattern reproduction, the search tree partitioning also enables direct conditioning to the coarse fraction data. By partitioning appropriately the large training image, each partition class will contain coarse pixels with homogeneous coarse fraction values. The simulation within a coarse pixel with, say, high proportion of category #1 will use a search tree that has recorded data events associated with high proportion of category #1. Hence most of the simulated labels within that coarse pixel will be of category #1 yielding the sought after high proportion. The quality of the coarse fraction reproduction is thus function of the number and composition of these partition classes. Beware though that each partition class must contain enough replicates to build a search tree.

The lesson learnt from the case studies is that better coarse fraction reproduction is obtained by combining the indicator cokriging (ICK)-derived probability and the search tree partitioning approach. If the reproduction is still not acceptable then

one should consider using the servo-system. However, the gain in coarse fraction reproduction brought by the servo-system may not be worth the additional artifact noise.

Simulation time

The partitioning approach brings a significant increase in simulation speed. In the case studies, partitioning cuts the simulation time by a factor between 3 and 6 compared to the single tree approach. Increasing the number of partition classes, hence defining more but smaller search trees, increases the simulation speed. Such speed improvement is a significant benefit. Since there is no single "best" map, a large number of simulated maps must be generated to model uncertainty. Any simulation algorithm to be practical must be fast enough to generate tens and possibly thousands of realizations within reasonable time.

6.3 Future Work

The proposed search tree partitioning can be extended by replacing the remotely sensed coarse data with simulated partition classes at the coarse scale. These partition classes can be either directly simulated or generated from a co-simulation of the coarse scale proportions. The direct simulation of the partition classes is easier since simulating highly correlated proportions would call for a difficult joint-simulation. A challenge is to ensure that the spatial distribution of these simulated partition classes is consistent with any available fine scale conditioning data. The simulated field of partition classes would aim at reproducing the large scale structure of the training image.

In this thesis, the partitioning was based on the coarse scale proportions, but proportions may not be the most relevant partition criteria in other applications. Some patterns may be better discriminated using, for example, gradient or curvature filter, principal components or any other image filters. Instead of deriving the partition classes from the coarse proportions, the partitioning could be done on the training image using large scale filter scores such as proposed by Zhang et al. (2006). Note that the proportion data is a special case of a linear filter with equal weights.

The main assumption underlying the search tree partitioning is that the coarse scale response is related to the fine scale patterns. That relationship need not be linear. One could upscale the training image with a non-linear upscaling function and derive the partition classes based on that upscaling. For example, the upscaled values could be hydraulic connectivity (Gomez-Hernandez, 1992). The partitioning would be done on a field of hydraulic connectivity upscaled from a fine scale training image of point support permeability values; fine scale pattern simulation would be function of the local upscaled hydraulic connectivity.

The proposed servo-system is not limited to downscaling applications. Its probabilistic formulation is generic and not tied to coarse fractions reproduction. Any constraint for which the mismatch can be incrementally computed at every node along a sequential path could potentially be enforced with the tau model formulation proposed. One only needs a function that transforms the mismatch value into a probability. For example, a servo-system could be used to improve a target connectivity between two contiguous pixels or to control the number of specific objects or data events to be simulated in any single realization.

The tau model is used extensively in this thesis yet never at its full potential since the tau exponents are most often arbitrarily set to their unit value. Optimizing the tau values could improve many aspects of the algorithms from the servo-system to integration of the ICK-derived probability.

These proposed algorithms should also be expanded to integrate information at intermediate spatial resolution, such as from a different satellite sensor data, and data defined on irregular support volumes, such as data from county/states defined on complex polygonal shapes as commonly found in GIS databases. The block kriging system using the regularized variogram is not limited by the shape and size of the data support. No equivalent general solution is readily available for the training image-based simulation approach.

More research is required to extend the finding of these case studies to different and more complex real data sets, so as to make the proposed mapping approach a robust tool for environmental monitoring. For instance, satellite sensors with low spatial resolution and short re-visit time (high temporal resolution) could be used in monitoring fast physical processes between re-visits of higher spatial resolution

satellite sensors with lower temporal resolution.

Finally, this work assumes the availability of proportion data, which are not direct satellite sensor measurements, but the output of spectral unmixing algorithms. Most of these unmixing techniques do not take advantage of the spatial distribution of training fine scale land covers when processing the satellite sensor measurements. The current two-step process, spectral unmixing followed by downscaling, could be merged into a single process where the spatial autocorrelation of the land covers would come from a training image. Adding more information has the potential to increase the accuracy of the spectral unmixing process.

Appendix A

Post-processing with simulated annealing

The major drawback of the servo-system approach to downscaling is the generation of noise (speckles) in the simulated fine scale images. This appendix proposes to reduce these unwanted speckles by post-processing the simulations with simulated annealing (Geman and Geman, 1984). Recall that simulated annealing (SA) is an optimization technique that aims at reducing an objective function by iteratively changing the simulated value(s). At each iteration a change is performed, that change is always accepted if it decreases the objective function. When the change increases the objective function, it is accepted with a certain probability.

Simulated annealing had been used in geostatistics with objective function a variogram reproduction (Deutsch, 1992), but had been left aside due to its slowness when processing large grids (potentially millions of nodes). By limiting the objective function to within a coarse pixel, simulated annealing becomes feasible because it is applied to one block at a time instead of the full domain. Optimizing M systems of F points is significantly faster than optimizing one large system of $N = F \times M$ points.

With the SNESIM algorithm there is no explicit variogram to be reproduced, the multiple-point structural model being that recorded in the search tree. The compliance of a simulated data event with the structural model is assessed by looking for replicates of that data event in the search tree. If the simulated data event is not found then the farthest away node is dropped until at least one replicate of the

reduced data event is found. When artifact noise is generated in the course of a simulation but is not present in the training image, replicates of the conditioning data events containing such noise may not be found in the search tree. Thus, the noise reduction issue is linked with the issue of minimizing the number of dropped nodes.

