UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERIA DE MINAS

TRAINING IMAGE SELECTION AND MODEL VALIDATION USING MULTIPLE POINT STATISTICS

TESIS PARA OPTAR AL GRADO DE MAGISTER EN MINERIA

CRISTIAN MARCELO PÉREZ STRUTZ

PROFESOR GUÍA:
JULIÁN M. ORTIZ CABRERA

MIEMBROS DE LA COMISIÓN
XAVIER EMERY
EDUARDO MAGRI VARELA
GREGOIRE MARIETHOZ

SANTIAGO DE CHILE
JUNIO 2013

ABSTRACT

Knowledge of geological properties is critical for accurate resource estimation. A nice feature of conventional variogram based techniques is that they allow inferring the spatial structure solely based on the available drill-hole data. These techniques allow generating probabilistic models that quantify the associated uncertainty, however only a second order spatial structure is inferred and imposed, not taking into account the correct reproduction of complex spatial continuities that are often observed in geological settings.

In order to overcome this limitation, pattern simulation algorithms have been developed over the last few years. Multiple-point simulation algorithms require a spatially exhaustive source of patterns, called training image, which provides a reasonable number of high order event replicates, allowing a representative inference of pattern statistics. This source corresponds generally to a conceptualization of the geological phenomenon that originates the spatial structure of the variable. However, preparation of training images is often based on subjective criteria; different experts could construct different training images guided by their personal interpretation of geology. As a consequence, selection of an appropriate training image is one of the main issues a modeling professional must face when using multiple-point simulation. It can be considered even more important than variogram modeling for classical geostatistical simulation as it controls higher-order as well as second order spatial relations.

This work addresses the development of a geostatistical tool that provides measurable criteria for training image selection. It allows ranking training images according to their relative compatibility and also obtaining an absolute compatibility measure to a data set, both in terms of the spatial structure of the variable considered. The algorithm achieves the first goal by calculating the conditional probability of finding patterns in the different training images subject to the condition of patterns being contained in the training image set available. The second goal is achieved by calculating the proportions of data patterns contained in a determined image. Two alternative pattern search approaches are developed. The first one computes the complete probability distribution function of each pattern in each training image. This method is statistically sound but computationally very demanding. The second approach, based on direct sampling allows obtaining similar results at a less computational cost.

The developed pattern search algorithms are finally modified in order to generate a multiple-point simulation results validation tool which allows computing the quality and completeness of training image pattern reproduction both expressed respectively in terms of the proportion of results patterns found in the training image and the proportion of training image patterns that are effectively reproduced by the results.

The applicability of the methodologies is successfully evaluated in three different 2-D and 3-D hydrology and mining case studies constructed with synthetic and real databases of categorical variables. The results validation tool is tested on a real mine-scale lithologies reconstruction case study at the Escondida Norte deposit. It turns out that the methods have applicability beyond the field of geostatistics and can be used as an aid to geological interpretation.

RESUMEN EJECUTIVO

La correcta caracterización de propiedades geológicas es esencial para la industria minera debido a su influencia en distintos procesos, especialmente en la estimación de recursos. Una característica positiva del variograma es su capacidad de inferir la continuidad espacial de la variable analizada a partir de los datos condicionantes, los cuales corresponden generalmente a datos de sondajes. Técnicas geoestadísticas convencionales basadas en el variograma permiten construir modelos probabilísticos y cuantificar la incertidumbre asociada a éstos. Sin embargo, estas técnicas son incapaces de reproducir estructuras espaciales complejas, frecuentemente observadas en atributos geológicos, debido a la inferencia de continuidad espacial de segundo orden realizada mediante el variograma.

Los algoritmos de simulación de patrones han sido desarrollados en los últimos años para superar este problema. Estos algoritmos requieren de una fuente exhaustiva de patrones, conocida como imagen de entrenamiento, a partir de la cual pueda ser realizada una inferencia representativa de estadísticas de patrones. La imagen de entrenamiento corresponde generalmente a un modelo conceptual del fenómeno geológico que genera la estructura espacial de la variable por lo que su construcción se ve frecuentemente sometida a criterios subjetivos. Distintos especialistas podrían incluso construir imágenes distintas basados en su interpretación personal de la geología. Por estos motivos es que la selección y construcción de la imagen de entrenamiento se ha convertido en una de las principales dificultades al implementar en la práctica estos nuevos algoritmos de simulación.

Este trabajo aborda el desarrollo de una herramienta geoestadística capaz de proveer criterios medibles para la selección de una imagen de entrenamiento. La herramienta permite generar un ranking de imágenes de entrenamiento de acuerdo a su compatibilidad espacial relativa y absoluta con los datos condicionantes. El algoritmo logra obtener la compatibilidad relativa mediante el cálculo de la probabilidad condicional de encontrar patrones de datos condicionantes en una determinada imagen de entrenamiento, sujeto a la condición de que los patrones se encuentren contenidos en el conjunto de imágenes disponibles. La compatibilidad absoluta es obtenida mediante el cálculo de la proporción de patrones de datos condicionantes contenidos en una determinada imagen. Se desarrollaron dos estrategias de cálculo de compatibilidad. La primera, sólida desde el punto de vista estadístico, requiere del cálculo explícito de la distribución de probabilidad condicional, mientras que la segunda, basada en un muestreo directo, permite obtener resultados similares a un menor costo computacional.

El algoritmo de búsqueda de patrones es finalmente modificado para generar una herramienta que permita validar resultados de simulaciones en términos de su calidad e integridad, expresados respectivamente en términos de la proporción de patrones de resultados encontrados en la imagen y la proporción de patrones de la imagen encontrados en los resultados.

La aplicabilidad de las herramientas desarrolladas se evaluó satisfactoriamente en casos de estudio generados con datos sintéticos y reales. Se puede determinar que los métodos son aplicables más allá del campo de la geoestadística y que pueden ser usados como ayuda para la interpretación y el modelamiento geológico.

A Annemarie, Hildegard y Tamara

AGRADECIMIENTOS

En primer lugar quisiera agradecer a mi madre, por todo el apoyo y ayuda recibido por ella durante todos los años de estadía en la universidad. Muchas gracias por todo el esfuerzo que has hecho por mí y todo el cariño que me has entregado. También quisiera agradecer al resto de los integrantes de mi familia, abuelos, tíos y primos, ustedes también han jugado un rol fundamental en mi desarrollo como persona que me ha permitido llegar acá.

A mi polola Tamara, quien ya me ha acompañado, ayudado y apoyado hace más de cuatro años. Te amo mucho mi amor y espero que esta compañía sea para el resto de nuestras vidas.

A mis amigos y compañeros del colegio. Muchas gracias, Marcel, Bernardo, Tomás, Cristóbal, Sebastián, Gabriel, Germán, Klaus y Christian.

A mis compañeros de la Universidad, especialmente a los mineros compañeros de generación. Las incontables clases, noches de estudio y actividades extra académicas (viaje a Brasil, partidos de fútbol, CAM 2008, entre muchas otras) marcaron nuestra estadía en la facultad y transformaron, al menos para mí, la época universitaria en una de las mejores de mi vida. Muchas gracias Javier M., Werner, Javier D., Winston, Felipe, Ignacio, Jorge S., Jorge C., Eduardo, Agustín, Cristian, Bernabé, Antonio, Diego, Jimmy, Matías, Sebastián, Luis Felipe, Miguel, Daniela, Montserrat, Constanza, Milivoj, Luis.

A Julián Ortiz, permitir especializarme en el área de geoestadística y ser un excelente docente y persona. Gracias por permitirme realizar este magíster y responder siempre con la mejor disposición la infinidad de preguntas relacionadas al tema u otros temas geoestadísticos. Gracias por darme la oportunidad de realizar el viaje a Australia para completar esta tesis.

I would like to thank the National Centre for Groundwater Research and Training of the University of New South Wales for supporting financially my stay in Australia. I would also like to thank Gregoire Mariethoz, your help was fundamental for the development of this thesis and for my stay in Australia. Without it I would not have been able to complete this and other related works. I would like to extend these acknowledgements to Bryce Kelly, Alessandro Comunian and Dane Burkett.

A Paula Larrondo, Eduardo Magri, Xavier Emery y a los integrantes del Laboratorio Alges de la Universidad de Chile. Ustedes han tenido un rol importante en mi desarrollo profesional y por lo tanto en el desarrollo de este magíster.

A Conicyt que financió esta memoria por medio del proyecto Fondecyt 1090056 y a Codelco por patrocinar la Cátedra de Evaluación de Yacimientos. Finalmente quisiera agradecer a Minera Escondida Limitada por facilitar datos con los cuales se realizó parte de este trabajo. Muchas gracias a todos ustedes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF TABLES

1. INTRODUCTION

Geostatistics is a science that addresses the study of regionalized phenomena, this means, phenomena that extend in space and present in greater or lesser extent a spatial organization or structure. It is applicable to various fields, as for example, the study of grades distribution in mining deposits (Ortiz and Emery 2006; Ortiz et al. 2012), atmospheric contaminant dispersion (Zhang et al. 2012), rainfall (Goovaerts 2000; Kyriakidis et al. 2001; Yates et al. 2007), reconstruction of past climates (Mariethoz et al. 2012), concentration of nutrients in agricultural soils (Emery and Ortiz 2007) and permeability modeling of underground aquifers (Zimmerman et al. 1998; Alcolea et al. 2009; Mariethoz et al. 2009) or oil reservoirs (Zhang et al. 2009), among others. Generally, the main objective of the constructed models is to accurately represent a certain reality with the minimum possible error while quantifying the associated uncertainty.

Geological and grades modeling, as well as quantifying the related uncertainty and forecasting ability related to the generated models, are essential for proper evaluation and development of a mining project due to various reasons:

- It allows generating quality grade models that can support mine planning to obtain reliable technical and economical assessments of the project.
- It permits the construction of stochastic simulated models of grades and geology, to perform risk analysis of the project, studying the impact of values fluctuations of certain key variables.
- Grades modeling can be improved due to correct geological modeling, as geological units support the spatial distribution of grades, especially in cases where significant structural control of the mineralization is observed.
- It allows studying the geotechnical quality (hardness, presence of structures) of the rock, which is essential to underground or open pit mine design.
- A good geological model helps defining correctly the metallurgical process to be applied to a certain mined rock volume, as well as it permits calculating the consumption of reagents, which is usually done as a function of geological variables.

Currently, grade models (generally continuous variables) are usually built using estimation techniques (nearest neighbor, inverse distance or some type of kriging). With increasing frequency in recent years, conventional geostatistical simulation techniques have been used to construct stochastic models. To this end, a number of techniques have been developed and most of them rely on the available sparse data (such as for example drill-holes) to infer the spatial continuity of the variable. This is accomplished using statistics

based on relations between pairs of points as for example variograms or correlograms (Journel and Huijbregts 1978; Isaaks and Srivastava 1989; Deutsch and Journel 1992; Dimitrakopoulos and Dagbert 1993; Caers 2005). These conventional 2-point simulation methods are relatively simple and generally show good results. However, the main problem or limitation of these types of techniques is that the inference of complex spatial continuities is not possible, although these are common in this kind of regionalized variables.

Geological models (categorical variables generally) are often constructed using deterministic techniques, meaning that their construction relies on the knowledge and experience of a specialist that assigns the geological attribute to a given volume. This method prevents the evaluation of the uncertainty associated with the generated models. There are techniques available that allow generating probabilistic models of categorical variables, however, as in the case of continuous variables, these techniques interpret the real distribution of the variables as the realization of a random function, inferring the spatial structure and continuity from spatial relations based on pairs of points, with the same disadvantages and limitations that were explained in the previous paragraph.

As an alternative to conventional geostatistical simulation methods, multiple-point simulation algorithms, also known as training image based methods, have recently become an important point of focus, with a wealth of different methods developed in the last years (Guardiano and Srivastava 1993; Strebelle 2002; Zhang et al. 2006; Arpat and Caers 2007; Eskandaridalvand 2008; Gloaguen and Dimitrakopoulos 2009; Honarkhah and Caers 2010; Mariethoz et al. 2010; Peredo and Ortiz 2011; Parra and Ortiz 2011; Straubhaar et al. 2011; Tahmasebi et al. 2012). These algorithms infer the spatial continuity of the variable using higher order statistics, i.e. statistics based on spatial relations between more than 2 locations at the time, and therefore they do not use variographic models to impose a spatial structure (Journel and Zhang 2006). These structures are directly inferred from a training image, which corresponds to a conceptual model and exhaustive representation of the geological phenomenon that generates them.

To date, much effort has been focused on developing multiple-point simulation algorithms with improved efficiency and usability and applying them to real cases. However, significant developments are lacking when it resorts defining objective criteria for training image selection or to generate tools for measuring the compatibility between the training image and the available scattered conditioning data. A notable exception is the use of spatial cumulants as high-order alternative to variograms (Dimitrakopoulos et al. 2010). There is additionally a lack of tools that allow the validations of results through the measure of the similarity between the stochastic generated models and the training image in terms of high order spatial structure. This means that it cannot be determined if the patterns found in the training image are correctly reproduced by the simulated models.

Problems related to the selection of training images and results validation that are not addressed in this thesis, are for example the existence of significant differences between the variables distribution of the training image and the conditioning data and training image scaling issues (Ortiz, Lyster and Deutsch 2007).

This work addresses the development of geostatistical tools that provide measurable criteria for training image selection and simulated model validation in the context of multiple-point simulation. Two different training image selection tools are developed. The first one, applicable in cases where more than one conceptual model is available, allows ranking training images according to their compatibility with a data set in terms of low and high order spatial structures of the regionalized variable. This represents a relative compatibility measurement, as even a top ranked training image might be incompatible in absolute terms if all the available images are not compatible. To overcome this shortcoming, an absolute compatibility measurement method is developed, which computes the probability of finding conditioning data patterns in a given training image. Regarding simulated model validation, the results validation tool allows determining if the spatial structure of the training image is kept by the simulated models.

The structure of the work consists of a review of relevant literature to the theme of research to know and understand the state of the art related to available multiple-point simulation, training image selection, and results validation methods. It continues with the development of the proposed tools in order to proceed with the validation of the training image selection tool using synthetic and real data sets.

Finally, the conditioning data event search algorithm, originally developed for training image selection, is modified in order to generate a results validation tool. The tool is able to compute the proportion of results patterns that are contained in the training image, as a measure of model quality, and the proportion of training image patterns that are reproduced by the resulting model, as a measure of results completeness. The simulations results validation tool is tested in the context of a real mine-scale modeling case study of the Escondida Norte deposit located in northern Chile, in which the performance of the Direct Sampling multiple-point simulation algorithm is compared to sequential indicator simulation.

1.1. Motivation

As the amount of higher order events contained in scattered data (in the mining case generally drill-hole data) is usually insufficient, multiple-point simulation algorithms require an exhaustive source of patterns that provides a reasonably large number of event replicates to allow a representative inference of spatial pattern statistics. This source corresponds generally to a conceptualization of the geological phenomenon that originates the spatial structure of the variable, called training image. Because training images are generally conceptual models, their preparation is often based on subjective criteria of the modeling expert. Different experts could even construct completely different training images based on their personal interpretations of the geological phenomenon. Alternatively, process-based mechanisms have been developed to generate training images that have realistic geological features (Pyrcz, Catuneanu and Deutsch, 2005) These mechanisms can generate a large number of training images with different morphologies, categories proportions and spatial structures. As a consequence, selection of an appropriate training image, that spans the uncertainty of a specific geological setting, is one of the main issues a modeling professional must face when using multiple-point simulation. This is a first order importance selection, as training image selection is analogous to variographic modeling in traditional geostatistical simulation (Boisvert, Pyrcz and Deutsch, 2007) and can be considered even more important, as it controls higher order, as well as second order spatial relations. As important as training image selection is to verify if the resulting simulated models keep the high order spatial structures observed in the training image.

Therefore, the main motivation of the present research is to develop tools that provide objective criteria for training image selection and simulated model validation. To achieve this, it is necessary to generate a metric that allows ranking training images according to their compatibility with the conditioning data and a second metric that permits computing an absolute compatibility between a single image and the conditioning data. Additionally, a tool is generated to determine if the spatial structure of the training image is correctly reproduced by the stochastic models.

1.2. Research objectives

1.2.1. General objective

There are two different main goals for this research. The first one is to generate geostatistical tools that provide measurable criteria for training image selection when more than one image is available, according to the spatial characteristics compatibility between the training images and the available conditioning data.

The second goal is to develop a multiple-point simulation results validation tool that allows validating models beyond the classical variogram and conditioning data reproduction verifications. It is aimed that the validation tool achieves its goal by verifying if the high-order spatial structure of the training image is preserved by the results.

1.2.2. Specific objectives

Spatial statistical tools are used to do the training image selection. This is done comparing the spatial relations observed in the drill-hole conditioning data with those observed in the evaluated training images. In order to determine the similarity between the two it is necessary to generate a metric to rank the training images according to their relative compatibility to the data set.

Additionally a tool to assess the absolute quality of training images regarding a data set is developed. This tool is required because training image ranking allows evaluating the conditional quality of images to the analyzed image set, but does not indicate whether the image is appropriate given the spatial relationships observed in conditioning data.

Both tools will be tested on case studies built with real and synthetic data in order to validate and evaluate the practical applicability of the developed algorithms. Different conditioning event replicates search strategies will also be tested to assess computing time improvements.

The training image selection algorithm is modified in order to extend the use of the developed high order conditioning event search algorithms to results validation purposes of multiple-point simulations. This new tool will be validated on a real mine-scale case study.

It is aimed to deliver all the developed tools in executable files with the necessary documentation to ease their application by a user and ease its application to real cases, so as to encourage the use of multiple-point simulation techniques

1.3. Scope

An algorithm and metrics are developed to assess the absolute and relative compatibility of a training image to a data set, regarding consistency conditions in terms of the spatial structure of the analyzed variable. The algorithm is also modified in order to extend its use to simulated model validations regarding the quality of high order spatial structure reproduction of the training image by the models.

However, there are many issues related to training image selection and model validation that are not addressed in this work. An important non-addressed issue regards how to obtain or construct the training image itself. Training images may come from unconditional object-based simulations (such as the TiGenerator tool implemented in SGeMS (Maharaja 2008, Boucher et al. 2010), wireframing of 2-D or 3-D interpretations of geologists about the geological phenomenon or characterization of outcrop data (Ortiz 2008).

Training image scale and resolution issues are also not addressed. The relationship between points depends on their values and distances, so generally training images have the same node spacing than simulated models. This problem can be solved using a multiple grid approach for integrating long and short range structures, but it is not addressed in this work. In this case same node spacing in training images and data grids is assumed.

Another non-addressed issue related to training images is stationarity. To perform most types of geostatistical simulation, the data set which provides spatial variability information (the training image in the case of multiple-point simulation) is required to have the property of stationarity (Mirowski et al. 2009). This means an assumption of pattern statistics translation-invariance across the image. Stationarity related issues are not addressed in this work, so the effect of trends in data and/or training images is not analyzed.

2. REVIEW OF RELEVANT LITERATURE

In this chapter an attempt to study the current state of multiple-point simulation is shown. Specifically the state of the art related to geological modeling in the mining industry and the current availability of training image selection tools. It is also aimed to summarize a theoretical support about the basic concepts related to multiple-point geostatistics in order to ease the understanding of the work.

2.1. Geostatistical simulation

Geostatistical simulation is used to construct a fictional regionalized variable, in such a way that it reproduces the inferred variability and spatial structure of a real variable (Emery 2008). To achieve this, the real regionalized variable is interpreted as a realization of a random function. The inference is performed to characterize this random function in order to allow the reproduction by the simulations of some parameters. This is generally done assuming some simplifications. Gaussian models for example reproduce the inferred distribution and 2-point spatial structure, generally imposed by a variogram. An advantage of this kind of inference is that it can be performed solely based on scattered information like for example drill-hole data. Meanwhile, multiple-point simulation requires adjusting a pattern frequency in order to reproduce higher-order relations. However, scattered data are generally scarce and do not provide a high enough amount of pattern replications to perform a representative multiple-point statistics inference. Therefore an exhaustive source of patterns needs to be provided, which corresponds generally to the training image (see section 2.5.1.2).

Once conventional or multiple-point statistical inference is performed, depending on the applied simulation method, geostatistical simulations are obtained as spatial realizations of the characterized random function. This allows building scenarios that reproduce real spatial variability, opposed to estimation methods, which smooth extreme values and hide useful information to perform certain analyzes.

The simple realization of a random function is known as non-conditional geostatistical simulation. If hard conditioning data are available in certain scattered locations, it is desirable that the simulated models reproduce those values. In these cases the simulation technique is known as conditional geostatistical simulation, which does not only reproduce the spatial structure of the variable, as it also reproduces some known values located in space. The simulated variables can be of different types, such as continuous variables like grades or concentrations of some elements; categorical variables as for example lithologies or alterations codes; or even whole objects with a given spatial morphology.

2.2. Geological modeling in the mining industry

A geological model corresponds to a mathematical or conceptual representation of a phenomenon. These models are generally constructed, using drill-hole information as input data, in order to predict the results of the phenomenon at uninformed locations. It is generally intended to interpret and predict the size, shape and location of geological units. The information that defines these units comes usually from categorical variables such as for example, alteration, mineralogy or lithology among others.

In the mining industry, the construction of these models is generally based on the knowledge of an expert who interprets the location and boundaries of geological units based on his personal understanding of geology and the information obtained from drill-hole samples. This interpretation procedure is generally performed by plan views and sections, interpolating subsequently the results, generating the 3-D spatial interpretation. Therefore and mainly due to the applied interpolation, the obtained results are geological units with smooth geometries (see Figure 1), models that are far from being similar to reality. The main disadvantage related to these deterministic models, is the impossibility of any kind of uncertainty quantification. There are also some conventional 2-point geostatistical techniques available that allow constructing probabilistic models of categorical variables. Some of these techniques are explained in the next section.

Figure 1: Deterministic geological model

2.3. Conventional categorical simulation techniques

The theoretical basis of conventional categorical simulation techniques is explained in this section.

2.3.1. Sequential indicator simulation

Sequential indicator simulation (Alabert 1987) is a widely used method to generate categorical models. The method encodes the samples (see Equation 1) as a vector of indicators and considers that categories cannot coexist at a same location.

$$I(S(x),S_k)=\begin{Bmatrix} 1\,if\;\; S(x)=S_k \\ 0 \qquad otherwise \end{Bmatrix} k=1,...,K$$

Equation 1: Indicator codification

In Equation 1, K is the number of categories, S(x) is the value of the category at location x and S_k is the value of category k.

Subsequently, the nodes of the simulation grid are visited using a random path. At each visited node, the algorithm searches conditioning data within a user-defined neighborhood and obtains the occurrence probability of categories through indicator kriging. This retrieves the conditional probability distribution function at the simulated location. Finally a category is drawn by Monte Carlo simulation.

The simulated value is saved and used as conditioning data for the simulation of subsequent visited nodes, repeating at each one the procedure explained above.

This algorithm also allows incorporating secondary data or "soft information", performing in this case a co-kriging instead of a simple kriging to build the conditional probability distribution prior to simulation.

2.3.2. Truncated gaussian simulation

This method (Matheron et al. 1987) requires defining a categorical variable through the truncation of a continuous random function Y(x) with multi-gaussian spatial distribution:

$$I(x,y)=\begin{Bmatrix} 1\,if\;\; Y(x)\le y \\ 2\,if\;\; Y(x)> y \end{Bmatrix}$$

Equation 2: Truncation of the gaussian random function

Therefore, first a realization of the random function is obtained, which is truncated, defining the categories to simulate by a threshold. The threshold "y" defines the proportion of each category.

The variogram of the gaussian random function controls the contact regularity between geological units. For example, a parabolic variogram would generate smooth contacts while less regular variograms (exponential, for example) would produce erratic contacts (see Figure 2). There is also a relationship between the variogram function of the gaussian random function and the indicator variable (Emery 2010).

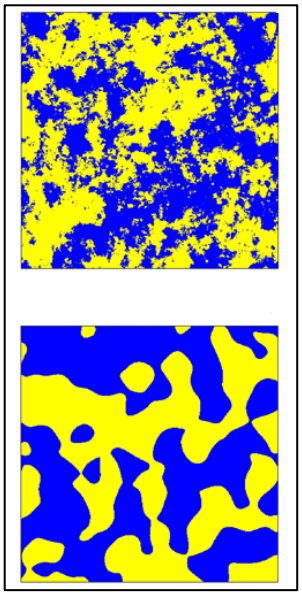

Figure 2: Effect of an exponential (upper) and gaussian (lower) variogram model of the gaussian random function on categories contacts (Emery 2010)

This method can be extended to cases with more than 2 categories, adding more thresholds and defining the number of truncations according to these thresholds (see Equation 3).

$$I(x, y_1, y_2) = \begin{Bmatrix} 1 \, if \;\; Y(x) < y_1 \\ 2 \, if \;\; Y(x) \geq y_1 \; and \; Y(x) < y_2 \\ 3 \, if \;\; Y(x) \geq y_2 \end{Bmatrix}$$

Equation 3: Truncated gaussian with multiple thresholds

The main limitation of this method is that the resulting models will reproduce only contacts between categories separated by a threshold. This means that to move from

category n to category n+2, it is necessary to go through category n+1 (see Figure 3). The plurigaussian simulation technique (Galli et al. 1994, Armstrong et al. 2003) is developed in order to solve this problem.

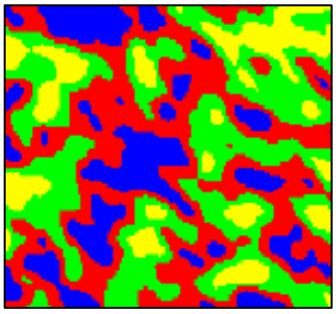

Figure 3: Truncated gaussian simulation model with 4 categories

2.3.3. Plurigaussian simulation

In order to provide more flexibility to the definition of geological units, the idea to truncate more than a single random function through the use of multiple thresholds is introduced (Galli et al. 1994, Armstrong et al. 2003).

$$I(x, y_1, y_2) = \begin{Bmatrix} 1 \, if \;\; Y_1(x_1) < y_1 \\ 2 \, if \;\; Y_1(x) \geq y_1 \;\; and \; Y_2(x) < y_2 \\ 3 \, if \;\; Y_1(x) \geq y_1 \; and \; Y_2(x) \geq y_2 \end{Bmatrix}$$

Equation 4: Truncation of multiple random functions through the use of multiple thresholds

More flexibility is obtained due to the fact that the user has several parameters to achieve the desired modeling, for examples the number and value of truncation thresholds, the amount of gaussian random functions used, the simple and cross variograms of these functions, among others.

As in the case of the truncated gaussian simulation, the variogram of the gaussian variables defines the regularity of the contacts between categories. The truncation model (or flag, see Figure 4) defines the types of contacts and the thresholds determine the categories proportions in space. Even if this simulation technique allows the use of an indefinite number of gaussian random functions, in practice usually 2 are used as this number defines the dimensions of the truncation flag.