Dropping nodes from a specific data event, say $D(v_i)$, indicates that the corresponding class label combination is not found in the training image. The pixels within $D(v_i)$ that cause nodes to be dropped will also affect other data events $D(v_j)$ in the vicinity of v_i. Changing a pixel class label to decrease the number of dropped nodes for $D(v_i)$ may not be effective since this may increase the number of dropped nodes in surrounding data events. One should aim at reducing the total number of dropped nodes within each coarse pixel.

The total number of dropped nodes within a coarse pixel V is computed by summing all the dropped nodes for all data events with their centers within that coarse pixel. Note that only the center of the data event needs to be inside the coarse pixel; the data event may extend on neighboring coarse fraction pixels. The computation of the objective function is shown in Algorithm A.1.

Algorithm A.1 Computing the objective function of a coarse pixel

1: Set the total dropped nodes to 0
2: **for** each node v_v within coarse pixel V **do**
3: Get the data event $D(v_v)$
4: Get the number of dropped nodes $n_{D(v_v)}$ for $D(v_v)$ from the search tree
5: Increment the total number of dropped nodes
6: **end for**

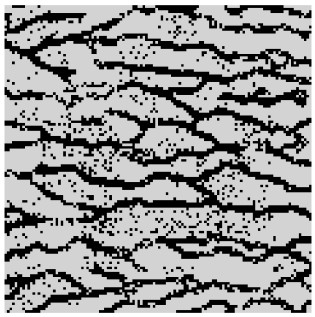
Before correction, the image must already reproduce the coarse proportions, that is any coarse pixel already should contain the correct number of each class label. For a given coarse pixel V, the number of dropped nodes is calculated. The annealing algorithm then swaps the labels at any two fine scale pixels located within the coarse pixel V. The number of dropped nodes is re-computed. If the post-swap number of dropped nodes is lesser than before the swap, the swap is accepted. Otherwise, that swap is only accepted with a certain probability given by an annealing schedule (Geman and Geman, 1984; Deutsch, 1992). It is easier to accept such swap at the beginning than at the end of the annealing process. The swapping continues until a

stopping criterion is met. That criterion is based on a maximum number of iterations and a minimum acceptable number of dropped nodes.

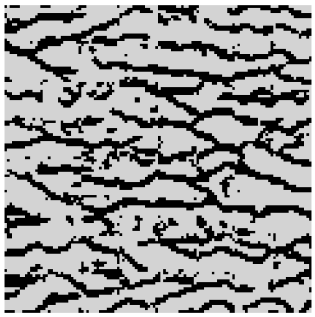
Algorithm A.2 Post-processing with simulated annealing

```
1: Generate a downscaled realization.
2: Choose a random path for visiting several times the N coarse pixels
3: for each node v_v along the path do
4:    Calculate the initial objective function O
5:    while  Objective function is greater than a threshold do
6:       Randomly swap two fine resolution pixels with different label
7:       Compute the post-swap objective function O'
8:       if O' < O or p U(0,1) < Annealing schedule then
9:          Accept the swap (O = O')
10:      else
11:         Refuse the swap
12:      end if
13:      Update the annealing schedule
14:   end while
15: end for
16: Repeat with another super-resolution realization
```

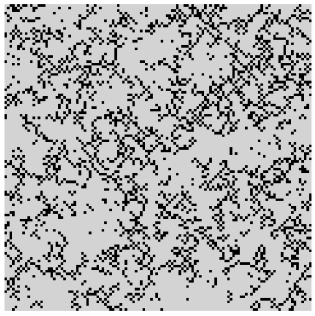
The simulated annealing algorithm is applied to remove noise on Figures 5.4 (a) and (c) and Figure 5.5 (a) of Chapter 5. The initial and post-processed images are displayed in Figure A.1. Note the decrease of noise artifact in the post-processed images.



(a) Downscaled fine spatial map from Case #1

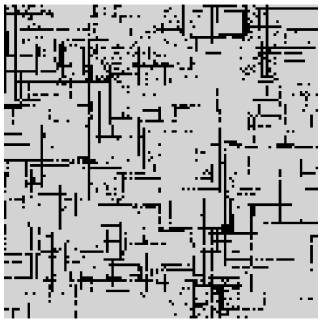


(b) Post-processing of (a) with simulated annealing

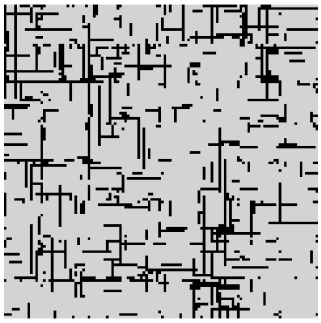


(c) Downscaled fine spatial map from Case #2

(d) Post-processing of (c) with simulated annealing



(e) Downscaled fine spatial map from Case #3



(f) Post-processing of (e) with simulated annealing

Figure A.1: Post-processing of Case #1 to #3 from Chapter 5.

Appendix B

Algorithm-driven and Representation-driven Random Function

This chapter makes the case for a new look at random function, one that is consistent with current and successful geostatistics practice. Two classes of random functions (RF) are presented defined either by an algorithm, termed algorithm-driven RF or explicitly through a set of (data-conditioned) representations, termed representation-driven RF. These new types of RF are alternatives to the classical analytical and stick closer to their actual utilization for modeling and imaging the spatial distribution of a natural phenomenon. The proposed RFs accepts that algorithm implementation is critical and that geostatistical simulations do not aim at visualization but at providing input to further processing by a transfer function such as a flow simulator.