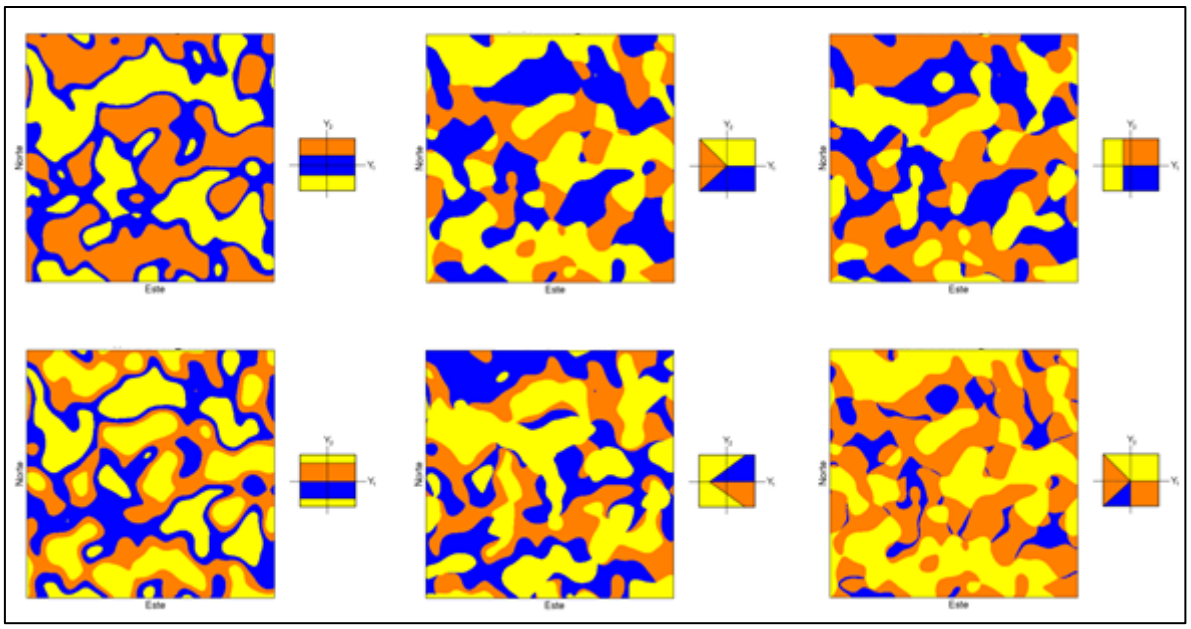

Figure 4: Truncation flag and its influence on contacts between geological units in space (Emery 2010)

2.4. Limitations of conventional simulation techniques

Although conventional Gaussian simulation methods, which use 2-point statistics, such as for example variograms to perform the inference of spatial continuity, are relatively simple and show generally good results, there are certain limitations. Among these:

- Curvilinear or complex structures require high order statistics rather than the traditional relations between pairs of points as these geometries are poorly represented by Gaussian simulations (Strebelle 2002).
- Variograms do not inform about contact geometries between different categories (Carle and Fogg 1996).
- The reliance of variogram-based geostatistics on the maximum entropy multi-gaussian distribution to model all statistics beyond the 2-point statistics result in maximum disconnectivity of extremes and the reproduction of only linear spatial features (Boisvert, Pyrcz and Deutsch 2007).
- Variograms do not capture complex relationships between multiple-points, making the description of complex geometrical properties impossible.

These limitations led to the introduction of pattern based simulation techniques, which were originally introduced by Guardiano and Srivastava (1993). The proposed single normal equation method simulates nodes sequentially, conditioning the simulation to nodes contained in a given neighborhood.

Contemporaneously Deutsch proposed an annealing technique, i.e. a technique that perturbs simulated node values to honor certain statistics, in this particular case, pattern statistics (Deutsch 1992). Since then, numerous pattern based simulation techniques and related optimizations have been developed, characterizing appropriately spatial structures.

2.5. Multiple-point geostatistical simulation

2.5.1. Basic concepts

2.5.1.1. Pattern

A pattern is a set of pixels or nodes, each with defined values of one or more variables, spatially arranged in such a way that allows the distinction from other patterns with a different spatial arrangement and/or different node values.

2.5.1.2. Training image

Usually the inference of complex spatial statistics from scattered conditioning data (drill-hole data in the mining case) is not sufficiently representative due to a lack of occurrences of conditioning events in the available data. To solve this problem the concept of training image is introduced. The training image is usually a conceptual exhaustive representation of the geological phenomenon that originates the spatial structure, and is generally prepared using the expert knowledge of a specialist. The main difficulties that arise when inferring statistics from a training image, in addition to obtaining or constructing the image or image set itself are:

1- The representativeness of the training image regarding the geological set.
2- Stationarity needs to be assumed.

This work aims at addressing point 1.The concept of training image and image selection will be explained in detail in section 2.6.

2.5.1.3. Search template

The search template is defined as a set of nodes with relative positions $h_1 \ldots . h_n$, where h_i corresponds to a vector. For a reference node *u*, the search template is the set of nodes whose positions are described by the following equation:

$$\tau_n(u) = \{u + h_1, ..., u + h_n\}$$

Equation 5: Search template

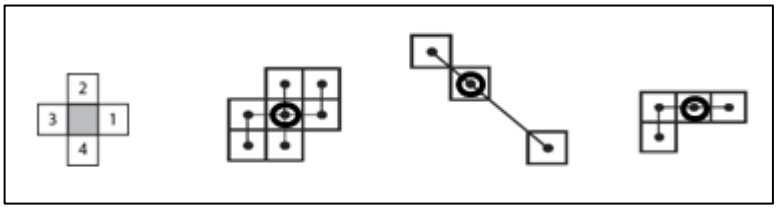

Figure 5: Search templates (Mariethoz et al. 2010)

In Figure 5 different search templates can be observed. The recommended size of a search template is conditioned by various factors. This size is also fundamental to infer representative patterns from the training image. A small template prevents the inference of large-scale structures, while large templates demand large computational resources like RAM memory and CPU time (Strebelle 2002). In addition, it is necessary to consider that the size of the search template is limited by the size of the training image and that large scale patterns tend to occur less, hindering a robust inference of them.

Although the size, shape and orientation of the search template are critical parameters for multiple-point simulation, as they allow the inference of representative patterns of certain geometries and sizes or reproducing certain geological structures, there has been a lack of efforts to define or select them. Eskandaridalvand (2008) proposes a method to select optimal search templates. The method defines initially a regular search template of dimensions N_x, N_y, N_z. The template scans the training image, defining at each visited node a smaller training image of the size of the template. Finally the array of smaller training images is analyzed by calculating the covariances between the central node and the rest of the nodes, generating a ranking of nodes ordered by covariances. The optimal template is defined by the N user-defined nodes with higher correlation to the central node.

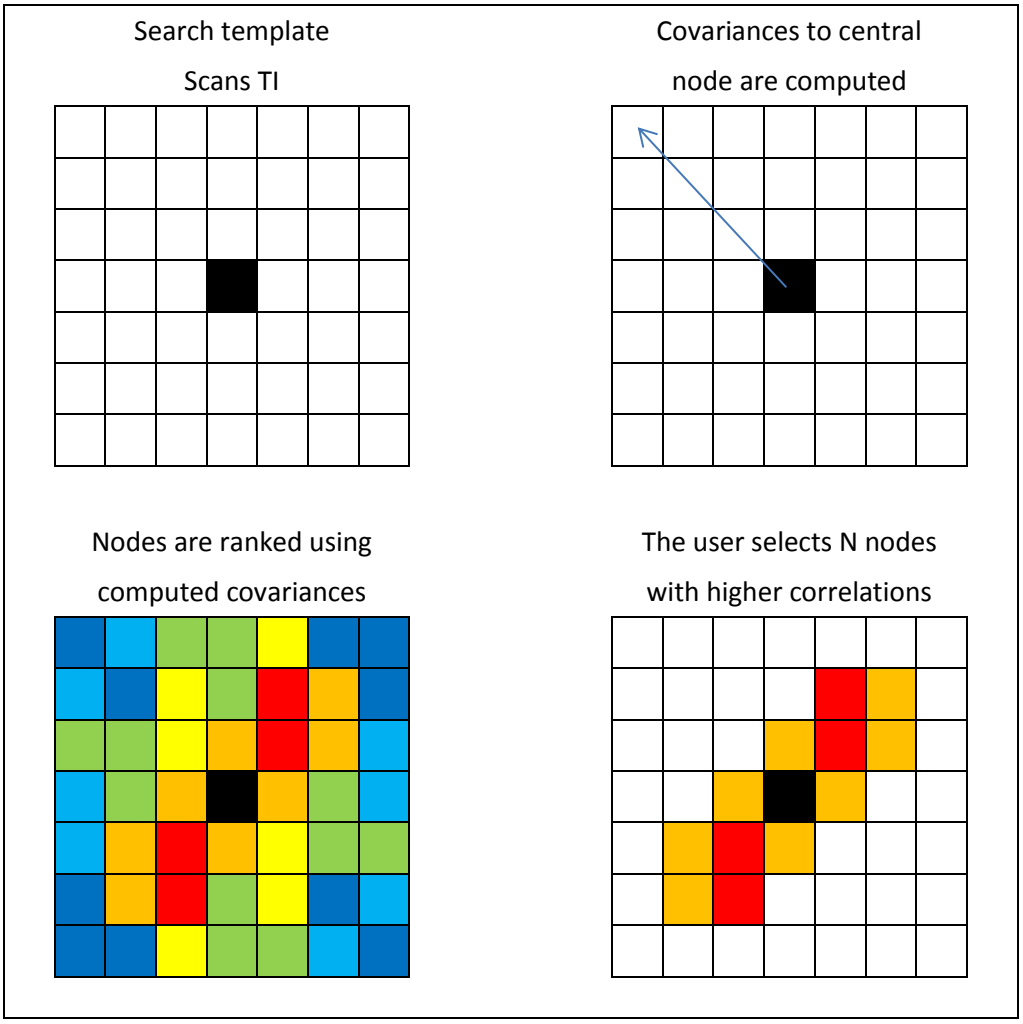

Figure 6: Optimal search template definition (Eskandaridalvand 2008)

2.5.1.4. Storing multiple-point statistics

The main limitation of the method proposed by Guardiano and Srivastava at the beginning of the 1990s was that it required to re-scan the whole training image to construct the conditional probability distribution function prior to the simulation of each node. This made the method too slow to be applied on a real size case study.

Strebelle (2000) implemented a dynamic data structure called search tree to optimize the construction of the conditional probability distribution function for a given conditioning event. The search tree is built at the beginning of the simulation and requires only a single training image analysis. The depth of the search tree equals the number of nodes of the template. Each cell of the tree is divided into sub-cells which equal the amount of different facies contained in the image. For a search template and training image such as those shown in Figure 7, a search tree as the one shown in Figure 8 is constructed.

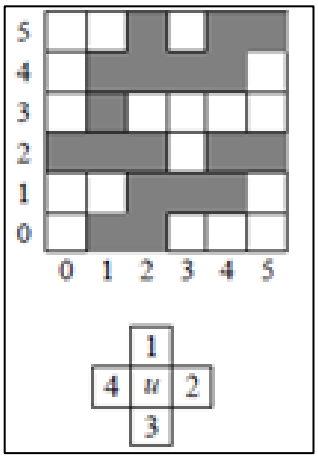

Figure 7: Training image and search template (Straubhaar et al. 2008)

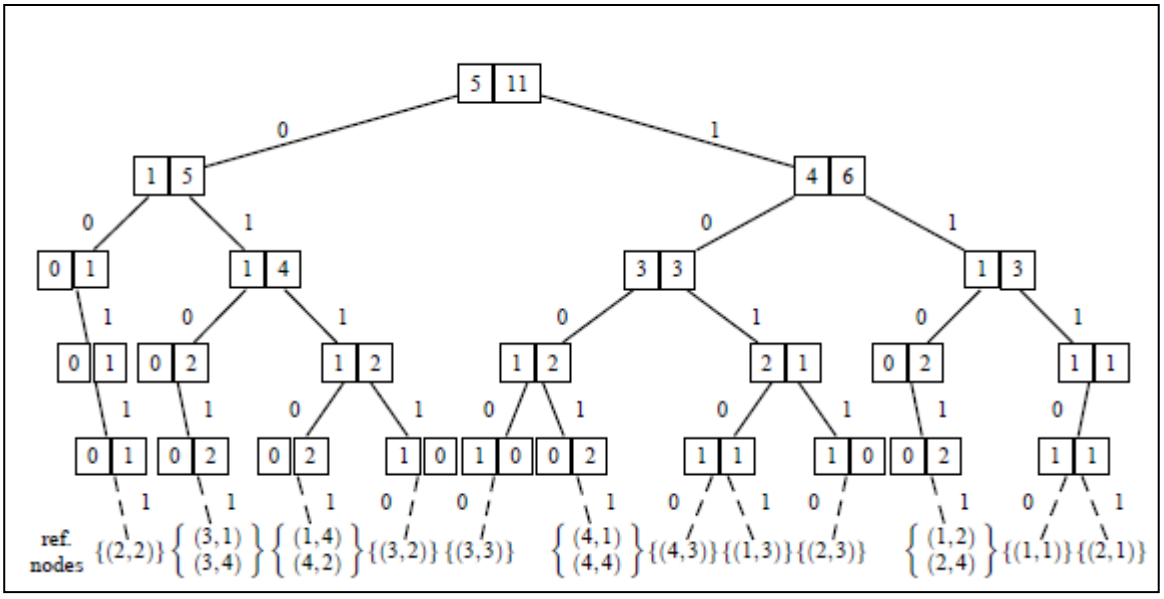

Figure 8: Search tree (Straubhaar et al. 2008)

The main limitation of search trees is the prohibitive RAM memory demand if the amount of nodes of the search template or the number of categories of the training image is large. An alternative lists method is proposed (Straubhaar et al. 2008, Parra and Ortiz 2009) to store the inferred pattern statistics. The lists are equivalent to the last level of the search tree which means that the RAM memory savings are considerable and that the stored statistics are the same than the ones stored in a tree. However, the lack of structure of lists involves high CPU time demand to build the necessary conditional probability functions.

The Direct Sampling simulation algorithm (Mariethoz et al. 2010) offers an alternative as it does not require storing inferred multiple-point statistics (see section 2.5.5).

2.5.1.5. Conditional probability distribution function

Obtaining the probability distribution, given a conditioning data configuration is highly important for the outcomes of a multiple-point simulation, as the knowledge of the conditional probability distribution function is required in order to perform a Monte Carlo simulation.

Generally, multiple-point simulation algorithms use the search template to obtain the distribution. To do this, the center of the template is located on the node to be simulated and conditioning data nodes are assigned to the template considering proximity. Conditioning nodes can be hard data (drill-hole data) or previously simulated nodes. The amount of assigned nodes constitutes the order of the conditioning event. Once defined, all occurrences of the conditioning data event are searched in the search tree (or list), considering eventually location or pattern matching tolerances. This procedure permits obtaining the facies proportions of the central node, given a certain conditioning data event.

An example of a categorical variable that takes values “black” or “yellow” is shown next. Figure 9 shows a training image and the search template used to build the search tree.

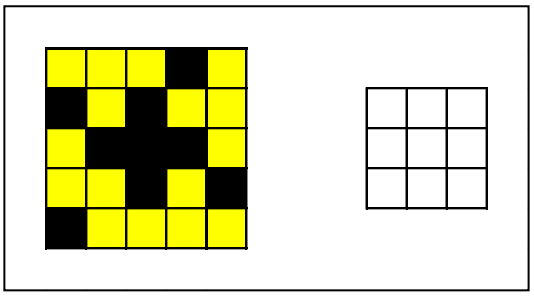

Figure 9: Training image (left) and search template (right)

The central node of the template is located in all the nodes of the training image so that the template is completely contained in the image.

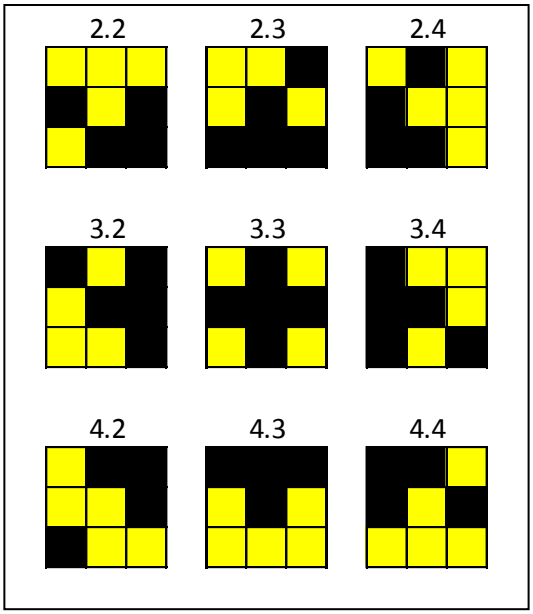

Figure 10: Inferred patterns. The number shows the location of the image from which they were inferred

In Figure 11, it is possible to observe the simulation grid with some informed nodes that will be used as conditioning data for the simulation and the obtained conditioning data event considering U the next node to be simulated and the search template.

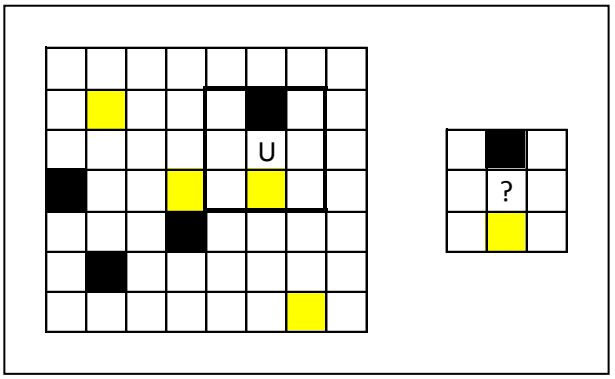

Figure 11: Simulation grid with conditioning information (left) and conditioning event (right)

It can be seen in Figure 10 that the conditioning event occurs in patterns inferred at locations 4.2, 4.3 and 4.4. In cases 4.2 and 4.4 the central node is "yellow", while in case 4.3 it is "black". Therefore, given the training image, search template and conditioning information, the node U has a conditional probability of being "yellow" equal to two-thirds and of one-third of being black. This conditional probability can be denoted as shown in Equation 6.

$$P(Z(u) = k | dat_j(u))$$

Equation 6: Conditional probability

Where Z(u) is the categorical regionalized variable, k is a possible value that can be taken by the variable (in this case yellow or black) and $dat_j(u)$ is the conditioning data event given a search template of size J (8 nodes in this case).

$$\sum_{k=1}^{K} \left[P(Z(u) = k | dat_j(u)) \right] = 1$$

Equation 7: Conditional probability distribution function

In Equation 7, K is the number of values that can be taken by the categorical variable, 2 in this case. If the multiple-point simulation algorithm generates stochastic models that use the conditional distribution function defined in Equation 7, the models would reproduce the categories proportions of the training image.

It is important to remark that all multiple-point simulation algorithms that store inferred patterns in a search tree or list retrieve the conditional probability distribution function from that data structure using the local conditioning data event. However, the Direct Sampling simulation algorithm (Mariethoz et al. 2010, see section 2.5.5) does not require to compute the function explicitly to reproduce its characteristics.

2.5.2. Multiple-point simulation algorithms

2.5.2.1. Single Normal Equation Simulation (SNESIM)

Proposed by Strebelle (2000), the SNESIM algorithm is a categorical multiple-point simulation algorithm. The training image is scanned only once at the beginning of the simulation using a user defined search template. The inferred multiple-point statistics are stored in a search tree.

The simulation process which simulates one node at a time, starts assigning the conditioning information to the closest simulation nodes. The search template is then centered on each uninformed node of the grid using a random simulation path. The conditioning nodes that are contained in the template define the respective conditioning data event, which is used to build the conditional probability distribution function from which the simulated value is drawn. If a minimum number of replications of the event is not found in the tree to consider the inference representative regarding that particular event, 2 different alternatives are presented to solve the problem. The first one is to re-simulate the node at the end of the process (previously simulated nodes are considered as conditioning information, which means that conditioning events change if the path or simulation order changes), while the second one consists in dropping the farthest conditioning data and re-searching the new conditioning event in the tree. The main problem regarding far data dropping is that long range structures will not be properly reproduced. A summary of the SNESIM algorithm process is shown in Figure 12.

Regarding training image selection, it can be considered that the number of re-simulated and dropped nodes constitutes a consistency measure between the training image and the conditioning data. However, there are no published studies in this regard and the number of dropped are also influenced by several other factors as for example the size of the training image with respect to the complexity and size of the considered structures.

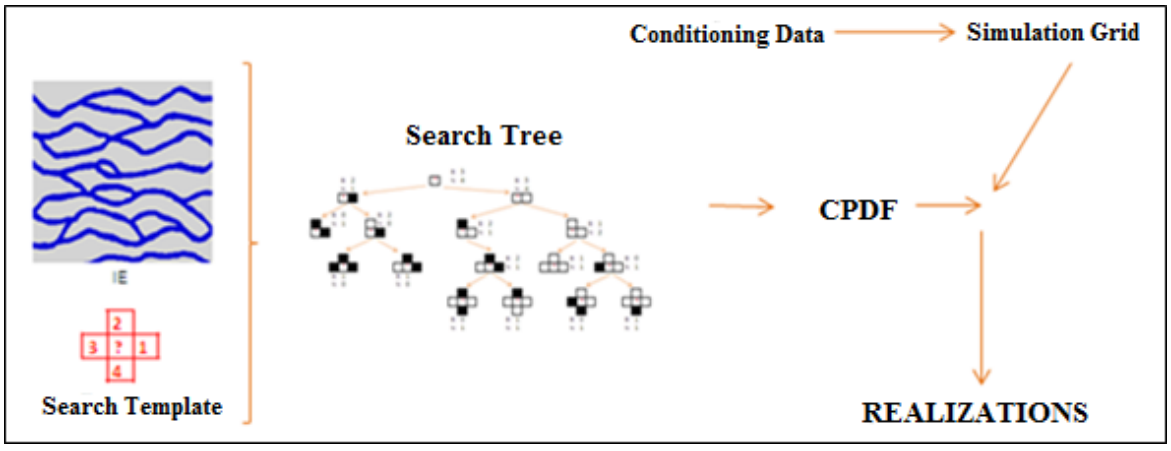

Figure 12: SNESIM simulation process (Hurtado 2009)

2.5.2.2. FILTERSIM algorithm

FILTERSIM is a pattern simulation algorithm that allows the simulation of categorical and continuous variables, proposed by Zhang et al. (2006). Again, the training image is scanned once at the beginning of the process using a search template of fixed size and regular geometry. The algorithm groups training image patterns into sets of patterns characterized by an average pattern called prototype, using filters. The prototype is obtained by averaging all patterns grouped in the same category.

The algorithm considers 3 different types of conditioning data: original hard data (k=1); previously simulated nodes of the same grid level (close to the central node) (k=2); and previously simulated nodes of coarser grid levels (k=3). The spatial configuration of conditioning events is obtained using the same search template as the one used to scan the training image. The distance between the weighted (where w(k=1) > w(k=2) > w(k=3)) conditioning information and the prototypes is computed. Finally, the closest prototype is simulated and pasted completely into the simulation grid. Simulated nodes corresponding to the same grid level are not re-simulated, while simulated nodes of coarser grid levels are used as conditioning information but are eventually re-simulated.

Regarding the filtering methods used by this algorithm, it is possible to imagine that as in this case sets of filters are applied to compare or "measure" distances between conditioning and training events, it might be possible to use filters to compare whole training images to conditioning data sets or pattern simulation outcomes.

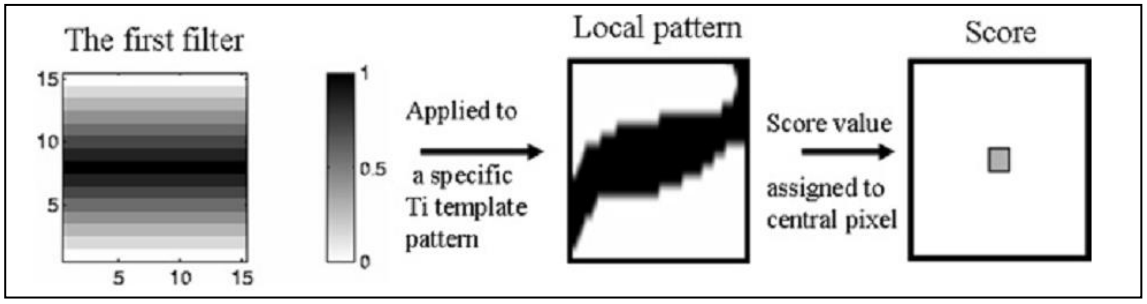

Figure 13: FILTERSIM (Zhang et al. 2006)

2.5.3. Growthsim

This algorithm, presented by Eskandaridalvand (2008) and intended for the modeling of oil deposits, is not exclusively a pattern simulation algorithm. It corresponds to a complete framework that also includes a procedure for training image selection and optimal search template definition given a certain set of conditioning information (explained in sections 2.6 and 2.5.1.3 respectively). It additionally considers options for managing non-stationary training images and the incorporation of other information, such as dynamic production information.

Growthsim proceeds selecting a compatible training image using pattern based statistical criteria and defining an optimal search template using second order statistics (based on relations between pairs of points). The algorithm simulates full patterns in the surroundings where conditioning information is available, so the simulated area or volume "grows" from the informed locations.

Growthsim gives the user the option to manage possible training image non-stationarity through the use of different, locally stationary training images or through the use of a Servosystem factor (described in section 2.6). Dynamic information is integrated to simulation incorporating history matched geological models, which satisfy the production history of the deposit, to training image information. This information is used to perturb the inferred statistics.

2.5.4. Texture based simulation algorithm CUTSIM

Proposed by Parra and Ortiz (2009), as SNESIM, this algorithm simulates a node at a time and infers the spatial structure of the variable from a training image. The algorithm initializes the simulation imposing random values to the search template nodes. The following nodes are visited in order, using as conditioning information the previously simulated values. Initially, the conditioning information comes from the zone initialized with random noise. Figure 14 is useful to clarify the application of the algorithm.

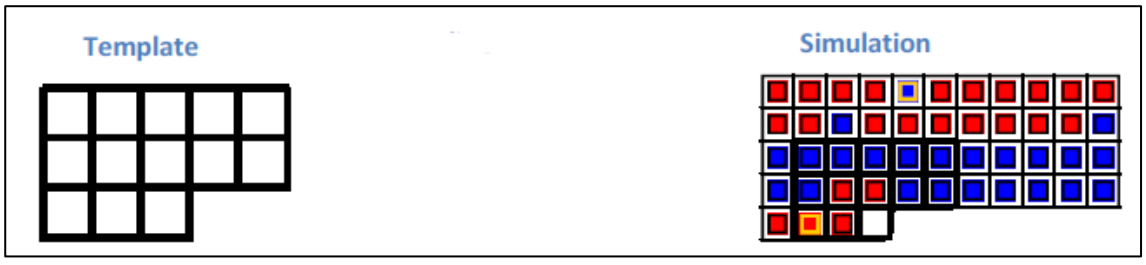

Figure 14: Non-conditional CUTSIM simulation algorithm (Parra and Ortiz 2009)

However, the described algorithm does not honor conditioning information, such as for example drill-hole data. In order to match the conditioning information, a search template that checks for the presence of conditioning information in the simulation direction is added. This new information is used to perturb the conditional probability distribution function obtained from the training image and the original search template. The complete algorithm is described in Figure 15.