B.1 Introduction

Stochastic methods are a proven and efficient tool to model our knowledge and simulataneously our uncertainty about a spatially distributed phenomena (Journel and Huijbregts, 1978; Chilès and Delfiner, 1999; Olea, 1999; Deutsch, 2002; Wackernagel, 1995; Goovaerts, 1997; Lantuejoul, 2002; Caers, 2005). For instance, from the information provided by a few wells with a sampling volume far less than 1 ‰ of the total reservoir volume,

the permeability distribution in a petroleum reservoir is highly uncertain, in particular the spatial locations of critical high and low values are poorly known. The same remark applies to mining and environmental applications. Therefore, any numerical representation of that phenomenon must account for uncertainty. In geostatistics, such uncertainty is modeled with a random function (RF), and is quantified through a set of equiprobable realizations of that RF which take the form of alternative numerical models or maps. The random function provides a "structural model" for the spatial relations linking data and unsampled values, while the data allows to anchor the realizations at sample locations.

That both data and structural model contribute to the results of an estimation or simulation process is well understood. What is often forgotten is the impact of the algorithm used to implement the structural model and the conditioning to data. A mathematical model or RF, i.e. one fully defined by a set of equations, does not necessarily lend itself into an efficient algorithm. Often iterations or implicit approximations are necessary, such as taking the first terms of a series or limiting the extent of the type of data utilized.

Implementation of any model calls for an algorithm which carries necessarily specific approximations resulting in non predictable deviations from the model. Also, there may be alternative implementation algorithms each with its own set of approximations, none perfect. When actual data are involved as is the case for all practical situations, the impact of the algorithm is even more critical. The final data-conditioned results, whether kriging map or simulated realizations, depart from the original model because,

1. the data never fit perfectly the model,
2. the algorithm is already imperfect in reproducing the model prior to data and,
3. most critically, that algorithm imposes additional approximations when conditioning the model to the data.

As a consequence, in actual applications, the final results (those actually used) are never exactly as expected from the prior structural model, whether RF or else. However, one can still analyze these final results and reject them if deemed inappropriate (Tarantola, 2005). The problem is that in many instances the impact of the

algorithm overwhelms that of the prior model yet the modeler believes he still deals with the original analytical model with all its clean and convenient properties. The honest solution consists of incorporating the algorithm within the model: the model is what the algorithm delivers! This calls for accepting that no result can be fully predicted or analyzed before-the-fact, that is before the prior model is implemented by the specific algorithm retained.

Instead of striving to produce an algorithm that fit best an ideal but limited-in scope RF model, we suggest to consider the algorithm as part of an implicit RF model then work on the algorithm to improve the model. Equations can be used to build a model but the algorithm implementation remains an integral part of that model and this should be formally recognized.

B.1.1 The Gaussian experience

Consider the simplest of all random function models, the standard multivariate normal model, hereafter called Gaussian model, fully characterized by a single covariance matrix. Even if that covariance matrix is inferred from the consistent normal score transformed data, the same used for estimation and simulation, it is highly unlikely that this Gaussian model is consistent with the data statistics of higher order. Sequential Gaussian simulation (sgsim) (Journel, 1993) gives an excellent example of a pristine analytical RF, fully defined through clean and concise equations (Anderson, 2003), but whose simulated realizations particularly after conditioning to real (not synthetic) data may display significant departures from the expected properties of the prior Gaussian RF.

Hence, this algorithm is "imperfect" due to unavoidable inconsistency between data and prior model. But much more troublesome is the fact that even in the absence of data (the case of non conditional simulation), that algorithm is already imperfect. Consider the non conditional simulation of a vector of N correlated Gaussian values, with N large, say $N = 10^6$, still a small size for a 3D application. The best algorithm for implementing that Gaussian model and generating such simulation would call for the LU decomposition of the N by N covariance matrix (Davis, 1987), a task that would ask for approximations with not fully predictable consequences on the actual statistics of the simulated realizations. In short, the results one gets

is not that specified by the model but that specified by the complex combination { prior model + somewhat inconsistent data + an implementation algorithm with its specific approximations }. We will define in the next section such a combination as an algorithm-driven random function (ADRF). The deviations from the prior model brought by the data and the implementation algorithm, although unpredictable, can be observed on the results, after-the-fact. For example, one can draw many realizations from that algorithm-driven random function and observe their statistics of any order (Deutsch, 1994). In many cases, the deviations from the prior model although substantial would be deemed beneficial because,

1. those deviations go towards aspects of the conditioning data that were not or could not be accounted for by the prior model

2. the algorithm was purposely designed to deviate from unwelcome side effects of the model.

Even if modern computers could handle with great accuracy the LU decomposition of the previous very large covariance matrix (Vargas-Guzman and Dimitrakopoulos, 2002; Dimitrakopoulos and Luo, 2004), one may still prefer using a sequential Gaussian algorithm with limited data neighborhoods and ordinary kriging replacing simple kriging. Indeed the latter algorithm shows greater robustness with regard to local departure of the data from the global stationarity required to infer the large (N by N) covariance matrix. But that sequential Gaussian simulation algorithm brings in its own set of approximations and departure from the clean, albeit restrictive, properties of the Gaussian RF model. In other words, one may prefer the algorithm-driven results of sequential Gaussian simulation to the results of the model-consistent LU algorithm. Some may argue that the numerous approximations of the sequential Gaussian algorithm prevent from predicting the distribution of the simulated results; in particular that distribution would not be anymore Gaussian. The answers to such criticism are multiple:

1. Since the conditioning data most likely do not fit a multivariate Gaussian RF, the results of the more rigorous LU decomposition algorithm are equally non Gaussian and as unpredictable before actually running the algorithm and checking the stats of the results.

2. Why should one be dogmatic about the results being Gaussian given that the prior Gaussian model is but only a model, not unique and known to be inconsistent with the data?

3. Last and not least for applications, sequential simulation over a large field is order of magnitude faster than the LU decomposition approach.