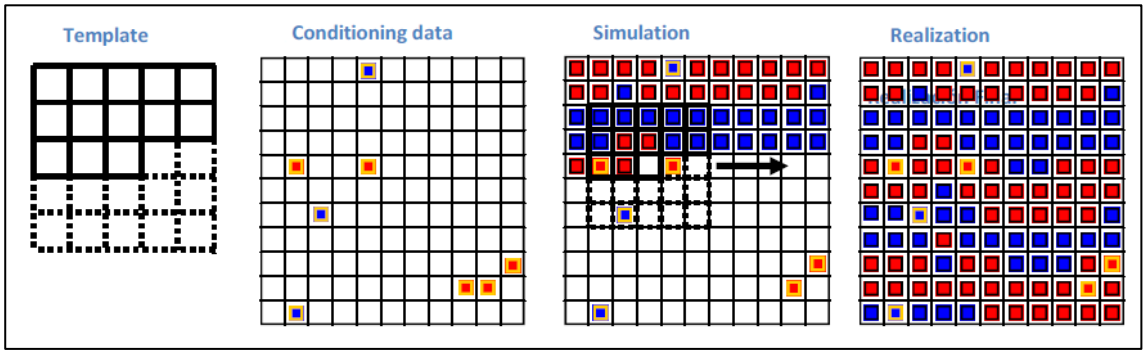

Figure 15: Conditional CUTSIM simulation algorithm (Parra and Ortiz 2009)

The CUTSIM algorithm allows quickly obtaining simulated models, even when large search templates are considered.

2.5.5. Direct Sampling simulation algorithm

This algorithm, proposed by Mariethoz et al. (2010) samples directly the training image for the occurrence of a given conditioning data event. This avoids the storing of inferred multiple-point statistics. The Direct Sampling simulation algorithm relaxes the need to compute explicitly the conditional probabilities and allows extending multiple-point simulation to continuous and multivariate applications. For each node to be simulated this simulation algorithm finds a user defined number of closest conditioning neighboring nodes (order of the conditioning event). The distances (different distance computing metrics can be applied) from the node to these closest neighbors are computed, defining the conditioning data event. A search window is defined using the mentioned distances to scan the training image using a random path. At each visited node of the training image, the closest event that matches a user-defined fraction of the conditioning event is searched and the distance to the analyzed node is computed. If the distance between conditioning and sampled events is smaller than a given threshold, the value of the visited node in the training image is assigned to the simulated node. When this simulation algorithm is used, the quality of pattern reproduction depends on the acceptance distance threshold and the matching fraction of the conditioning event by the sampled event. High matching fractions and low distance thresholds can be very CPU demanding, but one of the advantages of this algorithm is that it is easy to parallelize, as each CPU unit can be instructed to analyze a

certain region of the training image. Other advantages are massive RAM demand reduction and flexible geometries of the search template.

In this work, the Direct Sampling algorithm is applied to model lithologies at the Escondida Norte deposit in order to compare its performance to sequential indicator simulation. This case study is also used to validate the results validation tool.

2.6. Background on training image selection

There are many problems related to the selection of training images that represent correctly the structure and characteristics of a certain geological setting, as for example the existence of significant differences between the facies proportions of the training image and the conditioning data. This problem has been addressed through the implementation of corrections as the Servosystem Factor of the SNESIM algorithm (Strebelle 2000). However, there is no guarantee that the final realization will reproduce the statistics extracted from the training image if the correction to match the univariate proportions is significant (Ortiz 2008).

The Servosystem Factor corrects the conditional distribution function according to Equation 8:

$$P^*(Z(u)=k \mid dat_j(u)) = P(z(u)=k \mid dat_j(u)) + \frac{w}{1-w} * (P_k^t - P_k^c)$$
$$for\ w \in [0,1)$$

Equation 8: Servosystem Factor

Where P* corresponds to the corrected conditional probability distribution function, w is the value of the correction factor, P_k^c is the simulated proportion of a given category k and P_k^t is the target proportion of that category (usually the proportion of the category in the declustered conditioning data or a value chosen according to the criterion of the modeler). Therefore, if w=0, there will be no correction imposed by the Servosystem Factor, while the maximum correction is imposed if w=1.

There are also scaling issues related to training images. The image should be large enough to allow a representative inference of short and long range structures, especially in the case of the large ones, which may have a reduced amount of repetitions in a small training image.

However, the main problem related to training images is the lack of a tool to assess its consistency with a given conditioning data set, and its quality to model a certain deposit. Due to the fact that training image construction generally includes subjective judgment, different specialists could generate training images with different characteristics, which cannot be evaluated or selected based on objective criteria.

Ortiz and Deutsch (2004) and Boisvert et al. (2007) discuss two different methods for the evaluation and selection of training images. The first one, the distribution of runs, is applied in the context of one-dimensional multiple-point statistics. The concept is to count the number of consecutive occurrences of a certain category in a certain direction. This determines the length of the run. Runs are equivalent to high-order indicator covariances (Ortiz 2003). It should be noted that to obtain the cumulative distribution of runs, it is considered that for example a run of length 3 contains 2 runs of length 2 and 3 runs of length 1. The idea proposed by the authors consists in comparing the cumulative distribution of runs of the training image with the cumulative distribution of runs of conditioning data (both represented in relative frequency, as shown in Figure 17).

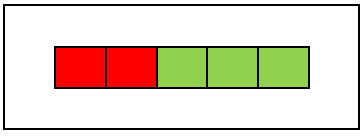

Figure 16: Run

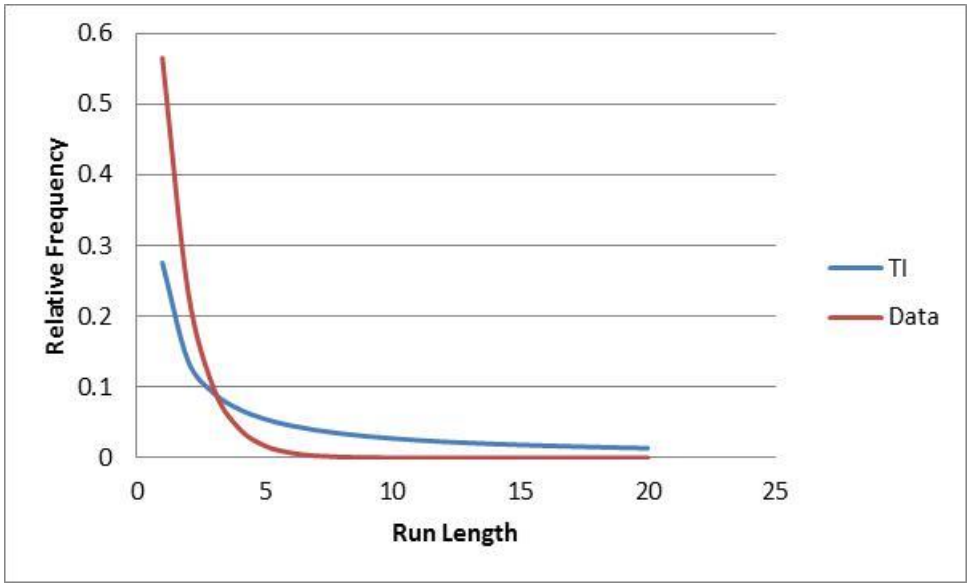

Figure 17: Comparison of cumulative distribution of runs

The second proposed method corresponds to a comparison of multiple-point statistics histograms of simple patterns, i.e. patterns that have a relatively simple spatial configuration. The training image and the conditioning data are scanned using a simple

search template and these statistics are compared. Because conditioning data usually correspond to drill-hole data, the vertical direction has generally higher information density than other directions. Therefore, the method only proposes the comparison of vertical one-dimensional pattern histograms, as shown in Figure 18 and Figure 19.

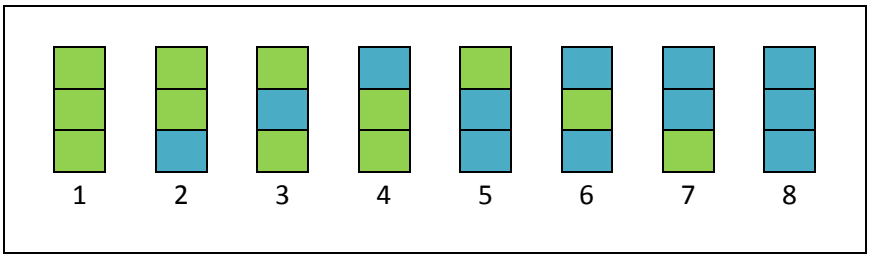

Figure 18: Vertical one-dimensional patterns

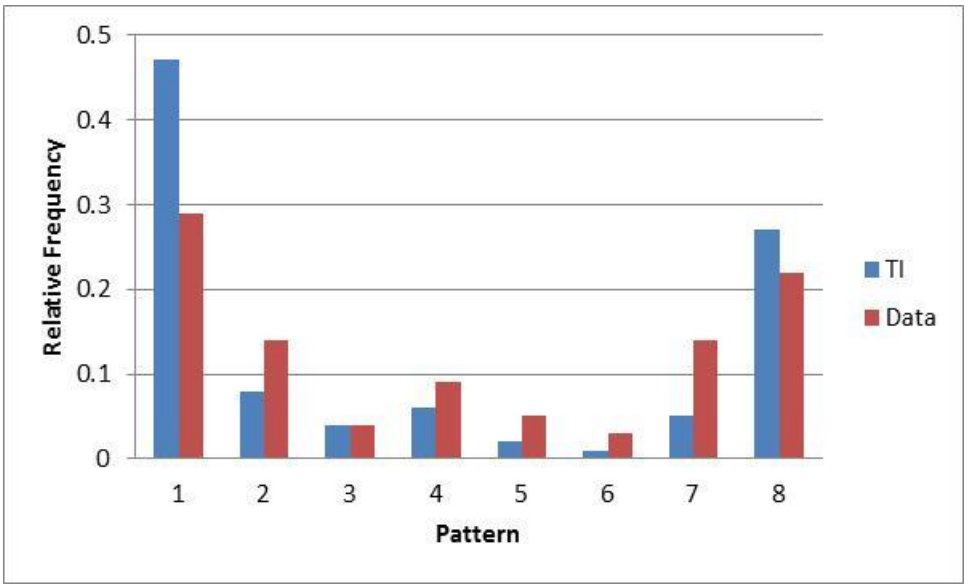

Figure 19: Patterns distribution

Eskandaridalvand (2008) proposes alternatively a spiral search method. The method loops over all conditioning nodes and for each of them over all training nodes. If the node in the training image has the same value than the conditioning node, the method loops over the closest conditioning nodes from the nearest to the farthest. The values are compared to the values of nodes contained in the training image that show the same spatial configuration to the central node. If both values have the same relationship, i.e. if both increase or decrease in relation to the last analyzed node, the same relationship is shown and the number of compatible nodes increases. If the relationship changes, the loop over the neighbors ends. The method allows obtaining a distribution of compatible nodes for each conditioning node and a unique distribution of maximum compatible nodes. Obviously the last one has the same dimension as the number of conditioning data. These distributions can

be used for deriving a measure of consistency between the training image and the conditioning information.

Mirowski, Tetzlaff et al. (2009) propose assessing the validity of training images in terms of stationarity. The authors consider that to perform most types of geostatistical simulation, the data set that provides spatial variability information (the training image in this case) is required to honor the property of stationarity. The proposed method assesses three types of stationarity: Statistical distribution stationarity (requirement that the statistical distribution of property pixel values is invariant under translation), orientation stationarity (requirement that the directional statistics do not change across the image) and scale stationarity (that the size of elements does not change across the image). In the case of considering that the training image does not meet the requirements of stationarity, several corrections are proposed that can allow meeting them. Although the proposed method (shown in Figure 20) does not perform a consistency analysis between the training image and the conditioning information, objective criteria are used to assess the quality of the image regarding the property of stationarity.

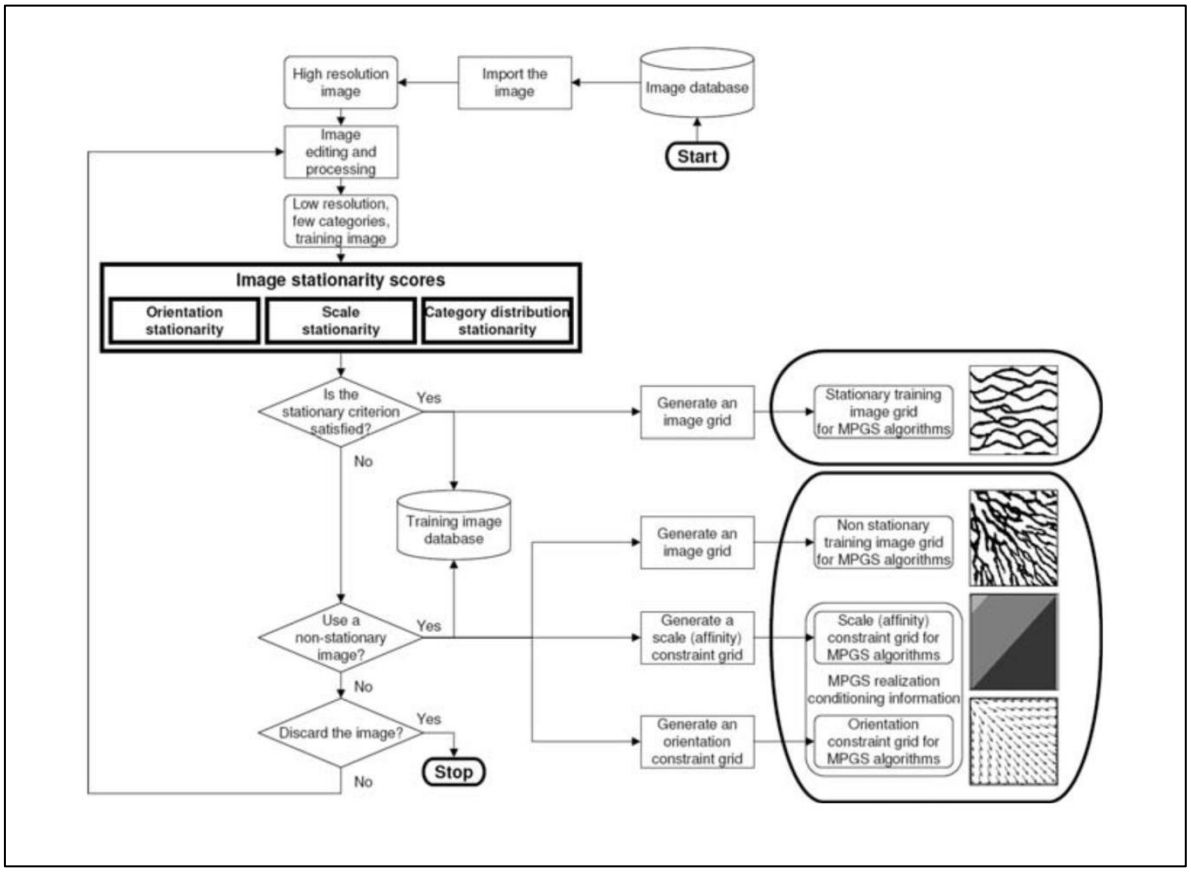

Figure 20: Training image validation method (Mirowski, Tetzlaff et al. 2009)

2.7. Background on multiple-point simulation results validation

Generally, once any type of model is completed it is required to check that some features are reproduced in order to validate the model. For example, for any model generated with hard conditioning data it should be checked that the values of the samples are locally reproduced. The results of a conventional variogram based simulation should also honor the variogram model and the distribution of the modeled variable observed in data. In the case of a multivariate model, it could also be checked if the correlations between variables observed in data and model are similar.

Al the mentioned conventional validations are easily performed through the use of well-known available techniques (qq-plots for distributions reproduction check, scatter-plots for correlation reproduction check, and comparison of experimental variograms of results against the imposed variogram model).

Even if these techniques are often also applied for validation of multiple-point simulation results, these methods are unable to verify if the patterns observed in the training image are reproduced by the model. Even if this can be considered as important as a variogram model reproduction check for conventional simulations, efforts to develop validation tools for multiple-point simulations results have lacked.

De Iaco and Maggio (2011) present a comparison between conventional and multiple-point simulation algorithms (sequential indicator simulation and SNESIM algorithm respectively), performing a quantitative validation and comparison of the results of both algorithms to check if the essential characteristics of the training image are reproduced. First physically measurable characteristics of both results, such as the proportion occupied by each geological unit (abundance measure), maximal directional extensions of bodies (connectivity measure) and ratio of surface to volume (measure of shape and ruggedness), are compared through the use of the faprop algorithm proposed by Soleng et al. (2006).

The reproduction of high order statistics of the training image is checked through the use of high order spatial cumulants (Dimitrakopoulos et al. 2010), based on case study specific orientated templates of orders 3 and 4 (see Figure 21). High order spatial cumulants can be considered a high order extension of the covariance function. A nice feature of this method is that it allows the user to check the reproduction of key patterns to somehow assure the reproduction of a desired anisotropy, curvature or more complex geometry contained in the training image. However, the checked amount of patterns is a low proportion of the total amount of patterns contained in the image.

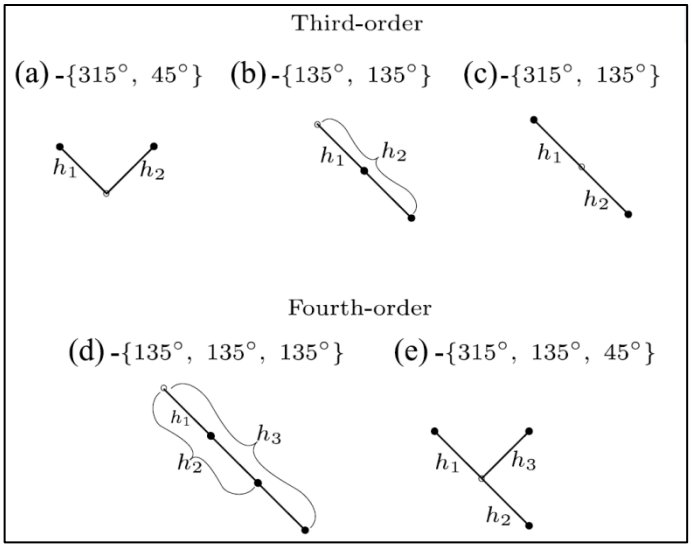

Figure 21: 3rd order and 4th order cumulant templates used for calculations (De Iaco and Maggio 2011)

3. METHODOLOGY

In this research, methodologies and algorithms are proposed and developed to achieve two different goals:

1. To select a training image when more than a single image is available. This goal is achieved measuring the compatibility of the training image to the available conditioning data in terms of low and high order spatial structure. Two different approaches are proposed:

    a. Relative compatibility measurement: Allows selecting a training image from an image set.

    b. Absolute compatibility measurement: Allows computing absolute spatial structure compatibility between a training image and the data.

2. To verify the quality and completeness of multiple-point simulation results. This goal is achieved by computing the proportion of training image patterns that are contained in the results and the proportion of results patterns that are contained in the training image.

Both algorithms are explained in sections 3.1 and 3.2 respectively. It is important to remark that due to the different characteristics (scattered or spatially exhaustive) of the data sets, different steps are required to be executed by the algorithms to achieve their goals.

Training image selection requires a training image set and conditioning data as inputs. Due to the scattered nature of conditioning data, a previous migration of data points to nodes of a regular grid is required (explained in section 3.1.1). The definition of the data event is performed considering that data points are located in scattered positions (3.1.2). To measure the relative compatibility between a training image and the conditioning data, once defined, occurrences of the conditioning event in the training images are searched. The scan of the training image can be performed using two different search strategies:

- Exhaustive scan (3.1.3), which is statistically sound but computationally demanding.
- Direct sampling (3.1.4), allows obtaining similar results at a less computational cost.

Finally, the direct sampling scan strategy is modified in order to measure absolute compatibility between a single image and the data set (3.1.5). The training image selection

algorithm is implemented in the TI-selector.exe program. Implementation details of the program are explained in section 3.1.6.

Simulation results validation requires a training image and the simulated model as input data, which correspond to two spatially exhaustive data sets. Therefore no data migration is required and definition of data events is modified (3.2.1). This algorithm only implements the direct sampling scan strategy (3.2.2). However, in this case the next two steps are performed:

- Results events are defined and occurrences are searched in the training image (measures quality of results)
- Training image events are defined and occurrences are searched in the results (measures completeness of the results).

The results validation algorithm is implemented in the results-validator.exe program. Implementation details of the program are explained in section 3.2.3. The implemented steps by algorithm are summarized in Table 1.

Algorithm	**Training image selection**		**Simulation results validation**
Input data	Training image set (exhaustive) Conditioning data (scattered)		Training image (exhaustive) Simulation result (exhaustive)
Data migration	Required		Not required
Definition of data event	Adapted to define data event from scattered data		Adapted to define data event from exhaustive data
Alternative implementations	**Relative**	**Absolute**	-
Search of data event occurrences	Exhaustive scan option and direct sampling scan option	Direct sampling scan option	Direct sampling scan option
Implementation details	Searches conditioning data event occurrences in training image	Searches conditioning data event occurrences in training image	Searches results event occurrences in training image (quality) and training image event occurrences in results (completeness)
Program	TI-selector.exe	TI-selector.exe	Results-validator.exe

Table 1: Summary of the developed algorithms

3.1. Training image selection

In this section, many abbreviations are used to explain the methodology. These abbreviations are summarized and explained below.

- $F_{i,\,n}$ = Facies value at node *n* of conditioning event *i*.
- $\mathbf{X_{i,\,n}}$ = Offsetting vector to node *n* from center of conditioning event *i*.
- **TI** = Set of ranked training images.
- *J* = Number of training images.
- $\mathbf{TI_j}$ = Training image *j*.
- *t* = Accepted match tolerance to accept a pattern replicate.
- $M_{i,j}$ = Number of occurrences of the conditioning data event *i* in the training image *j*.
- $P_{i,j}$ = Probability of finding event *i* in image *j* if event *i* is contained in the image set.
- C_j = Relative compatibility between conditioning data and training image *j*.
- L_j = Conditioning event match counter of training image *j* (Direct Sampling Approach).
- *f* = Checked fraction of the training image (Direct Sampling Approach).
- A_j = Absolute compatibility between conditioning data and training image *j*.

The purpose of the algorithm developed for training image selection is to generate a ranking of several training images, according to their compatibility with conditioning data. In essence, the algorithm is given a conditioning data set and a series of training images. The method works by defining conditioning events, which are patterns of spatially distributed values, and computing their probability of occurrence in the different training images. The training images that have the same distribution of data events as the data are deemed more coherent with the data. The result is a ranking of the training images according to their compatibility with the data. The overall algorithm can be divided in a number of steps whose implementation is described below in more detail. First, an algorithm that is statistically straightforward but computationally inefficient is presented. In a second step, an equivalent alternative is presented that yields similar results with a lesser computational burden.

3.1.1. Migration of conditioning data points

The first step is to migrate the scattered data to a regular grid. The grid structure permits to accelerate future spatial searches of data events, therefore reducing computing times. In this implementation, the user must enter the required parameters to define the regular grid: coordinate of the center of the node with minimum X, Y, Z coordinates, number of nodes and spacing of nodes in each direction. To avoid scale issues, the size of the nodes must be the same as in the training images. The algorithm loops over all conditioning data points, migrating each of them to the closest node of the grid. The value

of the conditioning point is migrated to a grid node if the coordinates of the point (X_p, Y_p, Z_p) meet the following conditions regarding the coordinates of the node (X_{nod}, Y_{nod}, Z_{nod}) and nodes spacing (D_x, D_y, D_z):

$$X_p \in \left[X_{nod} - \frac{D_x}{2}, X_{nod} + \frac{D_x}{2} \right] \wedge Y_p \in \left[Y_{nod} - \frac{D_y}{2}, Y_{nod} + \frac{D_y}{2} \right] \wedge Z_p \in \left[Z_{nod} - \frac{D_z}{2}, Z_{nod} + \frac{D_z}{2} \right]$$

Equation 9: Migration condition of conditioning point to grid node

In case two conditioning data points are inside the influence volume of one node, the first analyzed conditioning point is migrated. The number of migrated and not migrated data points (due to a data point already migrated to a node or if the node is located outside the grid) is reported in the output data file.

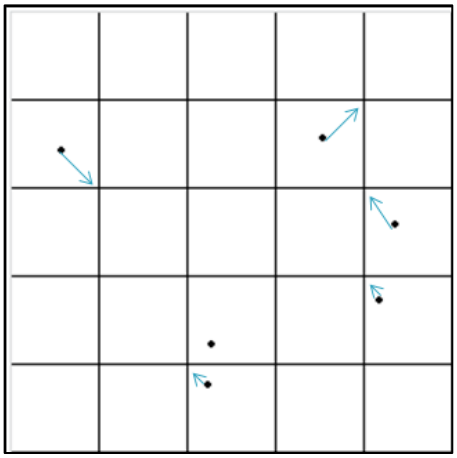

Figure 22: Migration of conditioning data points to regular grid

3.1.2. Definition of the conditioning data event in spatially scattered data sets

The second step of the method is to define the conditioning data events (CE_i) to be used for comparing spatial statistics of the data and of the training images. At every position of the migration grid, regardless of whether it contains a migrated data point, a spiral search is started. In order to avoid using very large, possibly non-stationary data events, the user needs to define a maximum search radius in each direction (R_x, R_y, R_z) and a minimum and maximum number of conditioning nodes that form the event (MIN_{con}, MAX_{con}). The spiral search goes from the nearest (starting at the center node) to the farthest point within the search neighborhood as shown in Figure 23.

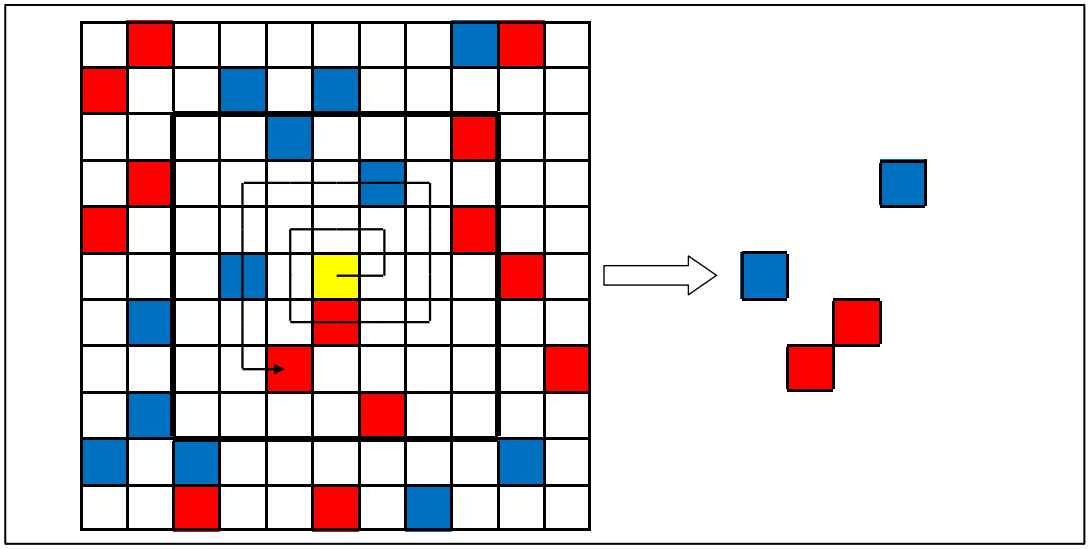

Figure 23: Spiral search (left) and defined conditioning data event (right)

The positions, defined as offsets to the central node ($OFX_{i,n}$, $OFY_{i,n}$, $OFZ_{i,n}$) and facies of the informed nodes ($F_{i,n}$) are saved in a list. If not enough data points are found in the prescribed radius (i.e. if the maximum search radius is reached without finding the required amount of conditioning node MIN_{con}), the data event is not considered for comparison and the next node of the conditioning grid is analyzed. However, if all nodes contained in the search neighborhood are analyzed and the minimum amount of conditioning nodes is reached, or the maximum amount of conditioning data nodes is reached before all the nodes in the neighborhood are analyzed, the spiral search is stopped (see Figure 23). The resulting list defines the data event considered (CE_i) consisting of N nodes as shown in Equation 10.

$$CE_i = OFX_{i,1}, OFY_{i,1}, OFZ_{i,1}, ..., OFX_{i,N}, OFY_{i,N}, OFZ_{i,N}$$

Equation 10: Conditioning event CE_i

The data events so defined will be used to compare training images with data. A difference between this approach and the use of multiple-point histograms as proposed by Ortiz and Deutsch (2004) and Boisvert et al. (2007) (see section 2.6) is that multiple-point histograms are limited to search conditioning data events of a fixed geometry through the use of a template, whereas this approach is able to characterize scattered values.