Regarding point (2) above, it serves to recall that a random function and its spatial law (multivariate distribution) is only a model, not some physical reality that one should strive to identify (Dubrule, 1994). A model is, by definition, not unique; its goodness can be judged only by its ability to provide the results expected. For the same data set, different models could be considered for different goals. If a model is to be judged by its results, it is perfectly acceptable to design the implementation algorithm to modify the starting model towards it yielding better results. Last, since there is no practice of a model without an implementation algorithm, and any practical implementation implies some departure from that model, why not accept that a model can only be defined fully through its implementation algorithm, which is exactly the concept of an algorithm-driven random function ?

B.1.2 The need for a new formalism

Should an algorithm be ignored just because the underlying RF cannot receive a concise analytical formulation? The simple fact that an algorithm produces realizations (and these realizations are reproducible) indicates that there is a RF; that RF is simply not analytically known.

This is not a new concept; the modeling of phenomena with high complexity such as climate can only be done with computer-based algorithms. Even the best such model can be seen as a patchwork of assumptions, calling for approximations and educated guesses. The complexity is such that no analytical formulation can represent or capture the non-linear intrication of the different components and parameters of the problem. To force an analytical solution would lead to such over simplifications that the resulting model would no longer be in agreement with the observations not to mention the physics of the phenomenon itself. Algorithms specifically designed for

a particular application provide a custom-made, fine-tuned, framework to integrate more information about the specific process under study.

In most applications, the critical variable is the output decision variable, i.e. the final variable that triggers the set of decisions. That decision variable is a function of the spatially distributed variables plus many other engineering parameters pooled together through a typically non-analytical transfer function, an example of which is a flow simulator. The compound effect of all approximations and assumptions involved and the non linearity of the transfer function cannot be summarized from a set of equations. The geostatistical solution consists of generating alternative numerical models of the spatially distributed variables, then process each such realization through the transfer function. These numerical models are treated as equiprobable realizations of the unknown spatially distributed reality. Geostatistical modeling aims at imaging the spatial process towards a specific goal, that of helping into a specific decision making process. The permeability field in reservoir modeling is not simulated for visualization, but to be processed through a flow simulator. The key is that these input realizations must be equiprobable to ensure an appropriate transfer of the spatial uncertainty depicted by these alternative realizations. If the model is analytical, the implementation must be such that the realizations generated constitute an unbiased sampling of the RF, not an easy requirement to check.

In summary, two practical observations drive the need to define and develop new classes of random functions:

Observation 1 In most applications, implementation modifies in an unpredictable way the properties of the prior model.

Observation 2 The uncertainty of interest is not that related to the spatially distributed attributes (be it gold grades or permeability values). It is the uncertainty related to the decison variable resulting from application of transfer function. That transfer function being complex there is no analytical linkage between the input spatial variable and the output decision variable.

B.2 Definitions

We propose two types of non analytical RF. The first is the algorithm-driven RF (ADRF), it is implicitely defined by an algorithm and its parameters. The second is the representation-driven RF (RDRF) which is explicitely defined by a finite number of numerical representations. The ADRF is build in response to Observation 1; because the results actually used depend so much on the implementation algorithm, that algorithm is made part of the model. The RDRF responds to Observation 2; given that in practice only a limited number of realizations can be processed through the transfer function and the final results depend on these specific realizations, the RF model should make explicit these realizations.

B.2.1 Algorithm-driven RF (ADRF)

An ADRF type, denoted $Z(v)$, is defined by an algorithm $F(\eta; \boldsymbol{\theta})$ parametrized by a vector of parameters $\boldsymbol{\theta}$ and a seed number η. That seed seed number is subsequently used to initiate the pseudo-random number generator. The ADRF contains all the L' realizations $z^{(l)}(v), l = 1, ..., L'$ that can be generated with its algorithm $F(\eta; \boldsymbol{\theta})$. Note that L' can be infinity. Selecting L seed values $\eta_l,\ l = 1..., L$ allows sampling that ADRF by generating L realizations $z^{(l)}(v), l = 1..., L$. We will use the symbolic notation:

$$F(\eta_l; \boldsymbol{\theta}) \mapsto z^{(l)} \tag{B.1}$$

Note that for a given set $\boldsymbol{\theta}$ of parameters values, the probability distribution of the ADRF is that of the seed number η. If that seed number η is uniformly distributed then the corresponding realizations $z^{(l)}(v)$ of the ADRF are equiprobable. We are assuming that the algorithm underlying the ADRF is such that any single realization $z^{(l)}(v)$ is completely identified by η_l, that is it can be completely and exactly retrieved through its seed number.

Two realizations are said to be different if their indices are different $l \neq l'$, even if the values of these two realizations are all equal :

$$\{z^{(l)}(v), v \in S\} \neq \{z^{(l')}(v), v \in S\} \quad \textbf{iff} \quad l \neq l' \tag{B.2}$$

Then, by definition every seed number η_l defines a realization $z^{(l)}(v)$ of that ARDF.

From an ADRF perspective, an algorithm is not developed to ensure a clean implementation of some prior analytical RF, the ADRF replaces or voids the need for such an analytical formulation. Properties of the ADRF can be evaluated a posteriori by generating any number L (possibly very large) of realizations. Whereas the implementation of an analytical RF is validated by comparing its realizations properties to the properties expected from the model, the properties of an ADRF are those of its realizations.

Note that in practice the advantage of having access to a possibly infinite pool of realizations ($L \to \infty$) is moot. As stated above in Observation 2, it is the uncertainty of the decision variable which is relevant. With limited time and computing power only a finite number of specific realizations $z^{(l)}(v), l = 1, ..., L$ can be processed to infer the distribution of the decision variable(s). In the end, what matters are the L realizations drawn and actually processed through the transfer function.

Conditioning realizations of an ADRF

Conceptually, conditioning an ADRF is done by ensuring that its defining algorithm is able to honor all the data provided. The notation (B.1) is then extended to :

$$Z(v|\Omega) = F(\eta; \boldsymbol{\theta}|\Omega) \mapsto \{z^{(l)}(v|\Omega), l = 1, 2, ..., L'\} \tag{B.3}$$

The observation data Ω can be "hard data", e.g. direct measurements of the attributes Z defined over any support volume, or include indirect data related to non-linear functions of mutiple Z-values such as a well test or known production data. Any algorithm that can reproduce to an acceptable level the data retained for conditioning would define its own ADRF. Note that the ADRF formalism requires that $F(\eta; \boldsymbol{\theta}) = F(\eta; \boldsymbol{\theta}|\Omega = \text{null})$ to include non-conditional simulations.