It should be noted that data events with a small number of nodes can be used to reflect low-order statistics, whereas data events with increasing nodes will represent higher order spatial distribution. For example, using a set of patterns of 1 node allows representing the marginal distribution, patterns of 2 nodes can represent 2-point dependence (similar to a variogram), etc. The appealing aspect of this approach is that it allows for comparisons

using multiple-order criteria. In the following, the terms “number of neighbors” and “data event order” will be used equivalently.

3.1.3. Relative compatibility with exhaustive scanning of the training image

The conditioning data events, defined as described in the previous section, are repeatedly extracted from the conditioning data and similar patterns in the different training images are inventoried. As the whole set of training images have to be analyzed, event occurrences are searched using a regular search path that runs over all the positions contained in the images as shown in Figure 24

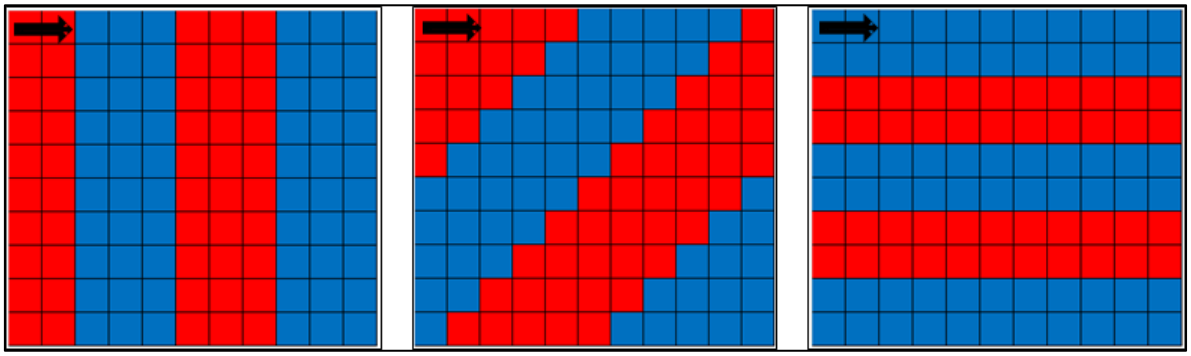

Figure 24: Regular training image scanning path

It is considered that an event found in the training image matches the conditioning data event if the positions and facies of all nodes of the conditioning event are matched by those in the training image. Since it can be difficult to find an exact match, a tolerance parameter t is introduced, which represents the proportion of nodes in the conditioning event whose facies are allowed to be miss matched in the pattern of the training image. This parameter can take values between 0 and 1. Considering $TI_j(\vec{X}_{i,n})$ the value of the facies in the training image j at the offsetting coordinate given by vector $\vec{X}_{i,n}$ and $F_{i,n}$ the facies value at the conditioning node, the indicator variable described in Equation 11 is used to quantify the match for each node in CE_i.

$$Ind_{i,n,j} = \begin{cases} 0 \;\; if \;\; TI_j(\vec{X}_{i,n}) = F_{i,n} \\ 1 \;\; if \;\; TI_j(\vec{X}_{i,n}) \neq F_{i,n} \end{cases}$$

Equation 11: Training image - conditioning node matching indicator for categorical variables

It is worth mentioning that the proposed algorithm can also be applied to continuous variables. However, for this kind of applications, another acceptance threshold ε would be needed. The threshold should be applied as shown in Equation 12.

$$Ind_{i,n,j} = \begin{Bmatrix} 0 \;\; if \;\; \left| TI_j(\vec{X}_{i,n}) - F_{i,n} \right| < \varepsilon \\ 1 \;\; otherwise \end{Bmatrix}$$

Equation 12: Training image - conditioning node matching indicator for continuous variables

ε should be expressed in terms of the distribution of the variable rather than being expressed as a simple value threshold. This algorithm includes this threshold but as a simple value threshold.

A pattern analyzed in an image *j* is considered a match if the condition expressed in Equation 13 is matched.

$$\frac{\sum_{n=1}^{N_i} Ind_{i,n,j}}{N_i} \leq t$$

Equation 13: Pattern match condition regarding the tolerance parameter t

Every time a match is found the value of the counter $M_{i,j}$ is increased, which corresponds to the number of matches for each data event in each training image. Once all occurrences are counted, the probability of finding this specific data event in each image ($P_{i,j}$) is computed (see Equation 14).

$$P(CE_i \in TI_j \left| CE_i \in TI \right.) = P_{i,j} = \frac{M_{i,j}}{\sum_{j=1}^{J} M_{i,j}}$$

Equation 14: Conditional probability of CE_i in TI_j relative to its occurrence in TI

$P_{i,j}$ can be interpreted as the proportion of occurrences of the conditioning event CE_i in the training image TI_j, relative to its occurrence in the set of analyzed training images ***TI***.

Finally, the relative compatibility between the training images and the conditioning data set is calculated by normalizing the sum of probabilities of finding each conditioning event in a given training image. This relative compatibility is calculated through Equation 15.

$$C_j = \frac{\sum_{i=1}^{I} P_{i,j}}{\sum_{i=1}^{I} \sum_{j=1}^{J} P_{i,j}}$$

Equation 15: Relative compatibility with exhaustive scanning

This relative compatibility value between the data set and each training image is the final result of the algorithm. As compatibilities are calculated through normalization, the sum of components of **C** equals 1.

It is important to note that this compatibility measurement is relative, as these probabilities are subject to finding the conditioning event in the analyzed image set. In other words, it tells us if one training image is more compatible with the data relatively to another training image. Therefore C_j (see Equation 15) would change if new training images are added to the set of images ***TI***. However, it does not give an absolute ranking of whether a training image is good or bad.

3.1.4. Relative compatibility with alternative direct sampling method

The previously proposed method is statistically sound as it checks the occurrence of conditioning patterns in every position of the ranked training images. However, because of the high amount of positions that need to be analyzed, CPU cost is very steep. As a cheaper alternative and to reduce computation times, an approach inspired from Direct Sampling (Mariethoz et al. 2010) is proposed.

Scattered conditioning data points migration and conditioning event search and definition remain unchanged. However, the occurrence count of conditioning events in the images and compatibility calculations are achieved differently. In this case, the search of occurrences of a conditioning pattern in the training images is done using a random path that is identical for all training images. Considering the tolerance parameter *t*, if one training image *j* matches the conditioning event at the checked position, a counter (L_j) for that image is incremented and the search is stopped. The algorithm then continues with the loop over the conditioning data grid. Note that in the algorithm described in section 3.1.3, the search would continue in order to count all matching occurrences. In contrast, here it stops at the first occurrence, allowing for a computational gain. In case more than one image matches the conditioning event simultaneously, the counter of every matching image is incremented.

Compared to the previous method consisting in scanning the entire training image, the user needs to define the fraction *f* of the training image that can be scanned. This parameter corresponds to the fraction of total positions (fraction of total nodes) of the training image that are analyzed in the search of a match, before stopping the loop. If a fraction *f* is reached without a match, it is considered that the searched pattern does not occur in the training image and the algorithm passes on to the next node in the conditioning data grid.

The relative compatibility between the conditioning data set and the training images is finally calculated using Equation 16.

$$C_j \underset{I\to\infty}{\approx} \frac{L_j}{\sum_{j=1}^{J} L_j}$$

Equation 16: Relative compatibility with direct sampling approach

If the amount of conditioning events and the size of the training images is high, this relative compatibility value should equal the value of relative compatibility presented in Equation 15.

3.1.5. Absolute compatibility measurement

In several cases it can be desirable to have an absolute measure of the compatibility between training images and data. For example, it may not be relevant to establish that one training image is slightly better than another, if both training images are grossly incompatible anyway. Therefore an absolute compatibility measurement is proposed, which is represented by the proportion of patterns contained in the data that are matched by the training image. This absolute compatibility is computed by analyzing each training image separately, as this absolute measure depends only of the analyzed training image and is independent from other images. It fluctuates between values of 0 and 1. If each conditioning event is found in the training image, which would represent the best possible case, the absolute compatibility is 1. Conversely, if no data event is found in the training image, the absolute compatibility value would be equal to 0.

To calculate this measure, an indicator ($Y_{i,j}$) is introduced. It takes a value of 1 if the conditioning event *i* is contained in image *j*, and a value of 0 if it is not, considering the tolerance *t*. Unlike $M_{i,j}$, $Y_{i,j}$ indicates if a pattern is present or absent in the training image, whereas $M_{i,,j}$ counts how many times it occurs in the training image. The absolute matching patterns proportion (*MP*) can be calculated using Equation 17, where I is the total number of conditioning patterns searched.

$$MP_j = \frac{\sum_{i=1}^{I} Y_{i,j}}{I}$$

Equation 17: Matching patterns proportion

A fast method to calculate this absolute compatibility is analyzing one training image at a time (as this absolute measure depends only of the analyzed training image and is independent from other images) using the direct sampling approach, with a training

image check fraction of 1 (f=1). In this case the absolute compatibility can be calculated as shown in Equation 18.

$$MP_j \approx A_j \approx \frac{L_j}{I}$$

Equation 18: Absolute compatibility

3.1.6. Ti-selector: Input, parameters and output files

The required parameter, input and output files are explained in this section. The program follows GSLIB conventions for parameter interface and input file format.

```
                                Parameters
                               ************

         START OF PARAMETERS:
Line  1: samples.out                    - samples datafile name
Line  2: 1091                           - sample number
Line  3: 100 0.5 1                      - nx, xmn, xsiz; in sample migration grid
Line  4: 100 0.5 1                      - ny, ymn, ysiz; in sample migration grid
Line  5: 1   0.5 1                      - nz, zmn, zsiz; in sample migration grid
Line  6: Training_images.out            - training images datafile name
Line  7: 100 0.5 1                      - nx, xmn, xsiz; in training images
Line  8: 100 0.5 1                      - ny, ymn, ysiz; in training images
Line  9: 1   0.5 1                      - nz, zmn, zsiz; in training images
Line 10: 25 25 0                        - search radii in x,y and z directions
Line 11: 15                             - minimum event order to analyze
Line 12: 25                             - maximum event order to analyze
Line 13: 0                              - compatibility mode: 0 for relative, 1 for absolute
Line 14: 1                              - column (for absolute mode)
Line 15: 0.1                            - tolerance parameter
Line 16: 1                              - variable type
Line 17: 0.05                           - acceptance threshold (continuous variables)
Line 18: 0.2                            - training image fraction to check
Line 19: 12345                          - random seed
Line 20: output.out                     - output file name
```

Figure 25: Parameters file of training image selection program

Line 1: Name of the input file with points (conditioning data). This file is assumed to be in a GSLIB format. It needs to contain 4 columns with from left to right x, y, z coordinates and facies information (in continuous cases the 4th column contains the variable).

Line 2: Total amount of samples contained in the input samples file.

Line 3: Number, origin (central coordinate of the node with lowest x coordinate) and spacing of nodes in the x direction of the sample migration grid. These parameters correspond to N_x, O_x and D_x.

Line 4: Number, origin (central coordinate of the node with lowest y coordinate) and spacing of nodes in the y direction of the sample migration grid. These parameters correspond to N_y, O_y and D_y.

Line 5: Number, origin (central coordinate of the node with lowest z coordinate) and spacing of nodes in the z direction of the sample migration grid. These parameters correspond to N_z, O_z and D_z.

Line 6: Name of the input file with training images. GSLIB format and node ordering is expected. Each column contains a different training image.

Line 7: Number, origin (central coordinate of the node with lowest x coordinate) and spacing of nodes in the x direction of the training images. Same parameters are expected for all the analyzed training images and nodes spacing should equal D_x to avoid scale issues.

Line 8: Number, origin (central coordinate of the node with lowest y coordinate) and spacing of nodes in the y direction of the training images. Same parameters are expected for all the analyzed training images and nodes spacing should equal D_y to avoid scale issues.

Line 9: Number, origin (central coordinate of the node with lowest z coordinate) and spacing of nodes in the z direction of the training images. Same parameters are expected for all the analyzed training images and nodes spacing should equal D_z to avoid scale issues.

Line 10: Search radii in x, y and z directions. The search radius in one direction is defined as the maximum distance, expressed in nodes of the migration grid to extend the spiral search for conditioning nodes. These parameters correspond to R_x, R_y and R_z.

Line 11: Minimum event order to analyze. If the spiral search does not find this minimum amount of conditioning nodes within the previously defined neighborhood, the conditioning event for that position is counted as invalid. This parameter corresponds to MIN_{con}.

Line 12: Maximum event order to analyze. The spiral search will stop when this amount of closest conditioning nodes is found within a neighborhood. This parameter corresponds to MAX_{con}.

Line 13: Absolute/relative compatibility measure switch. 0=relative compatibility, 1=absolute compatibility.

Line 14: Column of training image file with image to analyze. This parameter will be only considered if compatibility mode 1 is activated (absolute mode). This allows to quickly compute the absolute compatibility of each training image with the same input files.

Line 15: Tolerance parameter *t*, bound between 0 and 1.

Line 16: Variable type definition, 0 for categorical variables, 1 for continuous variables.

Line 17: Node equality threshold ε. This line is not considered if the analyzed variable is defined as categorical in *line 16*.

Line 18: Maximum training image fraction to scan *f*. By default the direct sampling based method is used (section 3.1.4), and this parameter corresponds to the number of nodes to scan, expressed as a proportion of nodes contained in the training images. If this value is set to 0, the exhaustive scanning method is used (section 3.1.3). If the compatibility mode is set to 1 (*line 13*, absolute compatibility), this line is not considered and this parameter is automatically set to 1.

Line 19: Random seed. Required number to generate the random path when direct sampling is applied. This line is not considered if an exhaustive scanning of the training image is done.

Line 20: Output file name. The generated output file is detailed in Figure 26.

Once the program starts, the amount of checked nodes is shown as a progress indicator. When finished (i.e. when the number indicated correspond to the number of nodes in the migration grid), an output file as shown in Figure 26 is generated. The information contained in this file is explained next.

```
                                Output
                               ********

START OF OUTPUTFILE:
Line  1: Number of samples located out of the training image grid:
Line  2: 0
Line  3: Number of migrated samples:
Line  4: 1091
Line  5: Number of repeated samples (not migrated and lost):
Line  6: 0
Line  7: Valid conditioning events:
Line  8: 10000
Line  9: Invalid conditioning events:
Line 10: 0
Line 11: Relative compatibility with training images (direct sampling):
Line 12: 0.0273 0.0951 0.8776
Line 13: Occurrences of conditioning events in each training image:
Line 14: 241. 840. 7753.
```

Figure 26: Output file of training image selection program

Line 2: Number of samples located out of the migration grid. It is an indication of how many samples fell outside the migration grid, as defined by lines 3, 4 and 5 of the input parameters file.

Line 4: Number of migrated samples. Number of samples that were effectively migrated to the migration grid.

Line 6: Number of repeated samples. Number of samples not migrated to the regular grid due to a previous sample already migrated to the same node.

Line 8: Valid conditioning events. Number of conditioning events found in the migration grid. The maximum amount is the same as positions (whether with or without migrated sample) contained in that grid. This output corresponds to *I*.

Line 10: Invalid conditioning events. Number of positions of the migration grid (defined by lines 3, 4 and 5 of the input parameters file) that do not have the minimum amount of migrated samples within the search radii (line 10 of parameters file) to get a conditioning event with the required minimum order to analyze (line 11 of parameters file).

Line 12: Compatibility vector. The compatibilities to the training images are ordered form left to right as the training images in the input training image set file (columns from left to right in file described in line 6 of parameters file). This line reports a single value if the absolute compatibility mode is activated. This output value(s) correspond(s) to C_j or A_j depending on whether the relative or absolute compatibility mode is activated.

Line 14: Number of occurrences of conditioning events in the training images. This vector has the same ordering as the compatibility vector. It reports a single value if the absolute compatibility mode is activated (A_j) and a vector if the relative compatibility is calculated through the direct sampling scanning method (L_j). This line is not reported when an exhaustive scanning is applied.

3.2. Simulation results validation

The development of a multiple-point simulations results validation tool is based on the training image selection method explained in section 3.1. The main differences between both algorithms are explained in this section.

3.2.1. Definition of data events in spatially exhaustive data sets

One of the main differences between both methods concerns the definition of the patterns that are used to perform the selection or validation. In the case of the selection tool, patterns of a scattered data base (drill-hole information for example) are compared to patterns of an exhaustive data base (the training image). Due to the scattered nature of the data, a sample migration needs to be performed, and only a limited amount of patterns can be extracted from the data.

In the case of the results validation tool, two exhaustive databases are compared, the training image and the resulting models. As every position in these data bases is informed, no migration is required, and a spiral search (as explained in section 3.1.2) does not accomplish the desired pattern definition, as always the same closest neighbors would be found and the size and geometry of the patterns would remain unchanged. The main problem related to this method is that only very small-scale patterns would be analyzed.

The results validation algorithm starts with a loop that runs over all the positions of the model using a regular path. At each position the algorithm simulates the order of the data event to search, which is contained between a user-defined minimum and maximum, and follows a uniform distribution between both values. A search spiral is started, but in this case, it goes from a nearest to a farthest point within the search neighborhood, leaving between the selected nodes a user-defined spacing, as shown in Figure 27 where every fourth node found is used. The size of the node spacing would represent the scale of the patterns whose reproduction is verified by the algorithm. When the required amount of conditioning nodes is reached, the spiral search is stopped.

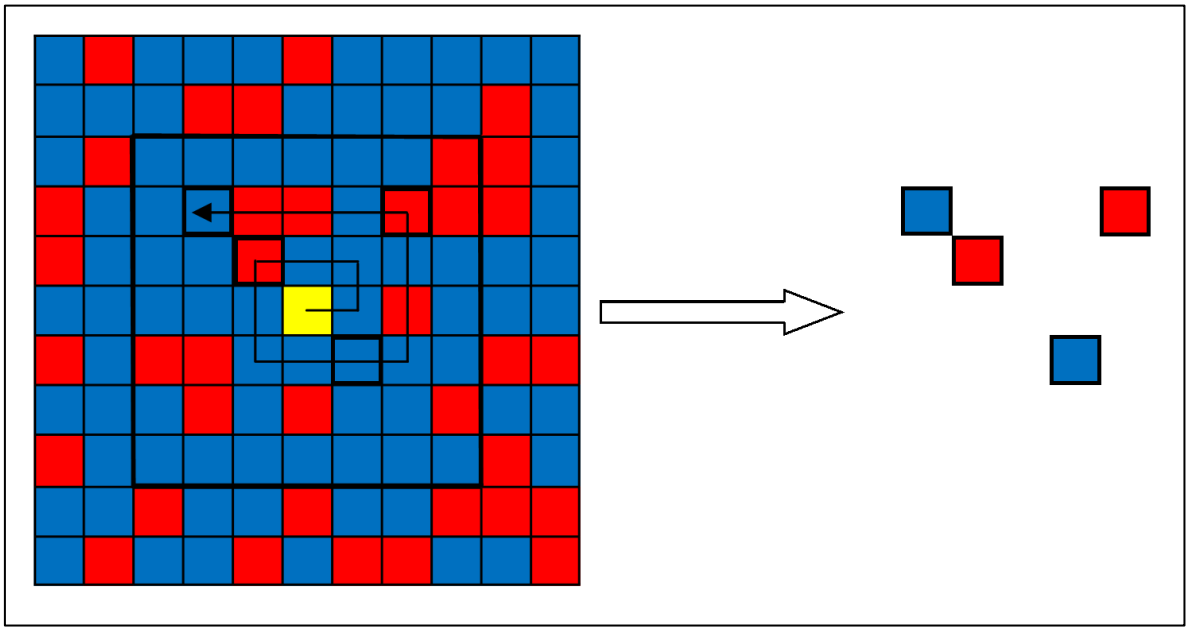

Figure 27: Spiral search with node spacing (left) and defined data event (right)

Again, the offsets to the selected nodes and their facies (or the value of a continuous variable) are saved in a list. The data event is not considered for comparison if not enough nodes are found within the defined neighborhood (if the maximum search radius is reached without finding minimum required conditioning nodes), but in this case it is important to remark that this could only happen in the borders of the results grid, or if the user defines a not big enough search neighborhood for the required minimum order event and node spacing.

3.2.2. Search of data event in the training image

The proposed results quality measurement corresponds to the proportion of patterns contained in the results (defined in section 3.2.1) that are matched by the training image. The search of occurrences of the data events in the image is done using a random path. Due to the high CPU times involved in this process, the user is allowed to compute an approximation of the results quality measure defining the fraction *f* of the training image that is scanned in the search of a match.

As in the case of the selection method, tolerance parameters *t* and ε are considered which correspond to the allowed pattern mismatch fraction and the node equality threshold (see Equation 12 and Equation 13).

The results validation tool presented searches occurrences of results patterns in the training image, therefore checking the proportion of results patterns that reproduce characteristics of the training image. However, the search is also done at the reverse, computing the proportion of training image patterns that are contained in the results. This measure reflects the completeness of the results, as it gives an idea of the proportion of the training image that is reproduced by the results.

Therefore, the whole procedures explained in this section and section 3.2.1, are repeated in the inverse way to obtain the second measure.

3.2.3. Results-validator: Input, parameters and output files

In this section, the required input and output files of the results validation program are explained. As in the case of the training image selection program, GSLIB conventions are followed.

```
                                  Parameters
                                 ************

START OF PARAMETERS:
Line  1: realization-1.out           - results file name
Line  2: 90 17500 25                 - nx, xmn, xsiz; in results grid
Line  3: 65 113050 25                - ny, ymn, ysiz; in results grid
Line  4: 42  2682.5 15               - nz, zmn, zsiz; in results grid
Line  5: 1                           - column in results file with results to check
Line  6: training-image.out          - training image datafile name
Line  7: 90 17500 25                 - nx, xmn, xsiz; in training image
Line  8: 65 113050 25                - ny, ymn, ysiz; in training image
Line  9: 42  2682.5 15               - nz, zmn, zsiz; in training image
Line 10: 4                           - column in training image file with variable
Line 11: 20 20 10                    - search radii in x,y and z directions
Line 12: 15                          - minimum event order to analyze
Line 13: 30                          - maximum event order to analyze
Line 14: 10                          - spacing
Line 15: 0                           - tolerance parameter
Line 16: 0                           - variable type
Line 17: 0.1                         - acceptance threshold (continuous variables)
Line 18: 0.1                         - grid fraction to check
Line 19: 98765                       - random seed
Line 20: output.out                  - output file name
```

Figure 28: Parameters file of results validation program

Line 1: Name of the input file with results of the simulation. This file is assumed to be in a GSLIB format and node ordering is expected.

Line 2: Number, origin (central coordinate of the node with lowest x coordinate) and spacing of nodes in the x direction of the results grid.

Line 3: Number, origin (central coordinate of the node with lowest y coordinate) and spacing of nodes in the y direction of the results grid.

Line 4: Number, origin (central coordinate of the node with lowest z coordinate) and spacing of nodes in the z direction of the results grid.

Line 5: Column of the input results file which contains the simulated variable values.

Line 6: Name of the input file with training image. GSLIB format and node ordering is expected.

Line 7: Number, origin (central coordinate of the node with lowest x coordinate) and spacing of nodes in the x direction of the training image. The same node spacing as in the results grid is expected to avoid scale issues.

Line 8: Number, origin (central coordinate of the node with lowest y coordinate) and spacing of nodes in the y direction of the training image. The same node spacing as in the results grid is expected to avoid scale issues.

Line 9: Number, origin (central coordinate of the node with lowest z coordinate) and spacing of nodes in the z direction of the training image. The same node spacing as in the results grid is expected to avoid scale issues.

Line 10: Column of the training image file that contains the simulated variable.

Line 11: Search radii in x, y and z directions. The search radius in one direction is defined as the maximum distance, expressed in nodes to extend the spiral search for data event nodes.

Line 12: Minimum event order to analyze.

Line 13: Maximum event order to analyze. Before each data event is defined, a value between this maximum and the minimum event order is simulated from a uniform distribution. The simulated value corresponds to the event order that is finally analyzed. If the user requires analyzing only events of a certain order, minimum and maximum values should be defined equal.

Line 14: Node spacing in the data event. This parameter reflects the scale of the analyzed patterns. A small spacing translates into an analysis of short scale pattern reproduction, while a big spacing translates into the analysis of large scale patterns. It is responsibility of the user to define correctly node spacing, event order and search radii, to allow the program to find valid data events within the neighborhood.

Line 15: Tolerance parameter *t*, bound between 0 and 1.

Line 16: Variable type definition, 0 for categorical variables, 1 for continuous variables.

Line 17: Node equality threshold ε. This line is not considered if the analyzed variable is defined as categorical in *line 16*.

Line 18: Maximum grid fraction to scan *f*. This parameter corresponds to the number of nodes to scan expressed as a proportion of nodes contained in the training image or results grid (depending on whether the algorithm is performing the scan of the training image searching results patterns or the reverse).

Line 19: Random seed. Required number to generate the random path.

Line 20: Output file name. The generated output file is detailed in Figure 29. The information contained in this file is explained next.

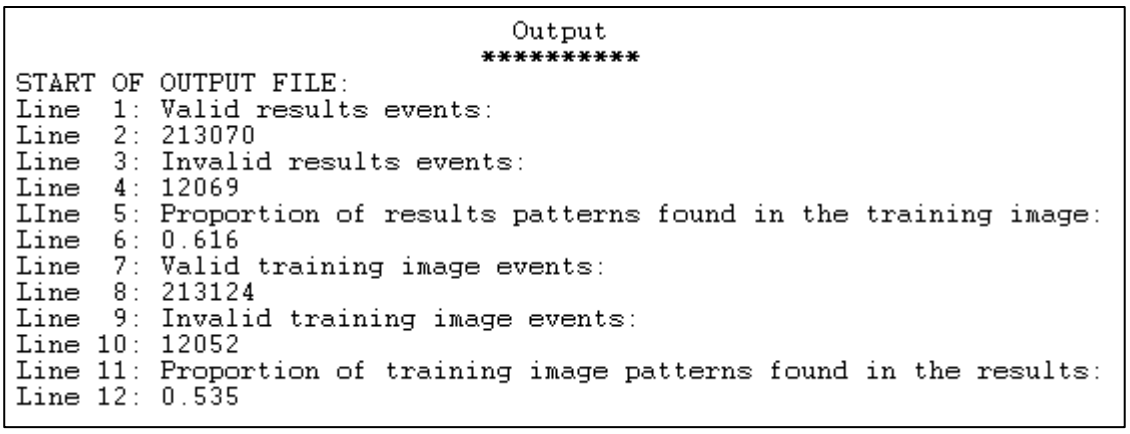

Figure 29: Output file of results validation program

Line 2: Valid data events found in the results grid. The maximum amount is the same as nodes located in the results grid minus the amount of masked nodes (to mark topography for example).

Line 4: Invalid data events found in the results grid. It corresponds to the number of positions of the results grid that do not have the amount of nodes within the search radii considering the node spacing and the analyzed data event order. Invalid data events should only occur at the borders of the grid or at nodes that neighbor a group of masked nodes for example. A high number indicates that the search neighborhood is not big enough for the expected event orders and node spacing.

Line 6: Proportion of results patterns found in the training image. This value is bound between 0 (no results patterns are found in the image, worst training image reproduction) and 1 (all the results patterns are found in the image, best training image reproduction).

Line 8: Valid data events found in the training image grid. The maximum amount is the same as nodes located in the image grid minus the amount of masked nodes (to mark topography for example).

Line10: Invalid data events found in the training image grid. The same comments as for *line 4* should be considered for this line.

Line 12: Proportion of training image patterns found in the results. This value is bound between 0 (no training image patterns are found in the results, incomplete training image reproduction) and 1 (all the training image patterns are found in the results, complete training image reproduction).

4. CASE STUDY RESULTS AND DISCUSSION

In this section, several application examples are presented in order to assess the proposed methods. The first example consists of a conceptual synthetic case, while the second one represents a real hydrology case study. Both examples are used to test the training image selection tool. The third and final case study corresponds to a real mine-scale case study, applied to the Escondida Norte deposit located in northern Chile. This example is used to assess the applicability of the simulation results validation tool as well as the training image selection tool to a real case.

4.1. Conceptual case studies

4.1.1. Case study 1

In this case, three different parameter sets were used to construct binary training images using the TiGenerator tool implemented in SGeMS (Maharaja 2008, Boucher et al. 2010). These parameters are presented in Table 2. All parameters were set such as to obtain the same proportion of 0.3 for facies 1. For each parameter set, two realizations of 100 by 100 nodes were constructed: the first realization is used as training image, and the second realization is used to extract data. Since both realizations are obtained with the same parameters, the data extracted from one realization should show a high level of compatibility with the other realization. Performing such comparisons allows validating the proposed approach.

Training Image	Geometry	Orientation	Structure Characteristics			Proportion
1	Ellipsoid	60° ± 15°	**Max Rad**	**Med Rad**	**Min Rad**	0.302
4			15	50	5	0.302
2	Sinusoid	0° ± 10°	**Amplitude**	**Wavelength**	**Width**	0.304
5			15	3	1	0.331
3	Cuboid	0°	**Width**	**Height**		0.320
6			3	1		0.300

Table 2: TiGenerator parameters, case study 1

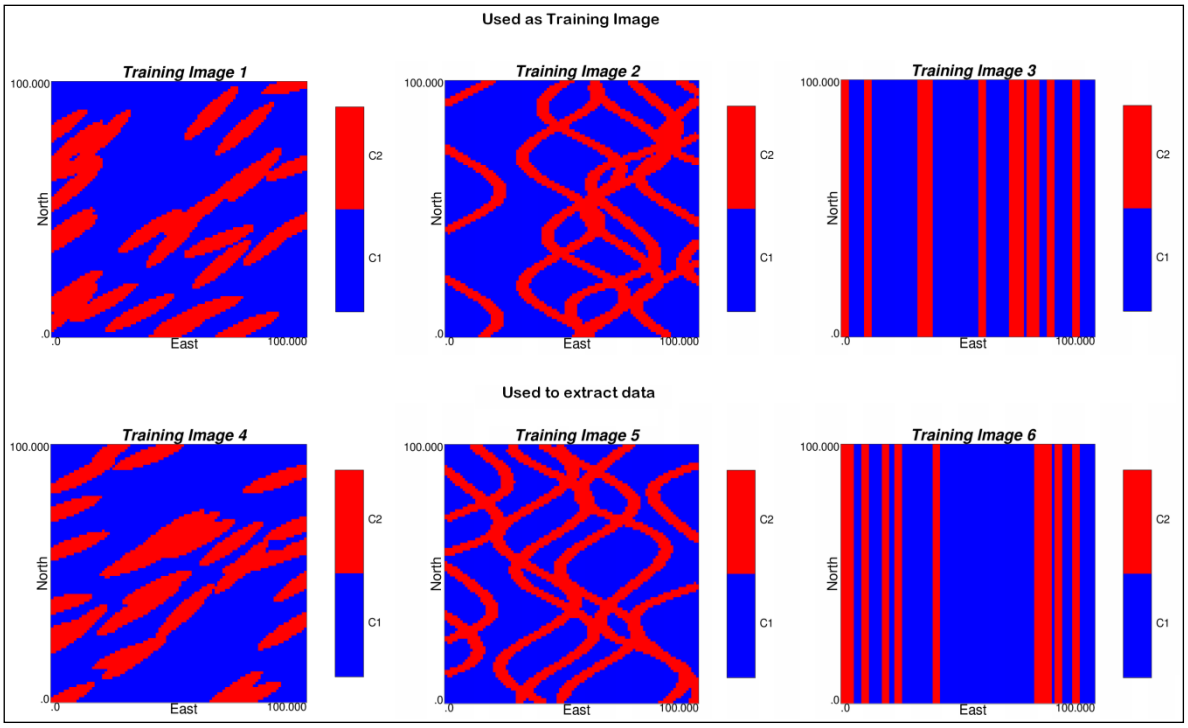

Figure 30: Training images of case study 1

For each parameter set, data are extracted from the second realization. Three sample sets were obtained, consisting of 1091, 222 and 80 random samples, which are used for validation against the training images.

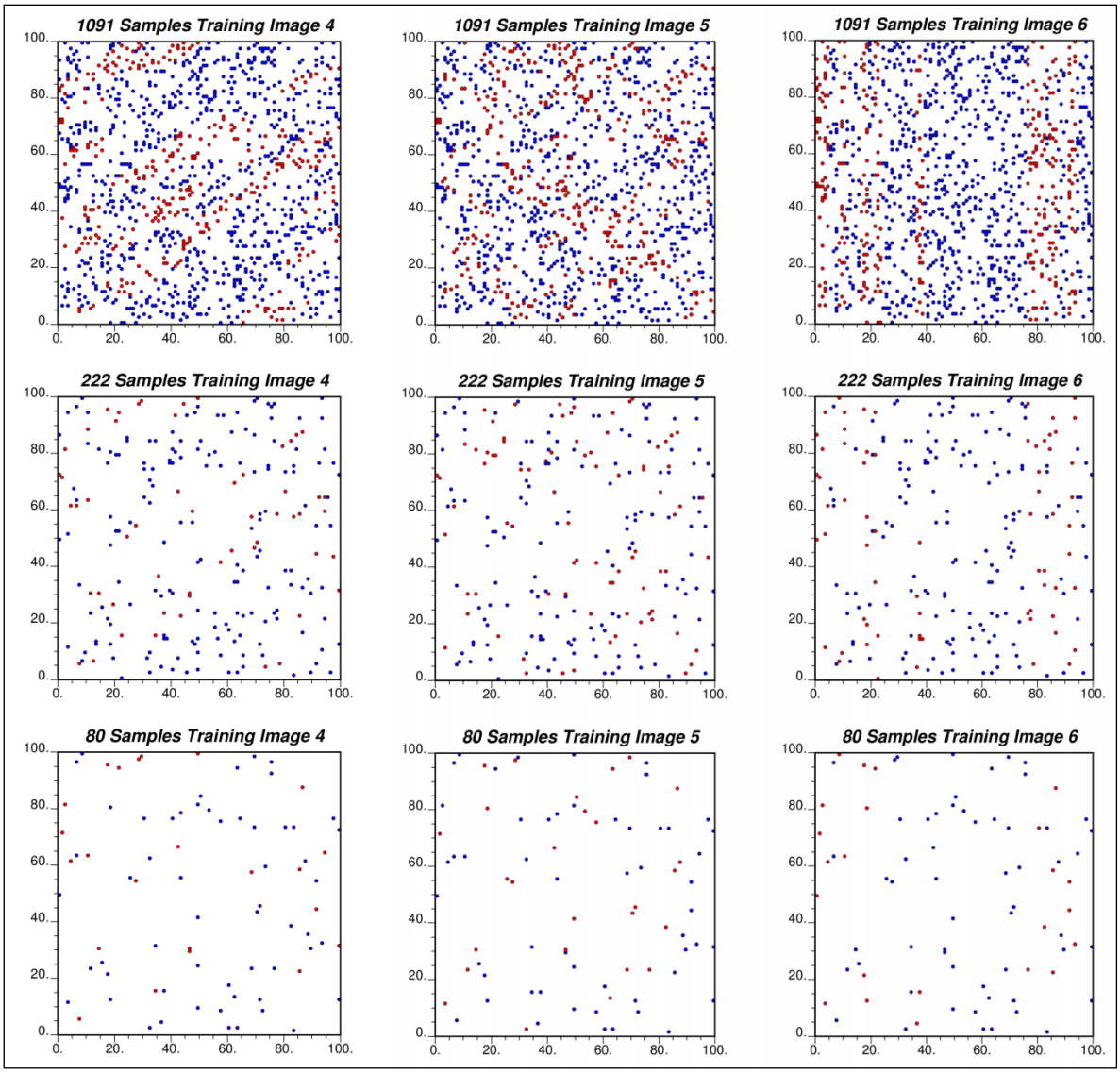

Figure 31: Samples, case study 1

To compute relative and absolute compatibilities fixed parameters of 25 were used for search radii in the x direction and y direction and 0 is used for tolerance *t*.

Compatibilities between the set of first realizations and the extracted data were obtained considering conditioning events of orders: 1, 2, 3, 4, 5, 10, 15, 20, 25 and 30. Results are shown in Figure 32. It is interesting to observe how the relative compatibility values behave for low order (≤5) conditioning events. Since the analyzed images have approximately the same facies proportions, the relative compatibility shows no preference for any image when assessing first order conditioning events. As the event order increases, a better compatibility measure with the expected images is observed. This trend is visible even in cases with a small number of conditioning data (bottom row Figure 32) although it is less marked in this case.

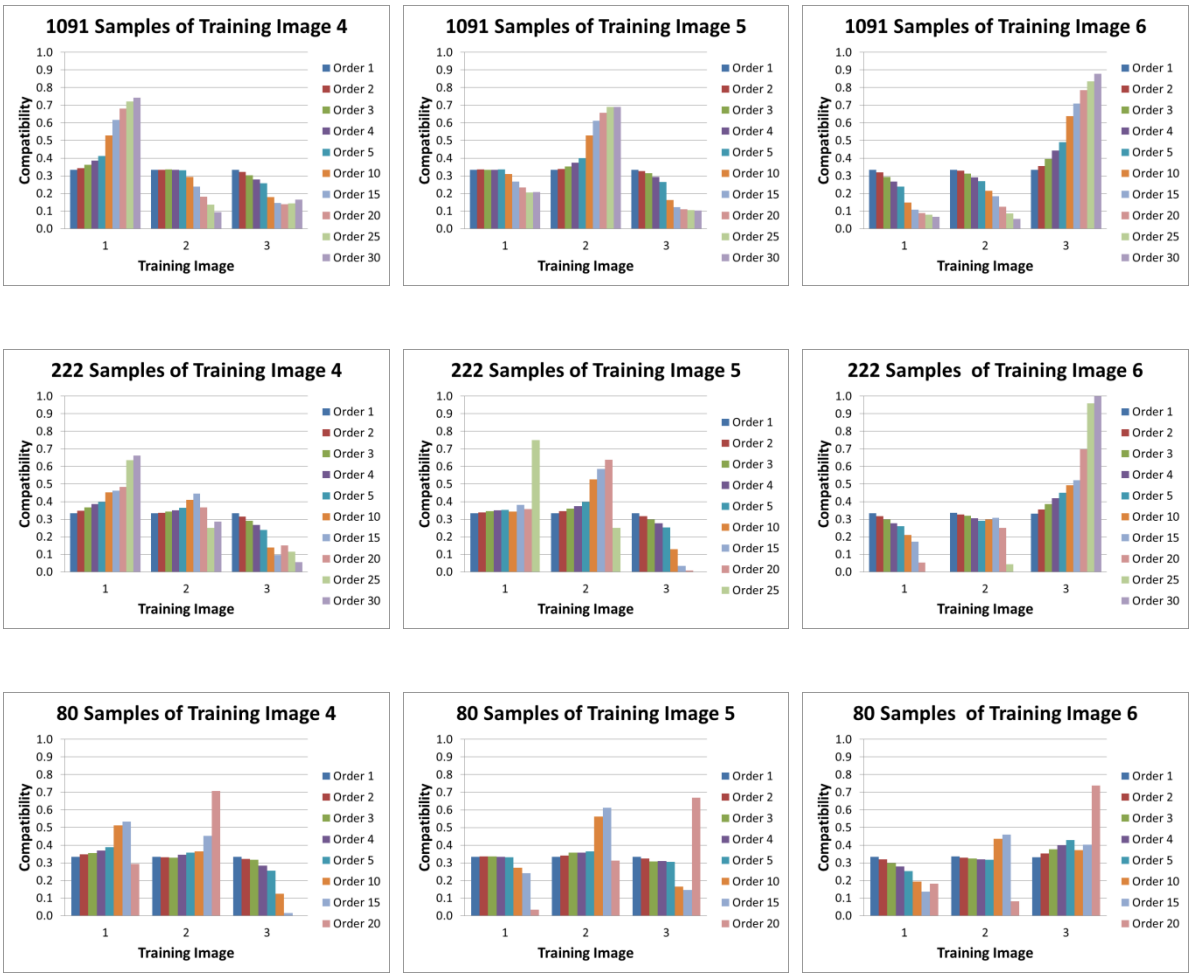

Figure 32: Relative compatibilities obtained with exhaustive scanning, case study 1

With an increasing amount of conditioning data, the compatibility measure becomes more stable for high order data events, and the results correspond to what one would expect: no possibility to distinguish between training image for low orders (because all the images have the same marginal proportion), and an increasing differentiation with higher orders. For cases with 222 and 80 samples results only remain stable until the 20^{th} and 15^{th} event orders respectively. This can be explained due to a poor and non-representative inference of such high order spatial structures in poorly informed scenarios (low amount of drill-holes). This insufficient amount of data for proper inference can be diagnosed by analyzing the change in the amount of patterns of a given order found in the data grid. This value is reported in the output file at line 8 (see Figure 26). The results of this analysis are shown in Figure 33.

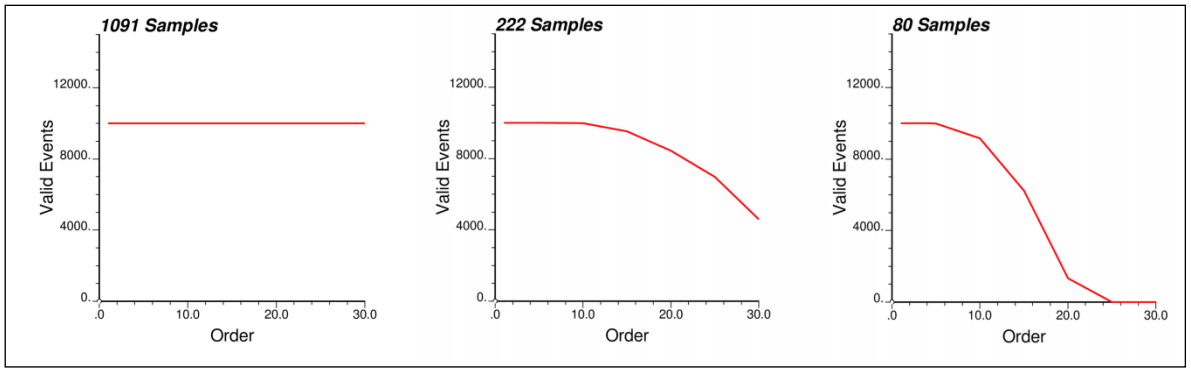

Figure 33: Valid conditioning events by sample density scenario and event order

This allows deciding whether the inference can be considered representative or not. In the 222 and 80 samples cases, it is possible to see an important drop in the amount of patterns found when the order of the conditioning events increases. In the case of 1091 samples, the high data density avoids this problem.

The same case study is used to test the direct sampling based algorithm. For each case, the same parameters and inputs considered for the exhaustive scanning method are used. Additionally a training image scan proportion of f=1 is used (the complete training image is analyzed). The results are summarized in Figure 34.

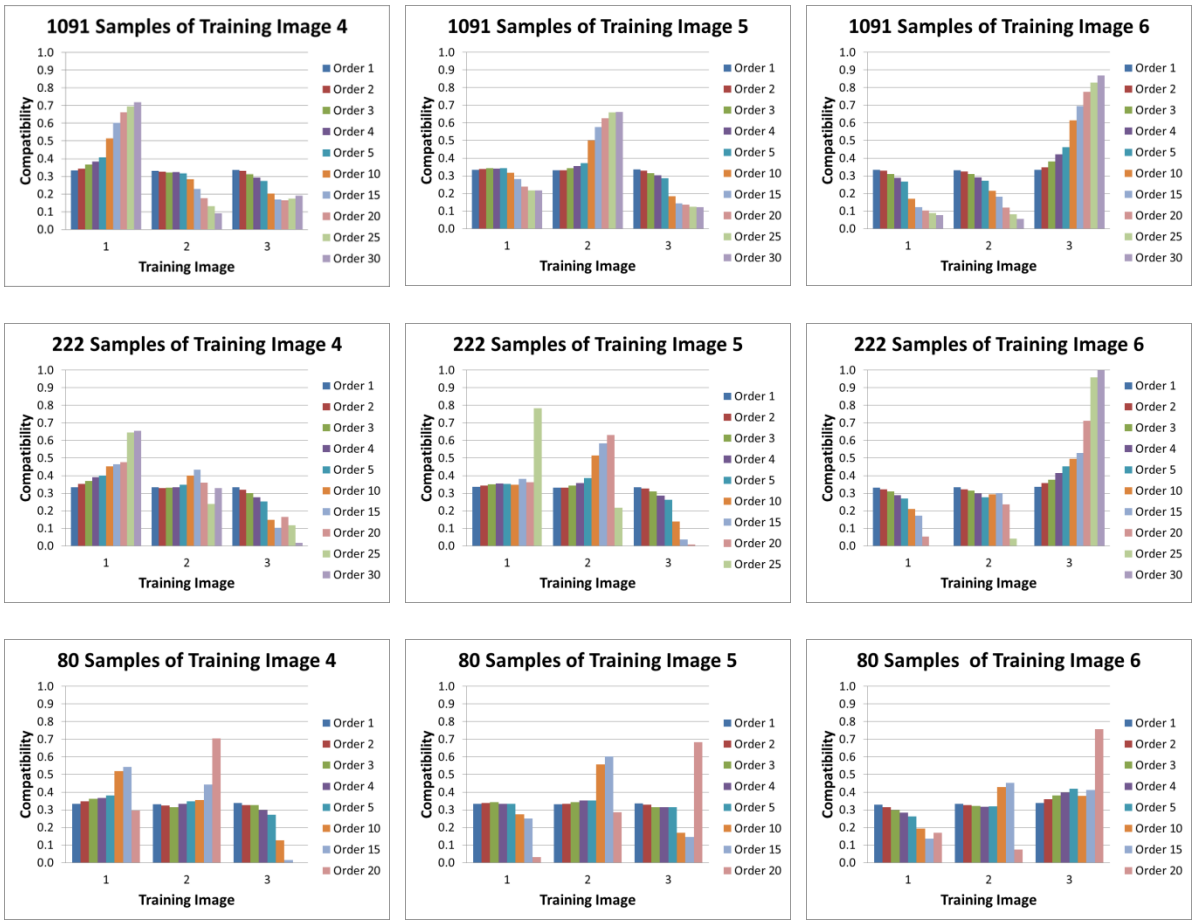

Figure 34: Relative compatibilities obtained with direct sampling approach, case study 1

It is possible to observe the high similarities between the results obtained using the direct sampling approach and the exhaustive scanning. The main advantage of the former is the speed of the analysis. A comparison of the computing times is shown in Figure 35.

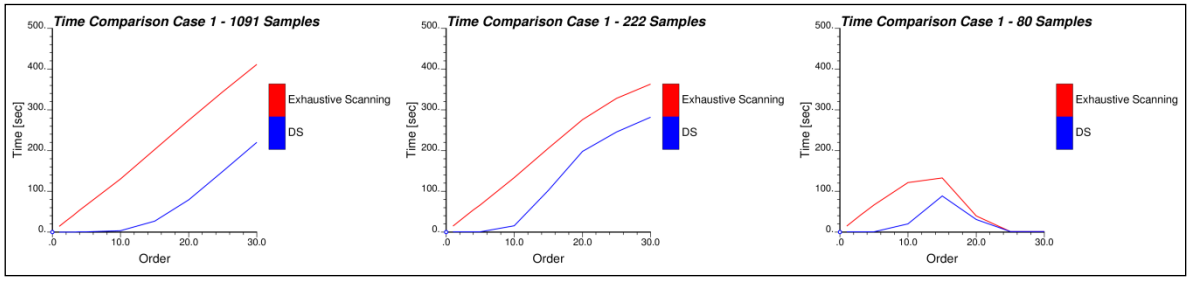

Figure 35: Time comparison exhaustive scanning versus direct sampling, case study 1

The computing time reduction is substantial when the analyzed data events are of low order. For high order events, the time reduction is not as important, but still significant, especially for a large number of samples. As big scale events occurrences in a relatively small image are rare or even inexistent, the direct sampling method needs to analyze all positions in the images to find a matching occurrence (when the image checking proportion *f* is set to 1). This effect explains the similar increase in computing time when the amount of samples decreases from 1091 to 222. However, as soon as computational gains reduce, it is also a sign that matching data events are hard to find, and that the results significance is therefore small.

The particular behavior of the computing times in the 80 samples case is due to the drop on the conditioning events found (see Figure 33). For both the exhaustive scanning and the direct sampling method, the occurrences search takes most of the computing time. It is the process where the direct sampling method saves computing time in comparison with the exhaustive scanning approach. Therefore, a lower amount of valid conditioning events explains similar computing times for both exhaustive scanning and direct sampling methods.

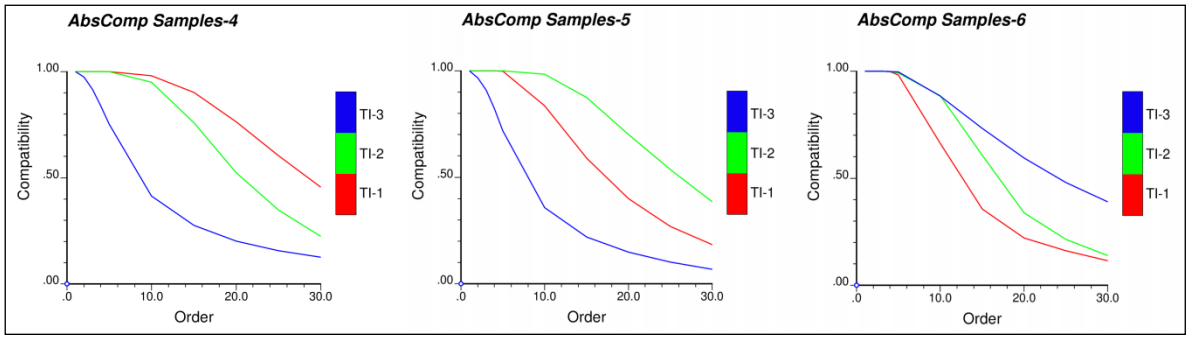

Figure 36: Absolute compatibilities, case study 1

Finally, absolute compatibility is obtained using the same setting. Results regarding the 1091 samples set are summarized in Figure 36. As expected higher absolute compatibilities with data sets are obtained when analyzing the training image with the same spatial features (i.e. samples from TI4 are similar to TI1, samples from TI5 are similar to TI2, and samples from TI6 are similar to TI3). It is important to remember that this compatibility takes a value of 1 if every analyzed conditioning event is contained in the training image. The maximum value is obviously reached for every analyzed image when low order events are considered. However, this indicator should be considered an aid for training image quality measurement only in terms of higher order event compatibility, as there are better tools available to measure first and second order spatial compatibility through histogram and variogram comparisons.

4.1.2. Case study 2

For this second case a reference dataset image is compared with four training images which are known to be consistent. It is expected to obtain equivalent compatibility values for all images and for all orders. For the purpose of this test, five realizations were constructed with the SGeMS TiGenerator and identical parameters shown in Table 3.

Training Image	Geometry	Orientation	Amplitude	Wavelength	Width	Proportion
Sampled	Sinusoid	0° ± 20°	10	25	2	0.277
1	Sinusoid	0° ± 20°	10	25	2	0.260
2	Sinusoid	0° ± 20°	10	25	2	0.266
3	Sinusoid	0° ± 20°	10	25	2	0.266
4	Sinusoid	0° ± 20°	10	25	2	0.280

Table 3: TiGenerator parameters, case study 2

1091 samples are obtained from the first realization, the other 4 being considered as training images for comparison. The compatibility results are shown in Figure 38 and are computed with the same fixed parameters of case study 1: search radii of 25 nodes in the x and y direction.

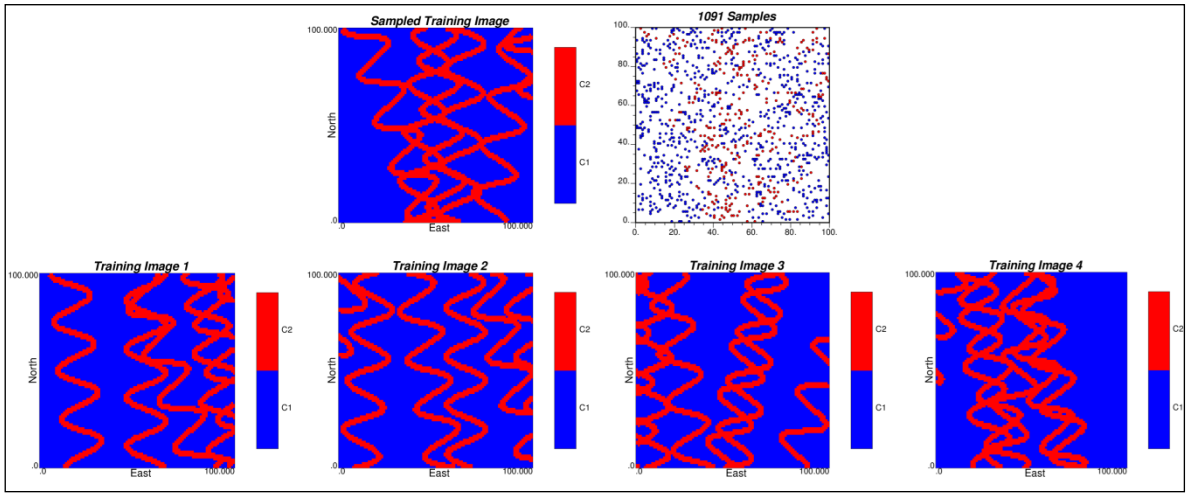

Figure 37: Training images and samples, case study 2

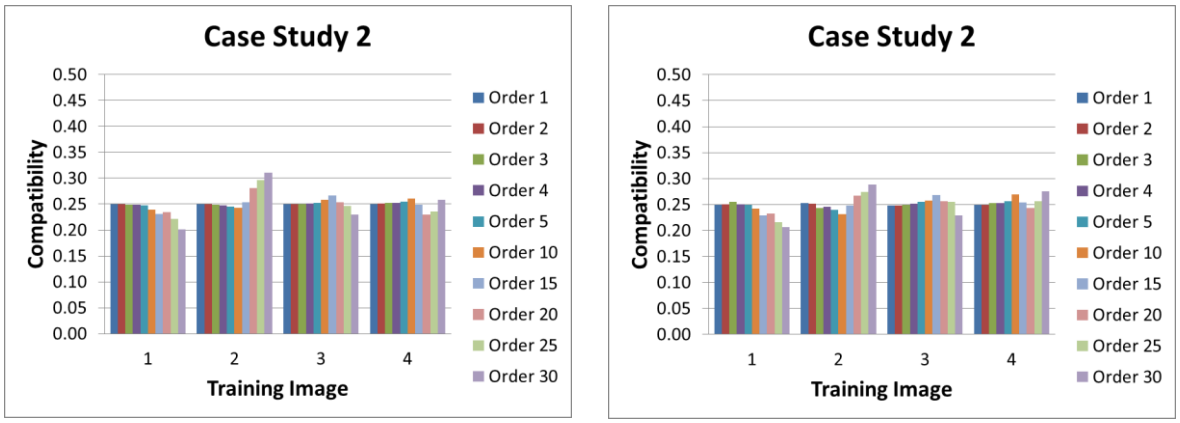

Figure 38: Relative compatibility obtained with exhaustive scanning (left) and direct sampling (right), case study 2

The results concur with the expected behavior. The same relative spatial compatibility between the scattered data and every training image is observed for both low and high conditioning event orders. For high orders, the compatibility varies slightly. This effect can be explained by an increase in the conditioning event size. In cases with a high number of conditioning nodes, the spiral search needs to visit farther nodes. With the largest patterns, the number of repetitions in the image can be low, especially for relatively small training images. As for case study 1, the results of the direct sampling based method are shown in Figure 38 (using the same parameters and f=1).