The main purpose of conditioning is to anchor a structural model to known observations. The spatial patterns of the structural model can be visualized by generating a set of unconditional simulations. A good algorithm will minimize the distortion of these spatial patterns when morphing them to match data. Both conditional and non-conditional simulations should share the same structural patterns. This requires

to identify which patterns are of importance for the final result (the decision variable), then define an ADRF that can generate these patterns correctly anchored to data.

Whenever the shape and distribution of patterns vary significantly between unconditional and conditional simulations, then

- either the structural model is deemed inconsistent with the data, then that structural model should be changed which amounts to change the ADRF.
- or the implementation algorithm for data conditioning is not good enough. Hence it should be changed which amounts to change the ADRF.
- or the data are considered only soft and their reproduction can be lax and the original algorithm can be kept which amounts not to change the ADRF.

Note that any algorithm that cannot perform data conditioning is of no use in practice and is only of academic interest.

B.2.2 Representation-driven RF (RDRF)

Instead of considering all the possible realization $\{z^{(l)}(v), l = 1, ..., L'\}$ of an ADRF with L' possibly infinite, we proposed to consider the more practical situation of a RF represented by a finite, although possibly very large, number L of numerical models of the natural phenomenon. The representation-driven RF (RDRF) is explicitely defined by this finite set of L representations:

$$Z(v) \mapsto \{z^{(l)}(v), l = 1, ..., L\} \tag{B.4}$$

As opposed to an ADRF defined by an algorithm $F(\eta; \boldsymbol{\theta})$, the RDRF makes no mention of the generating algorithm algorithm but only of its result, a set of L explicit numerical representations $\{z^{(l)}(v),\ l = 1, ..., L\}$. A realization is then one representation drawn among the finite number L of representations constituting that RDRF. This type of RF is the only one that is fully explicit in the sense that any expert can fully visualize it and decide if it is suitable for the application at hand.

In all generality, the L representations of an RDRF need not be generated by the same algorithm, some can be computer-generated, others can be hand drawn. Again,

the algorithm(s) used to generate the representations are irrelevant; an RDRF should be judged solely from its L constitutive representations irrespective of how they were generated.

Furthermore, it is possible to define a priori the probability of occurence of each of the L realizations. In most applications, however, equal probability will be considered with each realization having probability 1/L. Availability of additional information or expert opinion could be used to give higher or lower probability of occurence to some representations.

The main advantage of the RDRF is that it is unequivocally defined by a finite number L of visually explicit realizations; there is no "what" or "if other realizations" were to be drawn. It also sticks closer to actual practice whereby the input spatial uncertainty is frozen to be that provided by the L realizations retained for processing by the transfer function. The uncertainty is given by the L representations; adding one additional representation amounts to generating another RDRF.

A conditional RDRF is such that all its L constitutive representations honor all data to an acceptable level of exactitude:

$$Z(v|\Omega) = \{z^{(l)}(v|\Omega), l = 1, 2, ..., L\} \tag{B.5}$$

B.3 Validating an ADRF or RDRF

The choice of any random function is necessarily a subjective decision since natural phenomena are not created by any RF nor even by a computer-based algorithm but from past physical processes. To use a probabilistic formulation for a natural phenomena at any scale larger than the atom is to acknowledge our inability to model its underlying physics. The random function model only aims at a direct after-the-fact representation of the (spatial) patterns which are the consequences of the physical phenomenon. That random function model rarely aims at re-creating step-by-step the physical processes.

Whichever model is chosen, analytical RF, ADRF or RDRF, that choice is necessarily subjective. For instance, there is no a priori objective reason to choose a Gaussian RF to model permeability or to retain a training image-based algorithm,

such as snesim (Strebelle, 2002), to model a clastic reservoir. The choice of a particular algorithm or set of L representations is based on many subjective criteria, some being dependent on the particularly transfer function used to process these realizations.

Deutsch (1994) proposes three criteria to select an "algorithmically-defined" RF; a good algorithm must 1) be executed in a reasonable amount of time 2) integrate the maximum of prior information and 3) maximize the entropy of the distribution of the decision variable. These guidelines remains appropriate for the extended ADRF formalism proposed in this paper. However, in any practical application it would be difficult if not impossible to select the algorithm that ensures maximization of the entropy of the distribution of the decision variable, because of the complexity of the non linear transfer function. Instead between two ADRFs or two RDRFs of equal performance as to the two first criteria, it suffices to retain the one that maximizes the entropy of the distribution of decision variable.

From the point of view of structural patterns reproduction, choosing and fine-tuning an algorithm $F(\eta; \boldsymbol{\theta})$ and inferring its parameters $\boldsymbol{\theta}$ can be done by performing non-conditional realizations. The structural model implicit to that generation algorithm can then be visualized and validated in regards to the prior structural expectation and its relevance to the transfer function retained..

The selection of an ADRF and the inference of its parameters are open questions and questions that are case-dependent. A good ADRF should be such that the algorithm parameters can be easily inferred/optimized. This is no different from selecting an analytical RF, with some being easier to infer than others. The useability of an ADRF depends on that ease of inference of its critical parameters.

The building of a RDRF is done by first selecting one or several algorithm(s) to generate the L numerical representations. Each of these representations must honor all data and be a valid image representative of the natural phenomenon. Most of the time, a RDRF is build from a set of L realizations from an ADRF.

The spatial properties of an ADRF or a RDRF can only be characterized a posteriori. The spatial statistics of an ADRF is calculated from any number of realizations, while those of an RDRF are computed from the particular set of L representations that defines it.