Once again, a computing time comparison and an absolute compatibility measurement are performed. Results are shown in Figure 39. The results are similar than those obtained with the base algorithm, with improved computation time. This conforms to the conclusions drawn from case 1.

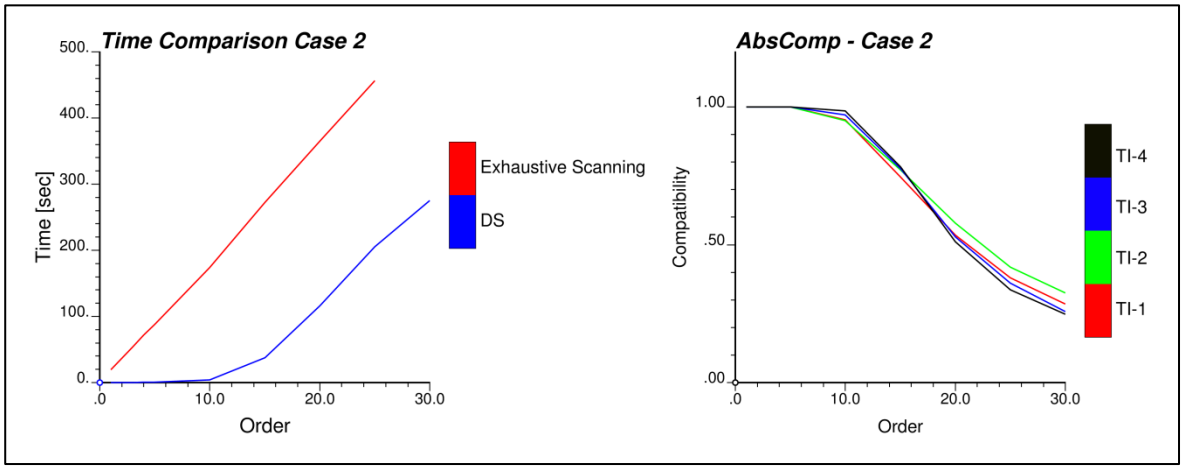

Figure 39: Time comparison exhaustive scanning versus direct sampling (left) and absolute compatibilities (right).

4.2. Realistic hydrology case study

This case study corresponds to a realistic alluvial formation obtained with process-based simulation (Lopez 2003; Chugunova and Hu 2006). The model of the aquifer consists of 1385 one meter thick layers (of 101 by 101 nodes in the X and Y directions), the lowest one is layer zero, whilst the highest layer corresponds to number 1384. The structures in the model show a transition from lobes in the lower layers to channelized structures in the upper layers (see Figure 40 and Figure 41). These 2-D layers were used to get training images and conditioning data. The layers selected as training images were chosen every 140 to 150 meters in the vertical direction. Scattered conditioning data of 1093 and 222 samples were extracted from layers 5 meters below the layers used as training images. Each pair consisting of a training image and the associated scattered data (taken 5 layers below) are termed a data set, and each data set is attributed a number (see Table 4). As the aquifer spatial characteristics and facies proportions change smoothly in the vertical direction, the result of this selection is 10 different training images, and 10 scattered data sets, each training image being similar, but not exactly matching the corresponding data set. The training image selection tool is used to find the compatibility between sampled sets and training images and verify if the method can be used to assemble the correct pairs data set/training image. The selected layers are summarized in Table 4, training image layers and sampled layers shown in Figure 40 and Figure 41 respectively.

Data set #	Training Image layers	Sampled data set layers
1	5	0
2	150	145
3	295	290
4	435	430
5	585	580
6	725	720
7	875	870
8	1020	1015
9	1165	1160
10	1310	1305

Table 4: Summary of sampled and training layers, case study 3

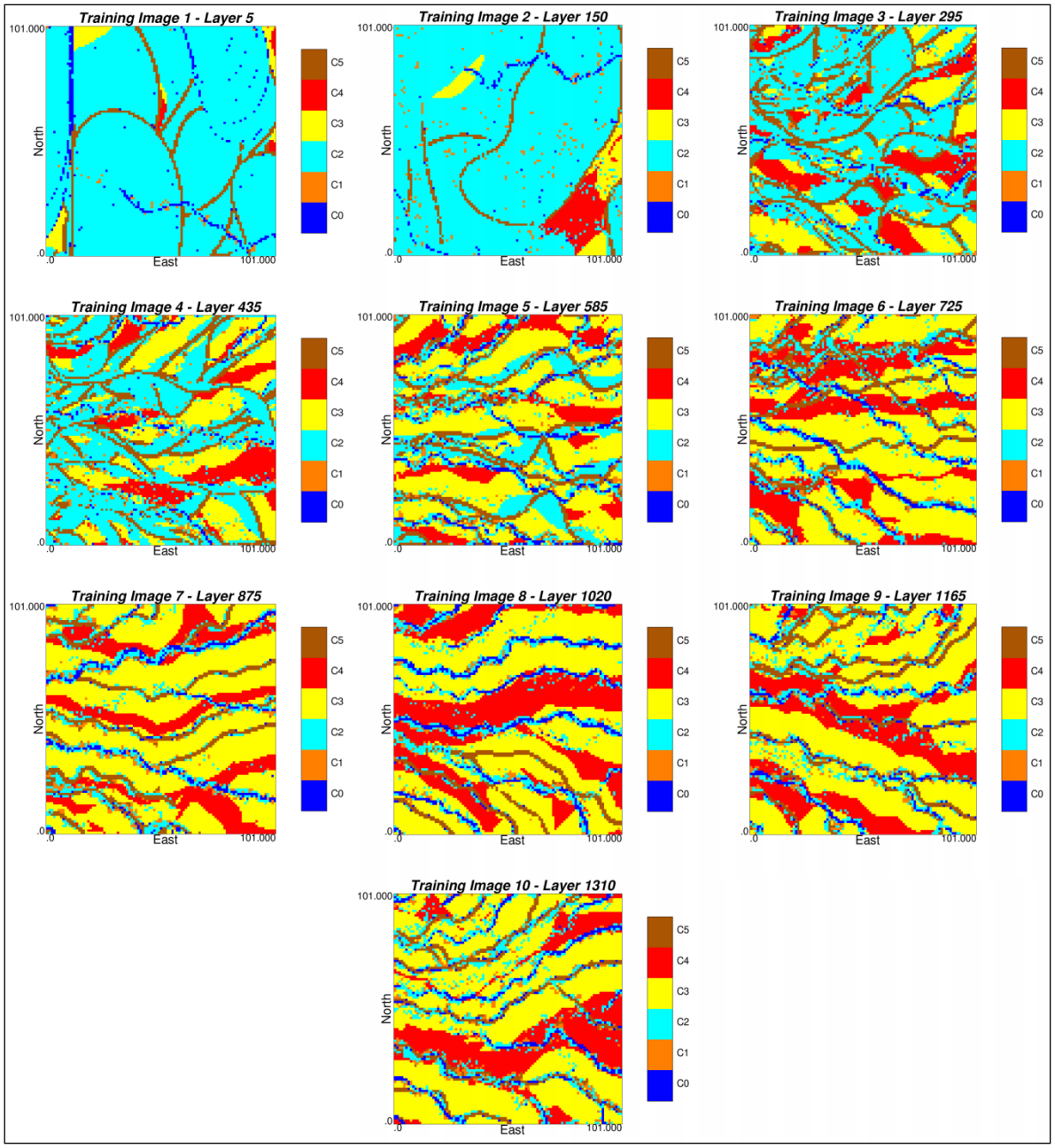

Figure 40: Summary of layers used as training images, case study 3

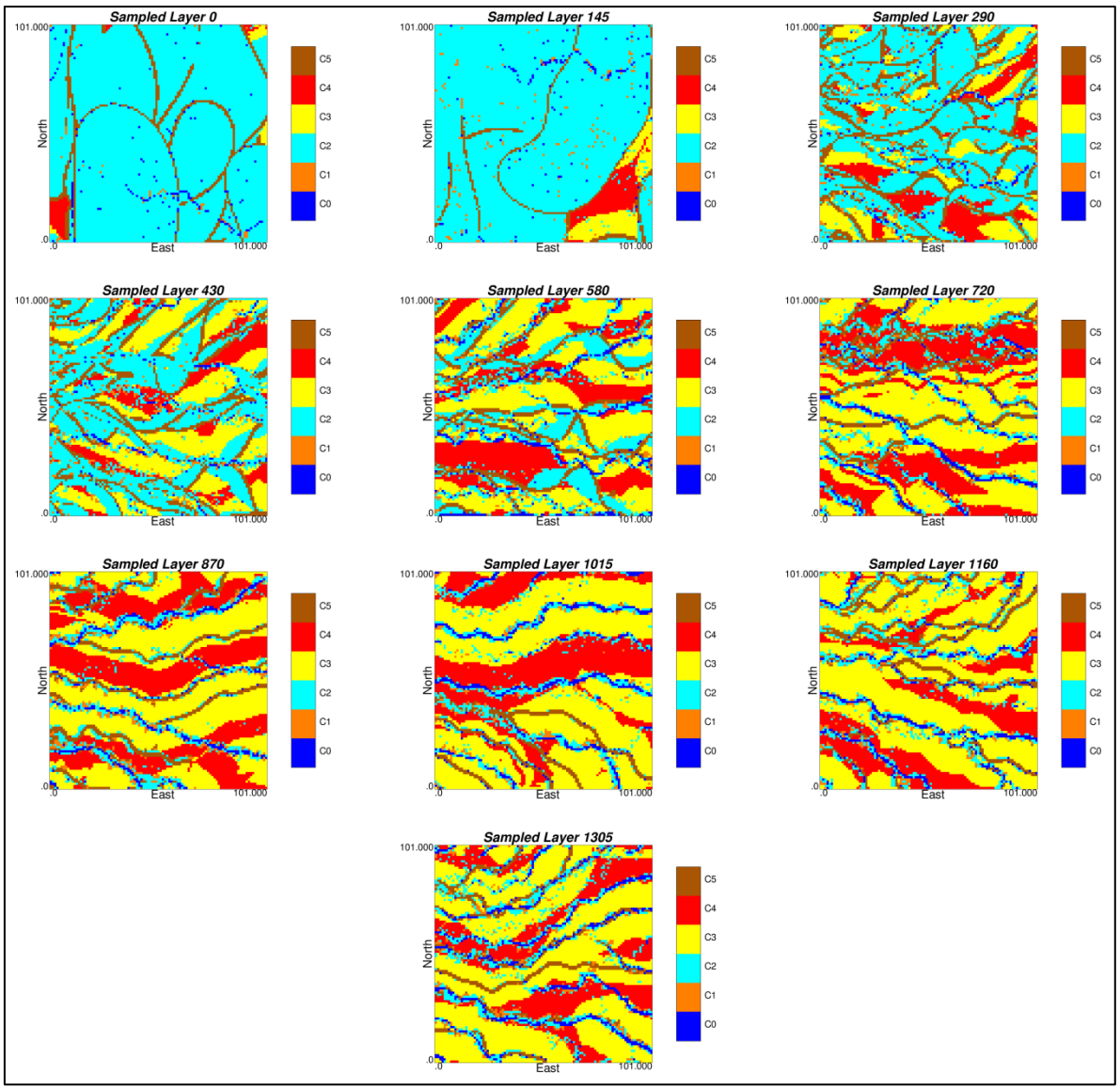

Figure 41: Summary of layer used to get samples, case study 3

Relative compatibilities between training images and 1093 and 222 samples data sets were calculated using both exhaustive scan and direct sampling approaches. Results regarding layers 0, 430, 580 and 1015 are summarized in Figure 42 and Figure 43. The remaining results are attached in Appendix A.

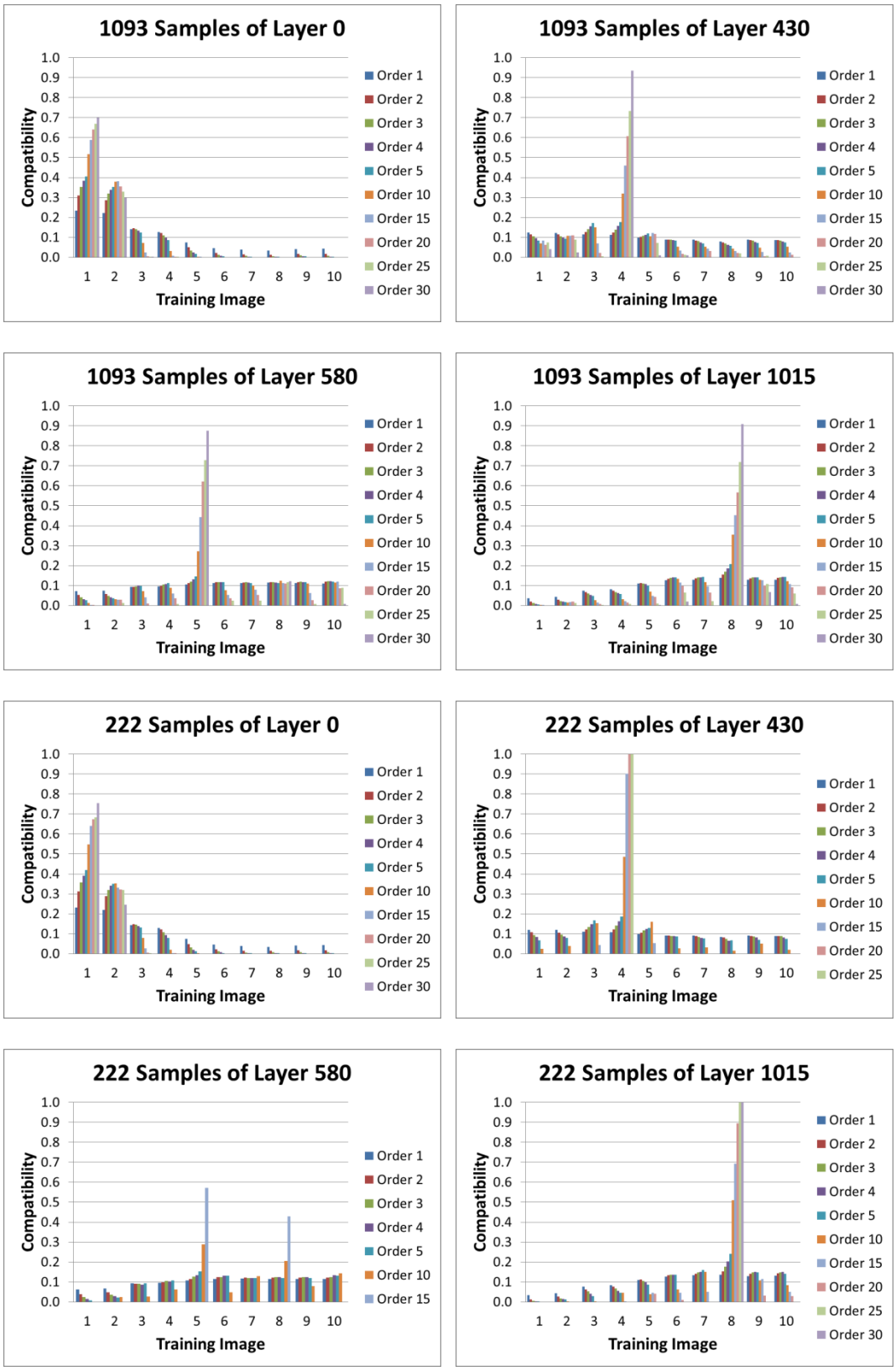

Figure 42: Relative compatibilities obtained with exhaustive scanning, case study 3

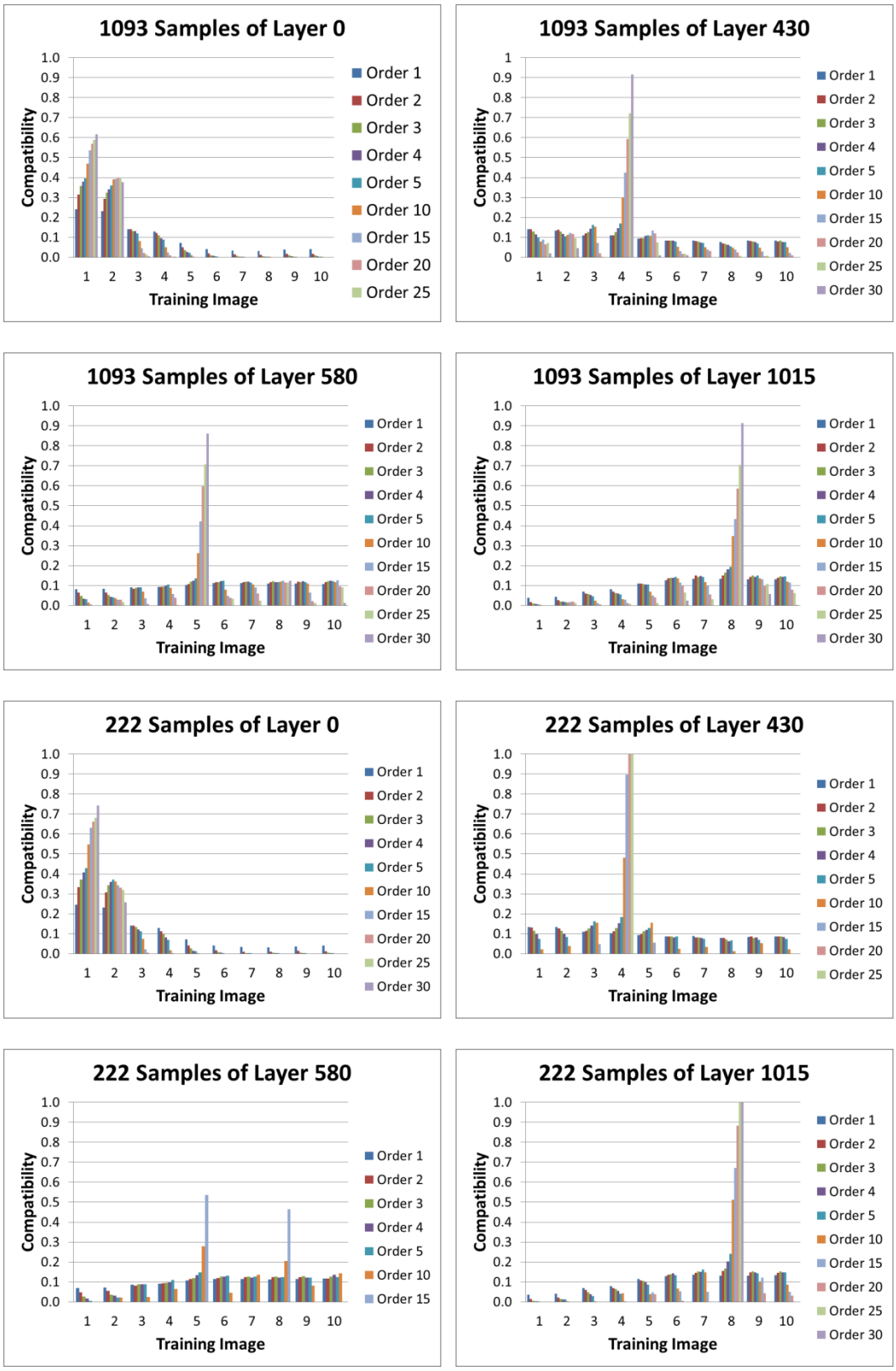

Figure 43: Relative compatibilities obtained with direct sampling approach, case study 3

In case studies 1 and 2, similar facies proportions in the training images caused relative compatibilities to become noticeably different only when analyzing conditioning events of higher order. In this case, large differences between images in terms of spatial structures and proportions result in higher variations of relative compatibility, even when low order events are considered and a lower amount of samples is available. Care should be taken when analyzing the results, as these can be driven by the differences in proportions rather than in spatial structure, especially when low order events are considered. Spikes observed in layer 580 can be explained due to a low amount of high order conditioning events found when conditioning information is scarce (222 conditioning samples). In those cases, a single mismatch (if a conditioning event is matched by the wrong image) affects considerably the results.

Once again exhaustive scanning and direct sampling methods show similar results in all cases. The CPU time comparisons between both methods are summarized in Figure 44.

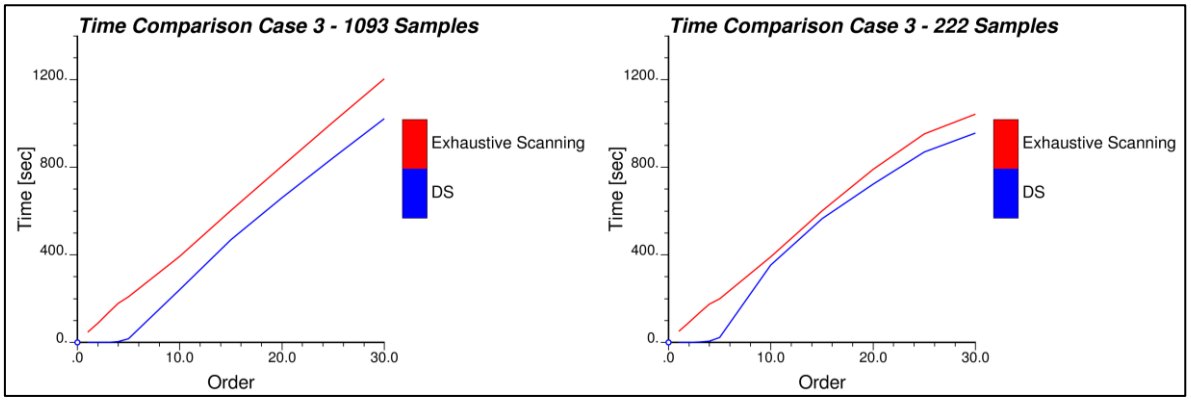

Figure 44: Time comparison exhaustive scanning versus direct sampling, case study 3

In the case of 1093 samples, a slightly lower time reduction to well informed scenarios of case studies 2 and 3 is achieved. The higher complexity and non-stationarity of images of this case study explains the reduced computational improvement, due to the difficulty of finding an event replicate in these scenarios, even in compatible training images. This can be explained due to non-stationarity restricting the possible area of the image where a match can be found, lengthening search loops of the algorithm.

Finally, absolute compatibilities are computed for this case study. Results regarding layers 0, 430, 580 and 1015 are summarized in Figure 45. The remaining results are attached in Appendix B. Absolute compatibility results behave as expected, assembling correctly pairs training image/scattered data. In the case of samples from the lowest layer, a simpler spatial structure characterized by a high proportion of nodes belonging to category n° 2, causes this measure staying stable and with relative high values.

However, the most interesting result related with case study 3 is the applicability of the method to assess training image compatibility with data in real cases.

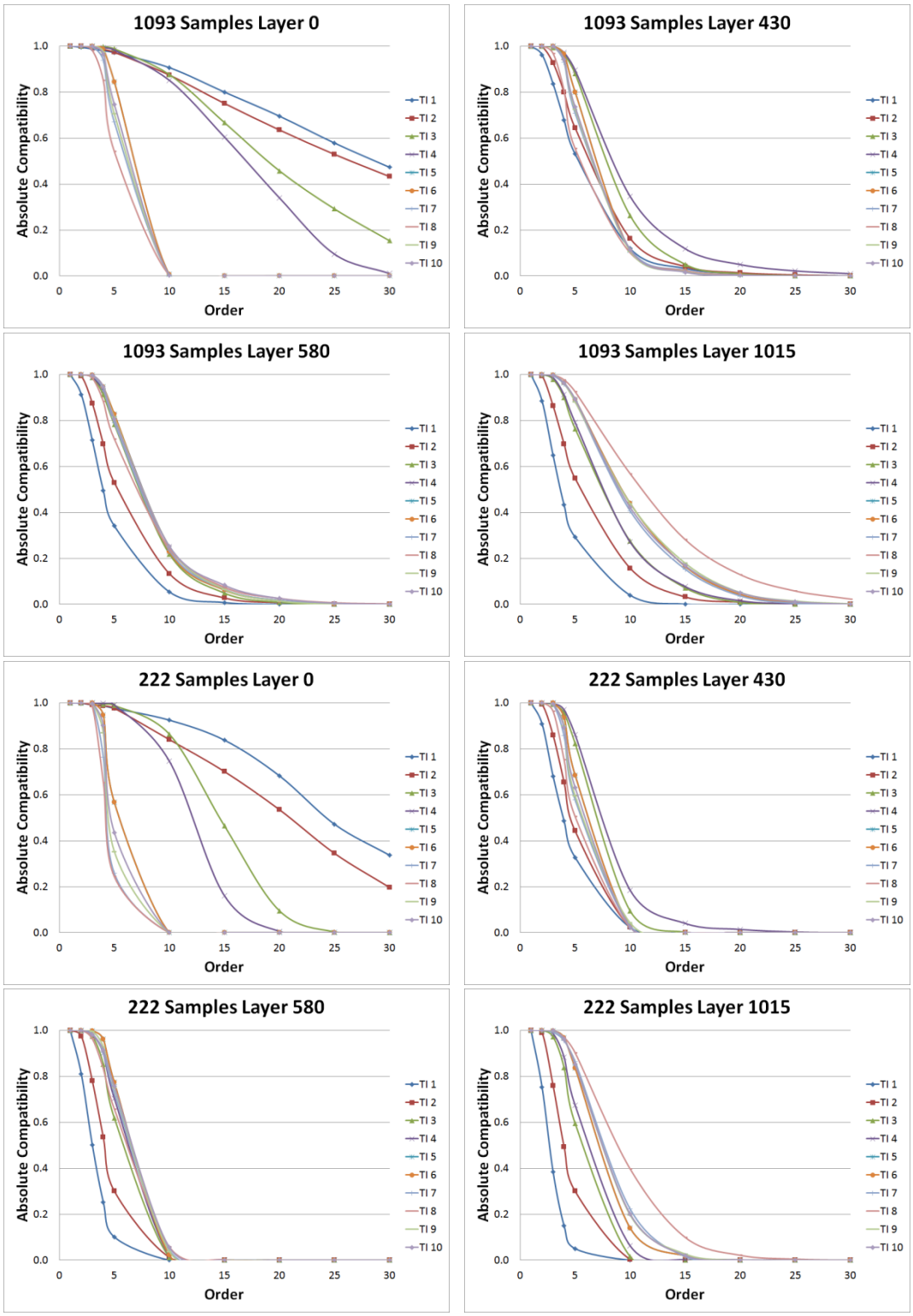

Figure 45: Absolute compatibilities, case study 3

4.3. Mining case study: Reconstruction and validation of Escondida Norte lithological model

This case study is used to test the results validation tool and address its applicability on a real case study. This example shows the 3-D modeling of lithologies at the Escondida Norte deposit using the Direct Sampling multiple-point simulation algorithm (Mariethoz et al. 2010) and sequential indicator simulation, particularly the BlockSIS program (Deutsch 2006).