B.3.1 ADRF-RDRF and nonlinear transfer function

The usefulness of random functions lies in their ability to model uncertainty through their simulated realizations. However, as mentioned before, the goal is not in the realizations themselves but in their processing through a transfer function. This processing is usually expensive, e.g. through a flow simulator or a mine planning software, and can only performed on a few realizations.

Consider the decision random variable T, obtained by processing the spatial RF $Z(v)$ by the transfer function $\Psi(\cdot)$.

$$T = \Psi(Z(v)) \tag{B.6}$$

In case of a reasonably complex and non linear transfer function $\Psi(\cdot)$ the complete distribution of T calls for processing all the L realizations of $Z(v)$, such that

$$\{t^{(1)}, t^{(2)}, ..., t^{(L)}\} = \{\Psi(z^{(1)}(v)), \Psi(z^{(2)}(v)), ..., \Psi(z^{(L)}(v))\}$$

Hence the sampling of T is directly related to that of $Z(v)$. Irrespective of other realizations that could have been generated with an ADRF or an analytical RF, the distribution of T is unequivocally identified by the previous L realizations $t^{(l)}, l = 1, ..., L$.

To further understand the differences between ADRF and RDRF consider the following hypothetical, yet common, scenario. One wants to estimate the time that a contaminant takes to travel between two wells using a flow simulator: $T = \Psi(Z(v))$, where $Z(v)$ is the permeability field. With a specific algorithm $F(\eta|\Omega; \boldsymbol{\theta})$ L representations are generated $\{z^{(1)}(v), ..., z^{(L)}(v)\}$ and processed by the flow simulator into L travel times $\{t^{(1)}, ..., t^{(L)}\}$. To check the robustness of this latter distribution, L more simulations $\{z^{(L+1)}(v), ..., z^{(2L)}(v)\}$ are generated and processed yielding a second distribution $\{t^{(L+1)}, ..., t^{(2L)}\}$.

With an ADRF model, one can evaluate the robustness of the first distribution $\{t^{(1)}, ..., t^{(L)}\}$. If the second distribution $\{t^{(L+1)}, ..., t^{(2L)}\}$ is deemed significantly different from that first one, then one can conclude that the first L realizations of $Z(v)$ were not sufficient to characterize the consequent distribution of the contaminant travel time T. Note that both set of realizations are coming from the same ADRF.

The interpretation is different when using the RDRF concept: one would be comparing the outputs of two different RDRFs, the first one with L realizations $\{z^{(1)}(v), ..., z^{(L)}(v)\}$, and a second one with the L different realizations $\{z^{(L+1)}(v), ..., z^{(2L)}(v)\}$. One would like these two RDRFs to yield similar distributions for the travel time T. By merging the two sets of realizations into one of size $2L$, one define yet a third RDRF. Everytime a new representation $z^{(l)}(v)$ is added to an existing set of representations, a new RDRF is being considered. Note also by adding L' representations to the original L realizations the equal likelihood of each realization decreases from $1/L$ to $1/(L + L')$.

B.4 Discussion

This paper has proposed a formalism for creating new random functions that are more consistent with the current and foreseeable practice in geostatistics. Instead of developing an algorithm that tries to mimic the properties of an analytical probabilistic model, the algorithm-driven RF (ADRF) includes in its definition the implementation algorithm while the representation-driven RF (RDRF) is characterized by a finite set of explicit numerical representations. Both ADRF and RDRF provide equiprobable realizations by definition, this is not an assumption but a property of the new random functions.

We stress that these random functions are neither a replacement for the more traditional analytical RF, nor an excuse for producing adhoc algorithms without a solid understanding of theory. They are valuables alternatives to the RF models. Instead of judging an algorithm in terms of it approximating an analytical formulation, an algorithm should be judged based on its output which are the realizations generated. An algorithm that fails to reproduce critical features is a bad one no matter the pedigree of the underlying model.

Of the two models proposed, ADRF and RDRF, the former is closest to the classical concept of a random function but with algorithms and computer codes replacing the equations. It also suffers a drawback shared by analytical RFs, that is providing a potentially large pool of possible realizations which can never be fully explored.

The RDRF model is more atuned to utilization through a transfer function, as

no claim is made about the existence of a possibly infinite numbers of alternative realizations.

B.4.1 Open source software and ADRF

Since an ADRF includes in its definition the implementation algorithm, that algorithm including its computer code should be supplied. The inspection and testing of the code should be done carefully and is as necessary as the understanding of the theory underlying a traditional RF.

We strongly believe that open source softwares are necessary with these type of RF. Not disclosing the code is analogous of not writing down the equations of an analytical RF. It is only by having access to the code that one can fully appreciate the assumptions and simplifications that were implemented.

B.5 Conclusion

The formalism of algorithm-driven and representation-drive RF is in line with the current practice and research in geostatistics. With the focus of these RF weighted toward the end-results, that is the simulated realizations, these two RF classes provides more latitude in developing algorithms. It also has the potential to lessen the grip of Gaussian-related RF in the modeling of natural phenomenon. Above all, it recognizes that the only component of a RF that is actually used is the realizations that have been actually drawn.

Bibliography

Anderson, T. W.: 2003, *An Introduction to Multivariate Statistical Analysis*, 3^{rd} edn, John Wiley & Sons, New York.

Atkinson, P. and Graham, A.: 2006, Issues of scale and uncertainty in the global remote sensing of disease, *Advances in Parasitology* **62**, 79–118.

Atkinson, P. M.: 2001, Super-resolution target mapping from soft-classified remotely sensed imagery, *Proceedings of the 6th International Conference on Geocomputation*. Brisbane, Australia: University of Queensland.

Atkinson, P. M.: 2004, Super-resolution land cover classification using two-point histogram, *in* X. Sanchez-Villa, J. Carrera and J. Gomez-Hernadez (eds), *geoENV IV Geostatistics for Environmental Applications*, Springer, pp. 15–28.