For Direct Sampling simulation, the current deterministic geological model of Escondida Norte is used as training image. For both, Direct Sampling and BlockSIS the available drill-hole information is used for the conditioning of the simulated models.

The results validation tool is finally used to compute the quality and completeness of patterns reproduction of the resulting models.

4.3.1. The Escondida Norte Mine

Minera Escondida is the largest copper mining operation in the world reaching an annual production of 1 086 700 [T] in 2010. The operation is located in northern Chile, 170 [km] southeast of the city of Antofagasta at an altitude of 3 100 [m] above mean sea level. Operations began in 1990 and since 2005 the company operates a second open pit, Escondida Norte, located 5 [km] from the main pit.

The sulphide ore extracted from both mines is sent to one of the concentrator plants (Laguna Seca or Los Colorados), while oxide and low grade sulphide ores are processed in an electro winning plant to produce copper cathodes. Two pipelines transport the copper concentrate to the Coloso Port located 15 km south of the center of Antofagasta, where it is filtered and exported. The cathodes are transported by truck to the port of Antofagasta for subsequent export.

4.3.2. Available information

The available information corresponds to three different data bases: The current available deterministic geological model of Escondida Norte, which is used as training image for Direct Sampling simulations, the drill-hole information, which is used as conditioning data and the topography of the Escondida Norte deposit.

Lithologies of a reduced area of the deposit, which corresponds mainly to the location of the open pit, are modeled (see Figure 46). The deterministic geological model corresponds to a block model with blocks of 25 x 25 x 15 meters in size. The simulated area is located between 2675 and 3305 meters above mean sea level. The characteristics of the location and size of the model are detailed in Table 5.

Direction	Origin Coordinate	Block Size [m]	Number of blocks
X	17500.0	25	90
Y	113050.0	25	65
Z	2682.5	15	42
		Total	225176

Table 5: Spatial characteristics of the simulation grid

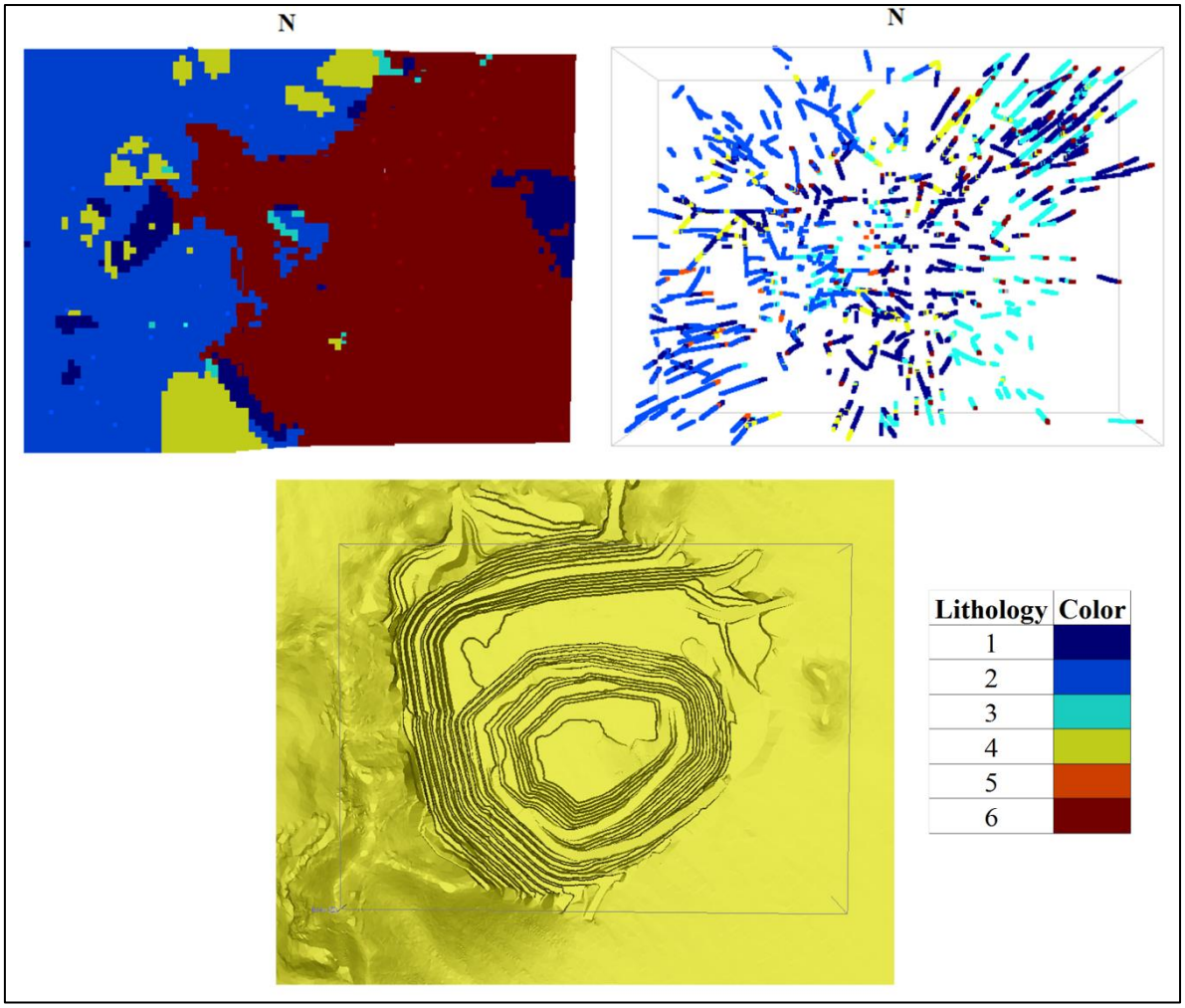

Figure 46: Training image (upper left), conditioning data & simulation area (upper right) and simulation area within the Escondida Norte open pit area (lower)

The X coordinate corresponds to the east-west direction while the Y coordinate corresponds to the north-south direction. The total number of blocks shown in Table 5 corresponds exclusively to blocks located under the topography. Lithologies codification used for this case study is summarized in Table 6.

Lithology	Code
Feldespar Porphyry	1
Rhyolitic Porphyry	2
Andesite	3
Breccias	4
Granitic Porphyry	5
Gravels	6

Table 6: Lithologies codification

In addition to the geological model there are approximately 1780 drill-holes available in the Escondida Norte area with about 90 [%] of them contained inside the simulation area. Drill-hole data are composited at 5 [m] length.

4.3.3. Exploratory data analysis

Due to the wealth of the available drill-hole information, firstly, it is necessary to carry out a selection of the samples that are used as conditioning data (for both, DS and BlockSIS simulations) and for the inference of experimental indicator variograms (only for BlockSIS simulation). This selection is performed in order to adjust the case study to a more realistic scenario (the amount of drill-hole information in this deposit is particularly high) and is done by drill-hole, not by composite, which means that whole drill-holes are left aside, not single samples. After selection, only a 30 [%] of the drill-holes were used for conditioning and inference purposes (see Table 7). However, it should be reminded that the deterministic geological model was constructed considering all the information available.

Conditioning drill-holes	Conditioning samples	Dropped drill-holes	Dropped samples
30 [%]	26926	70 [%]	62543

Table 7: Selection of conditioning data

Drill-holes are not equally distributed in space, as some data clustering can be observed in the center of the simulation area (the open pit area, see Figure 46). A data declustering is performed using the polygonal declustering method to obtain a representative distribution of lithologies. The declustered proportions are used as an input for BlockSIS. The original and declustered global lithologies proportions are shown in Table 8.

Lithology	Proportion (P)	Declustered proportion (P')	P' x (1 – P')
1	0.3973	0.3594	0.2395
2	0.2809	0.2524	0.2020
3	0.1994	0.2396	0.1596
4	0.0811	0.0688	0.0745
5	0.0234	0.0612	0.0229
6	0.0179	0.0186	0.0176

Table 8: Original and declustered proportions

4.3.4. Variographic analysis

The BlockSIS simulation algorithm requires variographic models that define the 2-point spatial continuity of lithologies. Therefore, experimental indicator variograms are computed from the selected scattered drill-hole data. No significant anisotropies are found beside the differences between the vertical and horizontal directions. Therefore, experimental variograms are computed in omni-horizontal and vertical directions (green and blue lines respectively in Figure 47).

Variogram models are adjusted to the computed experimental variograms, focusing on a good adjustment to experimental variograms at short scales, and that variogram models reach the theoretical sill, which depends of the proportion of the lithologies categories (see 4th column of Table 8). The resulting models are summarized in Table 9 and the adjustment to experimental variograms is shown in Figure 47.

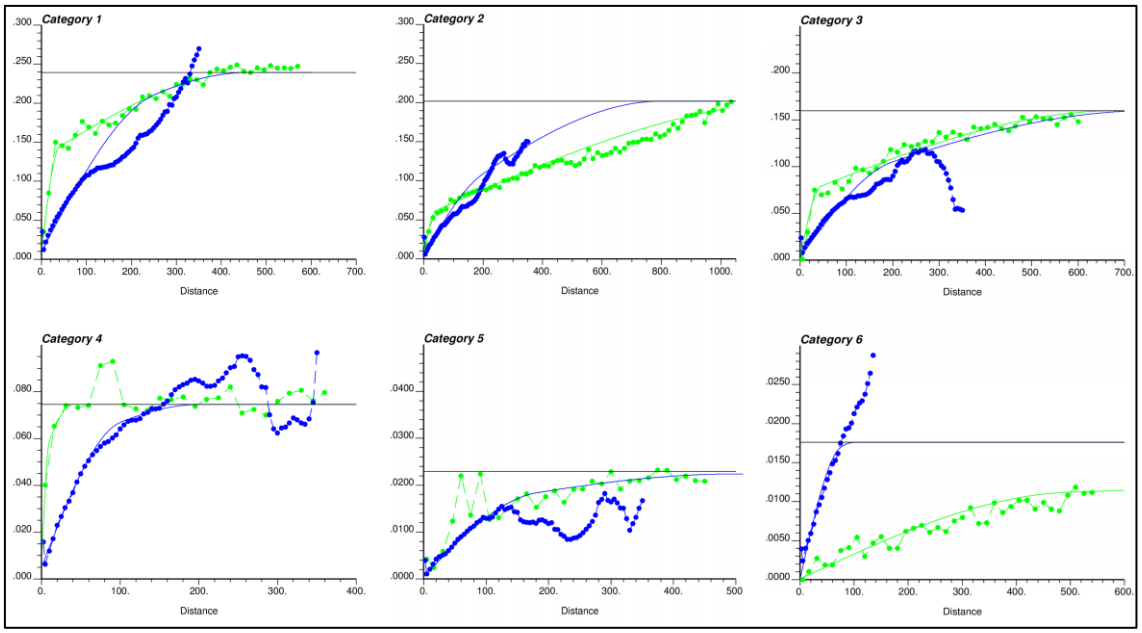

Figure 47: Experimental indicator variograms and variogram models

Category	Direction	Azimuth/Dip	Nugget effect	Spherical		Spherical	
				Sill	Range	Sill	Range
1	DIR 1	omni-horizontal	0.01	0.12	40	0.1095	460
	DIR 2				40		460
	DIR 3	0°/-90°			250		460
Category	**Direction**	**Azimuth/Dip**	**Nugget effect**	**Spherical**		**Spherical**	
				Sill	**Range**	**Sill**	**Range**
2	DIR 1	omni-horizontal	0.005	0.05	40	0.147	1300
	DIR 2				40		1300
	DIR 3	0°/-90°			200		800
Category	**Direction**	**Azimuth/Dip**	**Nugget effect**	**Spherical**		**Spherical**	
				Sill	**Range**	**Sill**	**Range**
3	DIR 1	omni-horizontal	0.003	0.067	45	0.0896	680
	DIR 2				45		680
	DIR 3	0°/-90°			200		750
Category	**Direction**	**Azimuth/Dip**	**Nugget effect**	**Spherical**		**Spherical**	
				Sill	**Range**	**Sill**	**Range**
4	DIR 1	omni-horizontal	0.005	0.045	10	0.0245	40
	DIR 2				10		40
	DIR 3	0°/-90°			100		200
Category	**Direction**	**Azimuth/Dip**	**Nugget effect**	**Spherical**		**Spherical**	
				Sill	**Range**	**Sill**	**Range**
5	DIR 1	omni-horizontal	0.0005	0.0135	180	0.0084	500
	DIR 2				180		500
	DIR 3	0°/-90°			180		500
Category	**Direction**	**Azimuth/Dip**	**Nugget effect**	**Spherical**		**Spherical**	
				Sill	**Range**	**Sill**	**Range**
6	DIR 1	omni-horizontal	0.0001	0.01	500	0.0075	5000
	DIR 2				500		5000
	DIR 3	0°/-90°			75		100

Table 9: Parameters of variogram models

It is important to note that these variogram models are only used as an input for sequential indicator simulation, as the Direct Sampling simulation algorithm infers the multiple-point spatial structure from the training image.

4.3.5. Simulation

Once variogram models are completed, simulations are performed. Three different simulations are considered:

1) Sequential indicator simulation (BlockSIS).
2) Direct Sampling simulation with a high distance threshold (see section 2.5.5).
3) Direct Sampling simulation with a low distance threshold.

For each one 20 stochastic realizations are constructed. The parameters used for BlockSIS and DS simulations are summarized in Table 10 and Table 11, respectively.

Maximum Original Data	16
Maximum previously simulated nodes	24
Search radius (x,y)	500 [m]
Search radius (z)	150 [m]

Table 10: BlockSIS simulation parameters

Parameter	**High-threshold**	**Low-threshold**
Distance threshold	0.2	0
Maximum number of points in neighborhood	30	30
Search radius (x,y)	20 [nodes]	20 [nodes]
Search radius (z)	10 [nodes]	10 [nodes]
Checked TI fraction	1.0	1.0

Table 11: DS simulation parameters

A randomly picked realization of each one of the three different simulations performed is shown in Figure 48.

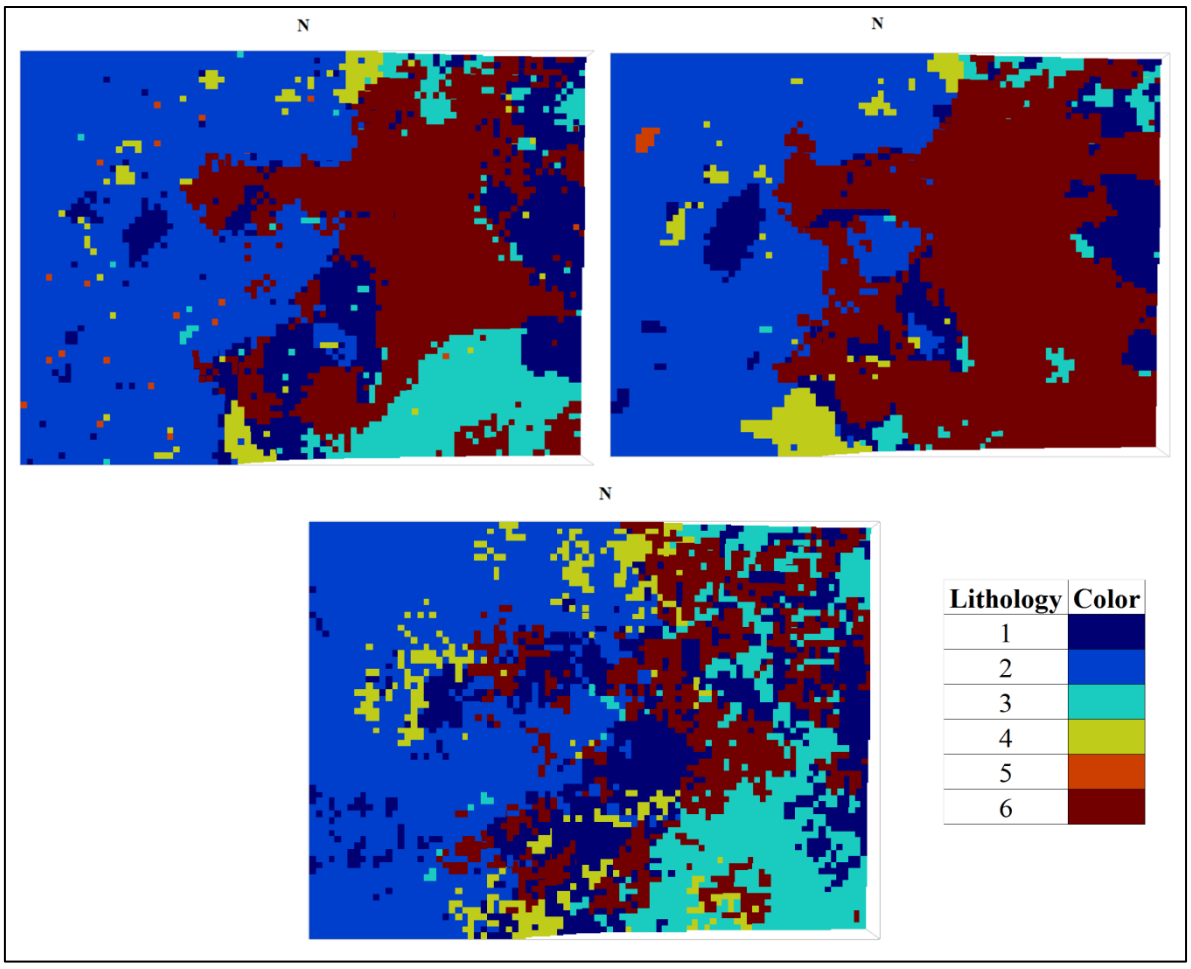

Figure 48: Simulated models: DS with high distance threshold (upper-left), DS with low distance threshold (upper right) and BlockSIS (lower)

4.3.6. Pattern reproduction validation

Finally, a results validation is performed in order to validate the developed tool. The three different simulations where constructed with this purpose, as the patterns reproduction quality should be very different in each case: Direct Sampling simulation with a low distance threshold should reproduce correctly the patterns observed in the training image, a higher distance threshold (0.2 can be considered a high threshold) should harm the quality of the patterns reproduction and sequential indicator simulation does not infer high order spatial statistics, which means that pattern reproduction is not explicitly imposed and should not be expected.

The parameters used to perform the patterns reproduction validation are summarized in Table 12. A low training image fraction is analyzed in the search of a match due to the high CPU times involved in this process.

Parameter	Value
Search radius (x,y)	20 [nodes]
Search radius (z)	10 [nodes]
Minimum event order to analyze	15
Maximum event order to analyze	30
Tolerance (*t*)	0.0
Analyzed fraction of the grids	0.1

Table 12: Validation parameters

The summary of the validations results is shown in Table 13. The expected behavior can be observed: the proportion of training image patterns contained in the results is lowest when sequential indicator simulation is performed, and improves when Direct Sampling simulation is applied, especially when the distance threshold decreases. The same occurs with the proportion of results patterns that are found in the training image. It is important to note that the obtained results look robust even with the low grid fraction checked (see Table 12). With higher fractions the results should become more robust and stable as the probability of finding complex patterns would increase, favoring models with a better pattern reproduction.

This case study also verifies the applicability of the method for validation of multiple-point simulation results of a real 3-D mine-scale reconstruction of a categorical variable. Considering results obtained in case study 3, it serves to verify that both developed algorithms, for training image selection and results validation, can be used to overcome some limitations and enhance the application of multiple-point simulation algorithms to hydrology and mining related models.

Realization	Results patterns in TI			TI patterns in results		
	BlockSIS	DS - Threshold = 0.2	DS- Threshold = 0	BlockSIS	DS - Threshold = 0.2	DS- Threshold = 0
1	0.337	0.582	0.616	0.473	0.509	0.535
2	0.249	0.562	0.664	0.481	0.512	0.542
3	0.236	0.569	0.647	0.481	0.525	0.560
4	0.335	0.532	0.598	0.487	0.519	0.566
5	0.205	0.558	0.609	0.481	0.513	0.571
6	0.244	0.565	0.562	0.485	0.524	0.564
7	0.291	0.551	0.586	0.465	0.513	0.580
8	0.309	0.501	0.629	0.473	0.517	0.546
9	0.279	0.552	0.633	0.480	0.517	0.560
10	0.303	0.572	0.591	0.478	0.524	0.532
11	0.267	0.548	0.621	0.481	0.515	0.571
12	0.286	0.553	0.468	0.477	0.513	0.587
13	0.393	0.598	0.644	0.479	0.509	0.561
14	0.354	0.529	0.627	0.472	0.514	0.562
15	0.363	0.565	0.580	0.491	0.514	0.577
16	0.349	0.564	0.571	0.482	0.520	0.605
17	0.302	0.564	0.609	0.472	0.517	0.560
18	0.235	0.582	0.659	0.477	0.512	0.532
19	0.338	0.496	0.576	0.471	0.521	0.597
20	0.306	0.580	0.634	0.485	0.513	0.597
Mean	**0.299**	**0.556**	**0.606**	**0.479**	**0.516**	**0.565**
Minimum	0.205	0.496	0.468	0.465	0.509	0.532
Maximum	0.393	0.598	0.664	0.491	0.525	0.605

Table 13: Results of validation

4.3.7. Training image selection tool test

Finally, once models are constructed, 20 different data sets are built in order to test the training image selection tool in a real case study. For each realization, 3 different models are selected as training images to build a training image set: The SIS model, the DS model with high threshold and the DS model with low threshold. The training image selection tool is used to select the most compatible model to the data in terms of high order spatial structure (events of 25^{th} order are considered, the other parameters (radii, checked fraction, threshold) remain unchanged from those shown in Table 12). Results are summarized in Table 14.

Realization	Relative compatibility		
	SIS	DS - Threshold = 0.2	DS - Threshold = 0
1	0.291	0.294	0.415
2	0.283	0.294	0.423
3	0.278	0.301	0.421
4	0.278	0.294	0.428
5	0.279	0.302	0.419
6	0.286	0.305	0.409
7	0.283	0.291	0.426
8	0.293	0.298	0.409
9	0.295	0.292	0.413
10	0.292	0.298	0.410
11	0.291	0.294	0.415
12	0.287	0.305	0.408
13	0.297	0.288	0.415
14	0.297	0.288	0.415
15	0.304	0.290	0.406
16	0.309	0.290	0.401
17	0.299	0.301	0.400
18	0.295	0.295	0.410
19	0.290	0.293	0.417
20	0.282	0.299	0.419
Mean	0.290	0.296	0.414
Minimum	0.278	0.288	0.400
Maximum	0.309	0.305	0.428

Table 14: Training image selection tool test results, Escondida Norte case study

Results show the expected behavior. A better high order spatial compatibility between the data and the low-threshold DS models is obtained, while the lowest compatibility is generally obtained when compatibility between data and SIS models is assessed. It is important to remark that this case study corresponds to a realistic application of the tool, considering the number of rock types, the size of the models, and that each one of them honors the conditioning data at the specific informed locations.

5. CONCLUSIONS

5.1. Training image selection

In this work, two criteria are proposed (relative and absolute compatibility) to compare a data set with a set of candidate training images. The advantage of this comparison approaches is that it simultaneously considers several orders of complexity.

The relative compatibility measure is especially useful as a training image selection tool in cases where more than a single conceptual model is available as training image. However, it does not allow assessing if all training images are incompatible with the available data. This is palliated by the absolute comparison criterion.

For both relative and absolute comparison criteria, two calculation approaches are tested, exhaustive scanning and direct sampling. Both show similar results, but considerably lower computation times are obtained when using the direct sampling algorithm, especially when lower order events compatibility is assessed. These computation times represent also an interesting research area as they could be used as an absolute training image quality assessment criterion. Better training images should contain numerous repetitions of patterns that appear often in the data, reducing search times of matching events.

The proposed training image selection methods are successfully tested in four case studies. The aquifer case study showed that the methods can be used also as an aid to geological interpretation since it allowed identifying the origin of the data taken at different depths in a vertically non-stationary 3D volume. It also demonstrates the applicability of the method to assess training image spatial compatibility with data in real cases, which was confirmed with the Escondida Norte case study. In addition to offering the ability of selecting training images, the method could also be used to assess the validity of the spatial parameterization in other contexts. For example, in the context of object-based models, it could help selecting different sets of rules for the placement of objects just as it was able to select the best simulation algorithm and parameters in the mining case study.

The absolute metric is presented in a single way (data patterns occurrences are searched in training images), but could also be done looking at the reverse. If every data event is found in the training image, this does not mean that every training image pattern is found in the data. Looking at the reverse problem represents another interesting future research direction, as measure of compatibility would be mixed with a measure of completeness of data as a characterization of the structural patterns found in the training image, allowing to assess the problem in a more complete way. If data are scarce and

structural patterns complex, this index would be low. However, it should be reminded that a training image aims at adding information that is not available in the data, thus its goal is to complete the information available in the scattered data.

5.2. Simulation results validation

Regarding the results validation tool, its applicability is successfully verified in a real mine-scale case study. It allowed obtaining the expected results even with the low grid check fraction used due to the high demanded CPU times. As in the case of the compatibility tool, the main advantage of the proposed method is the possibility to check the quality of reproduction of patterns contained in the training image of several orders. However it should be noted that even though the method represents an improvement regarding multiple-point simulations validation, the reproduction of only a small, randomly sampled, fraction of the patterns contained in the training image is verified. Therefore, the convergence of results is not assured, especially in small (small training image or simulation grid) and complex (high number of facies, big scale spatial continuities) case studies.

Differently to the compatibility tool, the results validation algorithm computes both, the proportion of simulated values contained in the training image, measure that gives an idea of the quality of the pattern reproduction, and the proportion of training image patterns that are effectively reproduced, measure that permits verifying the completeness of the reproduction. It should be reminded that an incomplete or poor pattern reproduction can be caused by different reasons which are not recognizable solely by this method. Among those, incompatibility between training image and data, training image-data scale issues, non-stationarity (differences of facies proportions, geo-bodies sizes and orientations, etc., across the training image or conditioning data), inadequate definition of search templates (regarding size and orientation) to reproduce certain geometries, significant (and not corrected for example by declustering, Servosystem factor, etc.) differences between facies proportions observed in training image and data.

Both algorithms are programmed in Fortran programming language and are implemented as GSLIB executable files. It is recommended to improve the computing efficiency of both algorithms to enhance their application to bigger case studies.

The relevance of this work in mining applications is that it provides new tools to ease the use of multiple-point simulation techniques which should allow obtaining better representations of the spatial distribution of geological attributes, leading to more realistic models.

6. REFERENCES

- ALABERT, F., 1987: *Stochastic Imaging of Spatial Distributions Using Hard and Soft Information.* Master's Thesis, Stanford University, Department of Applied Earth Sciences, 197 p.

- ALCOLEA, A., RENARD, P., MARIETHOZ, G., BRETONE, F., 2009: *Reducing the impact of a desalination plant using stochastic modeling and optimization techniques.* J. Hydrol., 365(3-4), 275-288, doi:10.1016/j.jhydrol.2008.11.034.

- ARMSTRONG, M., GALLI, A., LE LOC´H, G., GEFFROY, F., ESCHARD, R., 2003: *Plurigaussian Simulations in Geosciences.* Springer, Berlin, 160 p.

- ARPAT, B., AND CAERS J., 2007: *Conditional simulations with patterns.* Mathematical Geology, 39(2), 177-203.

- BOISVERT, J. B., PYRCZ, M. J., DEUTSCH, C. V., 2007: *Multiple-point statistics for training image selection.* Natural Resources Research 16(4): 313-321.