Atkinson, P. M. and Curran, P. J.: 1995, Defining an optimal size of support for remote-sensing investigations, *IEEE transactions on geoscience and remote sensing* **33**(3), 768 – 776.

Atkinson, P. M., Cutler, M. E. J. and Lewis, H.: 1997, Mapping sub-pixel proportional land cover with AVHRR imagery., *International Journal of Remote Sensing* **18**(4), 917 – 35.

Bachmann, A. and Allgöwer, B.: 2002, Uncertainty propagation in wildland fire behaviour modeling, *International Journal of Geographical Information Science* **16**(2), 115–127.

Benediktsson, J. A. and Swain, P. H.: 1992, Consensus theoretic classification methods., **22**(4), 688 – 704.

Bertero, M.: 1985, Linear inverse problems with discrete data. i: General formulation and singular system analysis, *Inverse Problems* **1**, 301–330.

Bordley, R. F.: 1982, A multiplicative formula for aggregating probability assessments, *Management Science* **28**(10), 1137–1148.

Boucher, A. and Kyriakidis, P.: 2007, Integrating fine scale information in super-resolution land cover mapping, *Photogrametry and Remote sensing* . Accepted.

Boucher, A. and Kyriakidis, P. C.: 2006, Super-resolution land cover mapping with indicator geostatistics, *Remote sensing of environment* **104**(3), 264 – 282.

Boucher, A., Seto, K. C. and Journel, A. G.: 2006, A novel method for mapping land cover changes ; integrating space and time with geostatistics, *IEEE transactions on geosciences and remote sensing* **44**(11), 3427–3435.

Caers, J.: 2005, *Petroleum geostatistics*, Society of Petroleum Engineers.

Caers, J. and Hoffman, T.: 2006, The probability perturbation method: A new look at bayesian inverse modeling, *Mathematical Geology* **38**(1), 81–100.

Carle, S. F. and Fogg, G. E.: 1996, Transition probability-based indicator geostatistics, *Mathematical geology* **28**(4), 453 – 476.

Chilès, J. and Delfiner, P.: 1999, *Geostatistics: Modeling spatial uncertainty*, John Wiley & Sons, New York.

Crosetto, M., Ruiz, J. A. and Crippa, B.: 2001, Uncertainty propagation in models driven by remotely sensed data., *Remote Sensing of Environment* **76**(3), 373 – 385.

Davis, M.: 1987, Production of conditional simulations via the lu decomposition of the covariance matrix, *Mathematical Geology* **19**(2), 91–98.

Deutsch, C. and Journel, A.: 1998, *GSLIB Geostatistical Software Library and user's guide, 2nd Edition*, 2^{nd} edn, Oxford University Press, New York.

Deutsch, C. V.: 1992, *Annealing techniques applied to reservoir modeling and the integration of geological and engineering (well test) data*, PhD thesis, Stanford University, Stanford, CA.

Deutsch, C. V.: 1994, Algorithmically-defined random function models, *in* Dimitrakopoulos (ed.), *Geostatistics for the Next Century*, Kluwer, Dordrecht, Holland, pp. 422–435.

Deutsch, C. V.: 2002, *Geostatistical Reservoir Modeling*, Oxford, New York.

Dimitrakopoulos, R. and Luo, X.: 2004, Generalized sequential gaussian simulation, *Mathematical Geology* **36**, 567–591.

Dubrule, O.: 1994, Estimating or choosing a geostatistical model?, *in* R. Dimitrakopoulos (ed.), *Geostatistics for the Next Century*, Kluwer, pp. 3–14.

Fragkias, M. and Seto, K. C.: 2007, Modeling urban growth in data-sparse environments: A new approach, *Environment and Planning B* . In press.

Geman, S. and Geman, D.: 1984, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-6**(6), 721–741.

Gomez-Hernandez, J.: 1992, Regularization of hydraulic conductivities: A numerical approach, *in* A. Soares (ed.), *Geostatistics Troia '92*, Kluwer academic, Troia, pp. 767–778.

Gonzalez, R. C. and Woods, R. E.: 2002, *Digital Image Processing*, Prentice Hall, New Jersey.

Goovaerts, P.: 1997, *Geostatistics for natural resources evaluation*, Oxford University Press, New York.

Hansen, T. M., Journel, A. G., Tarantola, A. and Mosegaard, K.: 2006, Linear inverse gaussian theory and geostatistics, *Geophysics* **71**(6), R101–R111.

Hastie, T., Tibshirani, R. and Friedman, J.: 2001, *The Elements of Statistical Learning*, Springer, New York.

Johnson, M.: 1987, *Multivariate Statistical Simulation*, John Wiley & Sons, New York.

Journel, A. G.: 1983, Nonparametric-estimation of spatial distributions, *Journal of the internation association for mathematical geology* **15**(3), 445 – 468.

Journel, A. G.: 1993, Geostatistics: roadblocks and challenges, *in* A. Soares (ed.), *Geostatistics-Troia*, Vol. 1, Kluwer Academic Publ., Dordrecht, pp. 213–224.

Journel, A. G.: 1997, Deterministic geostatistics: A new visit, *in* E. Baffi and N. Shofield (eds), *Geostatistics-Wollongong*, Vol. 1, Kluwer Academic Publ., Dordrecht, pp. 174–187.

Journel, A. G.: 2002, Combining knowledge from diverse sources: An alternative to traditional data independence hypotheses, *Mathematical Geology* **34**(5), 573 – 596.

Journel, A. G. and Alabert, F.: 1989, Non-Gaussian data expansion in the earth sciences, *Terra Nova* **1**, 123–134.

Journel, A. G. and Huijbregts, C. J.: 1978, *Mining Geostatistics*, Academic Press, New York.

Journel, A. and Posa, D.: 1990, Characteristic behavior and order relations for indicator variograms, *Mathematical Geology* **22**(8), 1011–1025.

Jupp, D. L. B., Strahler, A. H. and Woodcock, C. E.: 1988, Auto-correlation and regularization in digital images .1 basic theory, *IEEE transactions on geoscience and remote sensing* **26**(4), 463 – 473.

Kaipio, J. and Somersalo, E.: 2004, *Statistical and Computational Inverse Problems*, Spinger-Verlag.

Kasetkasem, T., Arora, M. and Varshney, P.: 2005, Super-resolution land cover mapping using a markov random field approach, *Remote Sensing of Environment* **96**, 302 – 314.

Krishnan, S., Boucher, A. and Journel, A. G.: 2005, Evaluating information redundancy through the tau model, *in* O. Leuangthong and C. Deustch (eds), *Geostatistics Banff 2004*, Vol. 2, Springer, pp. 1037 – 1046.

Kyriakidis, P. C. and Yoo, E. H.: 2005, Geostatistical prediction and simulation of point values from areal data, *Geographical Analysis* **37**(2), 124–151.

Lantuejoul, C.: 2002, *Geostatistical Simulation: Models and Algorithms*, Springer-Verlag, Berlin, Germany.

Makido, Y. and Shortridge, A.: 2005, Land cover mapping at sub-pixel scales:unraveling the mixed pixel, Geocomputation.

Mao, S. and Journel, A.: 1998, Conditional 3d simulation of lithofacies with 2d seismic data, *Technical report*, SCRF report, Stanford University.

Marcotte, D.: 1996, Fast variogram computation with FFT, *Computers & Geosciences* **22**(10), 1175 – 1186.

Menke, W.: 1989, *Geophysical Data Analysis: Discrete Inverse Theory*, Academic Press San Diego. Revised Edition.

Mertens, K. C., Verbeke, L. P. C., Ducheyne, E. I. and Wulf, R. R. D.: 2003, Using genetic algorithms in sub-pixel mapping, *International Journal of Remote Sensing* **24**(21), 4241 – 4247.

Mertens, K. C., Verbeke, L. P. C., Westra, T. and Wulf, R. R. D.: 2004, Sub-pixel mapping and sub-pixel sharpening using neural network predicted wavelet coefficients, *Remote Sensing of Environment* **91**(2), 225 – 236.

Mosegaard, K. and Tarantola, A.: 1995, Monte Carlo sampling of solutions to inverse problems, *Journal of Geophysical Research* **100**(B7), 12431 – 12447.

Muslim, A. M., Foody, G. M. and Atkinson, P. M.: 2006, Localized soft classification for super-resolution mapping of the shoreline, *International Journal of Remote Sensing* **27**(11), 22712285.

Olea, R. A.: 1999, *Geostatistics for Engineers and Earth Scientists*, Kluwer Academic Publishers.

Pardo-Igzquiza, E., Chica-Olmo, M. and Atkinson, P. M.: 2006, Downscaling cokriging for image sharpening, *Remote Sensing of Environment* **102**, 8698.

Richards, J. A. and Jia, X.: 1999, *Remote sensing digital image analysis*, 3 edn, Springer-Verlag, Berlin.

Sambridge, M. and Mosegaard, K.: 2002, Monte Carlo methods in geophysical inverse problems, *Reviews of Geophysics* **40**(3), 1–29.

Seto, K. C. and Fragkias, M.: 2005, Quantifying spatiotemporal patterns of urban land-use change in four cities of china with time series landscape metrics, *Landscape Ecology* **20**(7), 871–888.

Seto, K., Woodcock, C., Song, C., Huang, X., LU, J. and Kaufmann, R.: 2002, Monitoring land-use change in the pearl river delta using landsat tm, *Int. J. Remote sensing* **23**(10), 1985–2004.

Soares, A.: 1998, Simulation with correction for local probabilities, *Mathematical Geology* **30**(6), 761–765.

Strebelle, S.: 2002, Conditional simulation of complex geological structures using multiple-point statistics, *Mathematical Geology* **34**(1), 1–21.

Tarantola, A.: 2005, *Inverse problem theory and methods for model parameter estimation*, Society for Industrial and Applied Mathematics, Philadelphia.

Tatem, A. J., Lewis, H. G., Atkinson, P. M. and Nixon, M. S.: 2001, Super-resolution target identification from remotely sensed images using a Hopfield neural network., *IEEE Transactions on Geoscience and Remote Sensing* **39**(4), 781 – 96.

Tatem, A. J., Lewis, H. G., Atkinson, P. M. and Nixon, M. S.: 2002, Super-resolution land cover pattern prediction using a Hopfield neural network., *Remote Sensing of Environment* **79**(1), 1 – 14.

Tatem, A. J., Lewis, H. G., Atkinson, P. M. and Nixon, M. S.: 2003, Increasing the spatial resolution of agricultural land cover maps using a Hopfield neural network., *International Journal of Geographical Information Science* **17**(7), 647 – 72.

Tso, B. and Mather, P.: 2001, *Classification methods for remotely sensed data*, Taylor and Francis, London.

Vargas-Guzman, J. and Dimitrakopoulos, R.: 2002, Successive conditional simulation of random fields by residuals, *Mathematical Geology* **34**(5), 597–611.

Verhoeye, J; De Wulf, R.: 2002, Land cover mapping at sub-pixel scales using linear optimization techniques, *Remote Sensing of Environment* **79**(1), 96–104.

Viczek, T.: 1989, *Fractal growth phenomena*, World Scientific Publishing Co, Singapore.

Wackernagel, H.: 1995, *Multivariate Statistics*, Springer-Verlag, Berlin, Heidelberg.

Yao, T. T. and Journel, A. G.: 1998, Automatic modeling of (cross) covariance tables using fast fourier transform, *Mathematical Geology* **30**(6), 589–615.

Zhang, T., Journel, A. G. and Switzer, P.: 2006, Filter-based classification of training image patterns for spatial simulation, *Mathematical Geology* **38**(1), 63–80.