- BOUCHER, A., GUPTA, R., CAERS, J., SATIJA, A., 2010: *Tetris: a training image generator for SGeMS,* in Proceedings, Proceedings of 23rd SCRF annual affiliates meeting, Stanford University, May 5-6 2010.

- CAERS, J., 2005: *Petroleum Geostatistics,* 88 p., Soc. of Petroleum Eng., Richardson.

- CARLE, S. F., FOGG, G. E., 1996: *Transition probability-based indicator geostatistics.* Mathematical Geology, 28(4): 453-476.

- CHUGUNOVA, T., HU, L., 2008: *Multiple-point simulations constrained by continuous auxiliary data.* Mathematical geosciences, 2008, vol. 40, no 2, pp. 133-146.

- DEUTSCH, C. V., 1992: *Annealing techniques applied to reservoir modeling and the integration of geological an engineering (well test) data.* PhD dissertation, Stanford University.

- DEUTSCH, C. V., JOURNEL, A. G., 1992: *GSLIB: Geostatistical Software Library and user's guide.* Oxford University Press, New York, 340 p.

- DEUTSCH, C. V., 2006: *A sequential indicator simulation program for categorical variables with point and block data: BlockSIS.* Computers and Geosciences vol. 32, pp. 1669-1681.

- DE IACO, S., MAGGIO, S., 2011: *Validation techniques for geological patterns simulations based on variogram and multiple-point statistics.* Mathematical Geosciences vol. 43, pp. 483-500.

- DIMITRAKOPOULOS, R., DAGBERT, M., 1993: *Sequential modeling of relative indicator variables. Dealing with multiple lithology types,* in: Soares A. (ed.) Geostatistics Tróia '92, 4th international Geostatistics Congress, Dordrecht: Kluwer Academic, vol. 1, pp. 413-424.

- DIMITRAKOPOULOS, R., MUSTAPHA, H., GLOAGUEN, E., 2010: *High-order statistics of spatial random fields: Exploring spatial cumulants for modeling non-Gaussian and non-linear phenomena.* Mathematical Geosciences vol. 41, pp. 679-701.

- EMERY, X., ORTIZ, J. M., 2007: *Weighted sample variograms as a tool to better assess the spatial variability of soil properties.* Geoderma 140, pp. 81-89.

- EMERY, X., 2008: *Simulación estocástica y geoestadística no lineal,* Universidad de Chile, Facultad de Ciencias Físicas y Matemáticas, Departamento de Ingeniería de Minas, Santiago, 189 p.

- EMERY, X., 2010: *Apuntes del curso MI75D, Tópicos Avanzados en Evaluación de Yacimientos,* Universidad de Chile, Facultad de Ciencias Físicas y Matemáticas, Departamento de Ingeniería de Minas, Santiago.

- ESKANDARIDALVAND, K., 2008: *Growthsim: A complete framework for integrating static and dynamic data into reservoir models.* Ph. D. Dissertation, University of Texas at Austin 2008.

- GALLI, A., BEUCHER, H., LE LOC´H, G., DOLIGEZ, B., HERESIM GROUP, 1994: *The Pros and Cons of the Truncated Gaussian Method.* In Armstrong, M. and Dowd, P.A. (eds.), Geostatistical Simulations. Kluwer Academic, Dordrecht, pp. 217-233.

- GLOAGUEN, E., DIMITRAKOPOULOS, R., 2009: *Two-dimensional conditional simulations based on the wavelet decomposition of training images.* Mathematical Geosciences, v. 41, pp. 679-701.

- GOOVAERTS, P., 2000: *Geostatistical approaches for incorporating elevation into spatial interpolation of rainfall.* J. Hydrol. Amsterdam, 228, 113-129, doi:10.1016/S0022-1694(00)00144-X.

- GUARDIANO, F., SRIVASTAVA, M., 1993: *Multivariate geostatistics: Beyond bivariate moments,* in: Soares A. (ed.) Geostatistics Tróia '92, 4th international Geostatistics Congress, Dordrecht: Kluwer Academic, vol. 1, pp. 133-144.

- HONARKHAH, M., CAERS, J., 2010: *Stochastic Simulation of Patterns Using Distance-Based Pattern Modeling.* Mathematical Geosciences, 42: 487-517.

- HURTADO, S., 2009: *Simulación de variables categóricas considerando estadísticas de patrones.* Memoria para optar al título de ingeniero civil de minas, Universidad de Chile, Facultad de Ciencias Físicas y Matemáticas, Departamento de Ingeniería de Minas, Santiago, 61 p.

- ISAAKS, E. H., SRIVASTAVA, M., 1989: *An introduction to applied geostatistics.* Oxford University Press, New York, USA.

- JOURNEL, A., ZHANG, T., 2006: *The necessity of a multiple-point prior model.* Mathematical Geology 38/5): 591-610.

- JOURNEL, A. G., HUIJBREGTS, C. J., 1978: *Mining Geostatistics.* Academic Press.

- KRISHNAN, S., JOURNEL, A. G., 2003: *Spatial connectivity: From variograms to multiple-point measures.* Mathematical Geology, v. 35, pp. 915-925.

- KYRIAKIDIS, P. C., KIM, J., MILLER, N., 2001: *Geostatistical mapping of precipitation from rain gauge data using atmospheric and terrain characteristics.* Journal of Applied Meteorology 40: 1855-1877.

- LOPEZ, S., 2003: *Modélisation de Réservoirs Chenalisés Méandriformes, Approche Génétique et Stochastique.* Paris School of Mines.

- MAHARAJA, A., 2008: *TiGenerator: object-based training image generator.* Computers & Geosciences 34(12): 1753-1761.

- MARIETHOZ, G., RENARD, P., CORNATON, F., JAQUET, O., 2009: *Truncated plurigaussian simulations to characterize aquifer heterogeneity.* Ground Water, 47 (1), pp. 13-24.

- MARIETHOZ, G., RENARD, P., STRAUBHAAR, J., 2010: *The direct sampling method to perform multiple-point geostatistical simulations.* Water Resources Research 46(11). W11536.

- MARIETHOZ, G., KELLY, B. F. J., BAKER, A., 2012: *Quantifying the value of laminated stalagmites for paleoclimatic reconstructions.* Geophysical Research Letters, vol. 39, L05407, 6 p., 2012. doi:10.1029/2012GL050986.

- MATHERON, G., BEUCHER, H., DE FOUQUET, C., GALLI, A., GUÉRILLOT, D., RAVENNE, C., 1987: *Conditional Simulation of the Geometry of Fluvio Deltaic Reservoirs.* In: 62nd Annual Technical Conference and Exhibition of the Society of Petroleum Engineers, Dallas, Texas, SPE 16753, pp. 123-131.

- MIROWSKI, P., TETZLAFF, D., DAVIES, R., McCORMICK, D., WILLIAMS, N., SIGNER, C., 2009: *Stationary Scores on Training Images for Multipoint Geostatistics.* Mathematical Geosciences, vol. 41, pp. 447-474.

- ORTIZ, J. M., 2003: *Characterization of High Order Correlation for Enhanced Indicator Simulation.* PhD thesis, University of Alberta, Edmonton, AB, Canada, 2003.

- ORTIZ, J. M., DEUTSCH C. V., 2004: *Indicator simulation accounting for multiple-point statistics.* Mathematical Geology 36(5): 545-565.

- ORTIZ, J. M., EMERY, X., 2006: *Geostatistical estimation of mineral resources with soft boundaries: a comparative study.* Journal of the South African Institute of Mining and Metallurgy, 106(8): 577-584.

- ORTIZ, J.M., LYSTER, S., DEUTSCH, C. V., 2007: *Scaling multiple-point statistics to univariate proportions.* Computers and Geosciences 33(2): 191-201.

- ORTIZ, J. M., 2008: *An overview of the challenges of multiple point geostatistics.* Geostats 2008 – Proceedings of the Eight International Geostatistics Congress, J.M. Ortiz and X. Emery (eds.), Quebecor World Chile, Santiago, Chile, vol. 1, pp. 11-20.

- ORTIZ, J.M., MAGRI, M., LIBANO, R., 2012: *Improving financial returns from mining through geostatistical simulation and the optimized advance drilling grid at El Tesoro Copper Mine.* Journal of the South African Institute of Mining and Metallurgy, 112(1): 15-22.

- PARRA, A., ORTIZ, J. M., 2009: *Conditional multiple-point simulation with a texture synthesis algorithm.* Computational Methods for the Earth, Energy and Environmental Sciences IAMG 2009, August 23-28, Stanford University, USA, Electronic Conference Proceedings (ISBN # 978-0-615-33449-3).

- PARRA, A., ORTIZ, J. M., 2011: *Adapting a texture synthesis algorithm for conditional multiple point geostatistical simulation.* Stochastic Environmental Research and Risk Assessment, 25 (8), pp. 1101-1111.

- PEREDO, O., ORTIZ, J. M., 2011: *Parallel implementation of simulated annealing to reproduce multiple-point statistics.* Computers & Geosciences, 37(8): 1110-1121.

- PYRCZ, M., CATUNEANU, O., DEUTSCH, C. V., 2005: *Stochastic surface-based modeling of turbidite lobes.* AAPG bulletin 89 (2), pp. 177-191.

- SOLENG, H.H., SYVERSVEEN, A.R., KOLBJØRNSEN, O., 2006: *Comparing facies realizations – Defining matrices on realization space.* In: Ecmor X, Proceedings of the 10th European conference in the mathematics of oil recovery, p.A014.

- STRAUBHAAR, J., WALGENWITZ, A., RENARD, P., FROIDEVAUX, R., 2008: *Optimization issues in 3D multipoint statistics simulation.* Geostats 2008 – Proceedings of the Eight International Geostatistics Congress, J. M. Ortiz and X. Emery (eds.), Quebecor World Chile, Santiago, Chile, vol. 1, pp. 1035-1045.

- STRAUBHAAR, J., RENARD, P., MARIETHOZ, G., FROIDEVAUX, R., BESSON, O., 2011: *An improved parallel multiple-point algorithm using a list approach.* Mathematical Geosciences 43 (3), pp. 305-328.

- STREBELLE, S., 2000: *Sequential simulation drawing structure from training images.* PhD Thesis, Stanford University, Stanford, CA, USA.

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

- TAHMASEBI, P., SAHIMI, M., MARIETHOZ, G., HEZARKHANI, A., 2012: *Accelerating geostatistical simulations using graphics processing units (GPU).* Computers and Geosciences, vol. 46, September 2012, pp.51-59.

- YATES, E., CREUTIN, J. D., ANQUETIN, S., RIVOIRARD, J., 2007: *A Scale-Dependent Quality Index of Areal Rainfall Prediction.* J. Hydrometeor, 8, 160-170. doi:10.1175/JHM563.1

- ZHANG, T., SWITZER, P., JOURNEL, A. G., 2006: *Filter-based classification of training image patterns for spatial simulation.* Mathematical Geology 38(1): 63-80.

- ZHANG, X., PYRCZ, M., DEUTSCH, C. V., 2009: *Stochastic surface modeling of deepwater depositional systems for improved reservoir models.* Journal of Petroleum Science and Engineering, 68(1-2): 118-134.

- ZHANG, X., JIANG, H., ZHOU, G., XIAO, Z., ZHANG, Z., 2012: *Geostatistical interpolation of missing data and downscaling of spatial resolution for remotely sensed atmospheric methane column concentrations.* International Journal of Remote Sensing, vol. 33, issue 1, pp. 120-134.

- ZIMMERMAN, D. A., de MARSILY, G., GOTAWAY, G. A., MARIETTA, M. G., AXNESS, C. L., BEAUHEIM, R., BRAS, R., CARRERA, J., DAGAN, G., DAVIES, P. B., GALLEGOS, D., GALLI, A., GRINDROD, P., GUTJAHR, A. L., KITANIDIS, P., LAVENEU, A. M., McLAUGHLING, D., NEUMAN S. P., RAMARAO, B. S., RAVENNE, C., RUBIN, Y., 1998: *A comparison of seven geostatistically-based inverse approaches to estimate transmissivities for modeling advective transport by groundwater flow.* Water Resources 34(6): 1373-1413.

7. APPENDIX

Appendix A: Relative compatibilities, case study 3

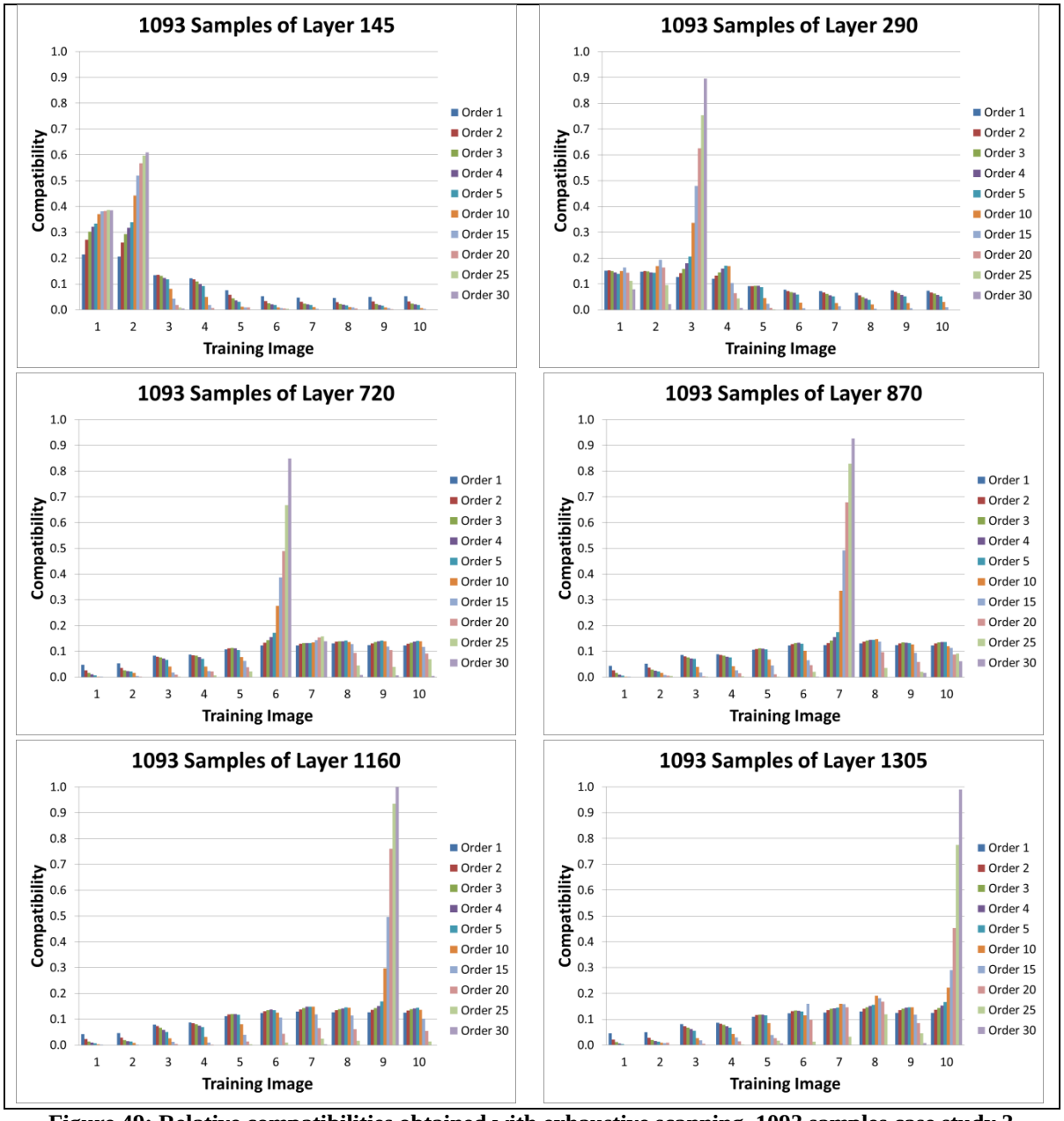

Figure 49: Relative compatibilities obtained with exhaustive scanning, 1093 samples case study 3

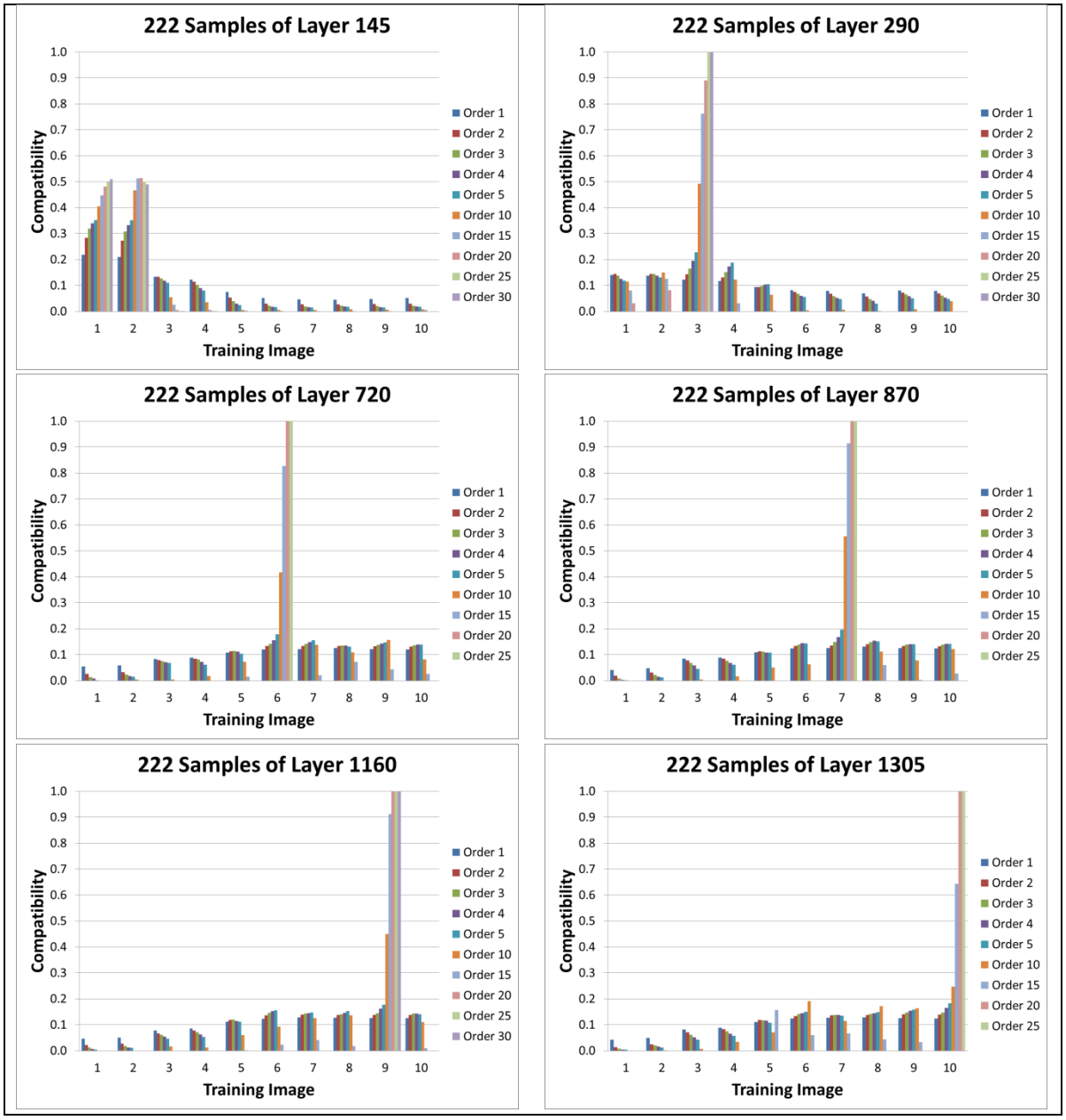

Figure 50: Relative compatibilities obtained exhaustive scanning, 222 samples case study 3

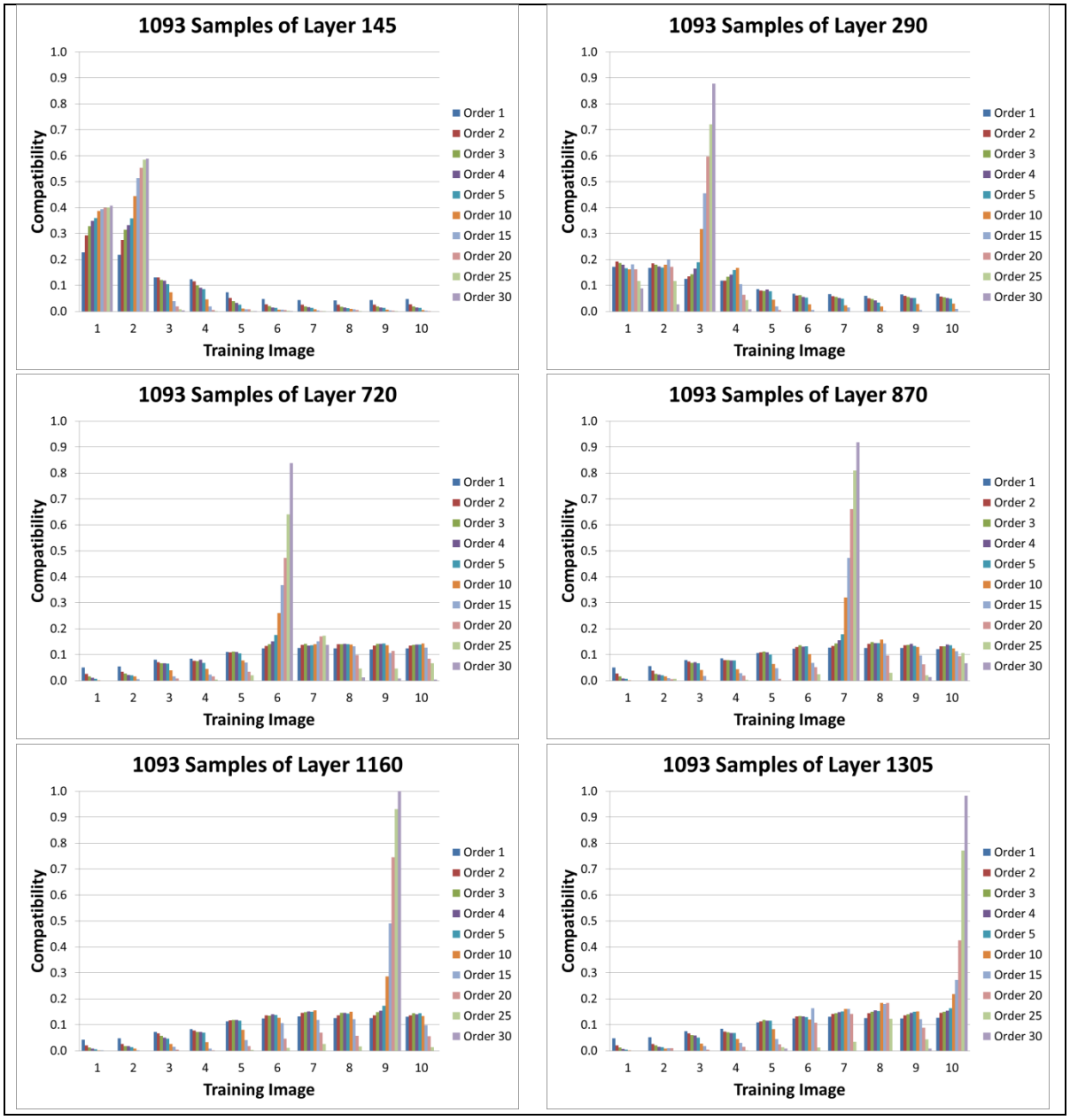

Figure 51: Relative compatibilities obtained with direct sampling, 1093 samples case study 3

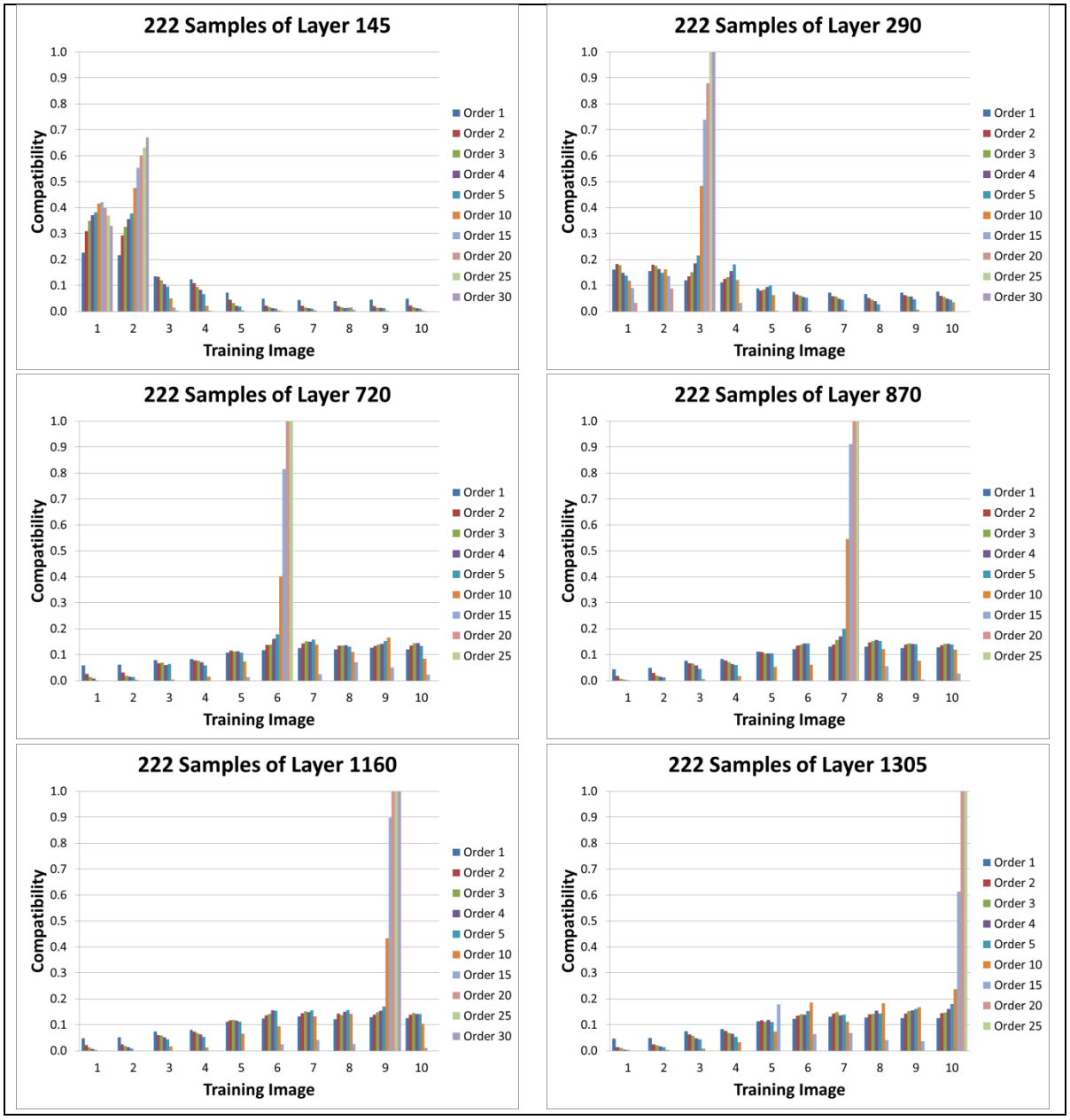

Figure 52: Relative compatibilities obtained with direct sampling, 222 samples case study 3

Appendix B: Absolute compatibilities, case study 3

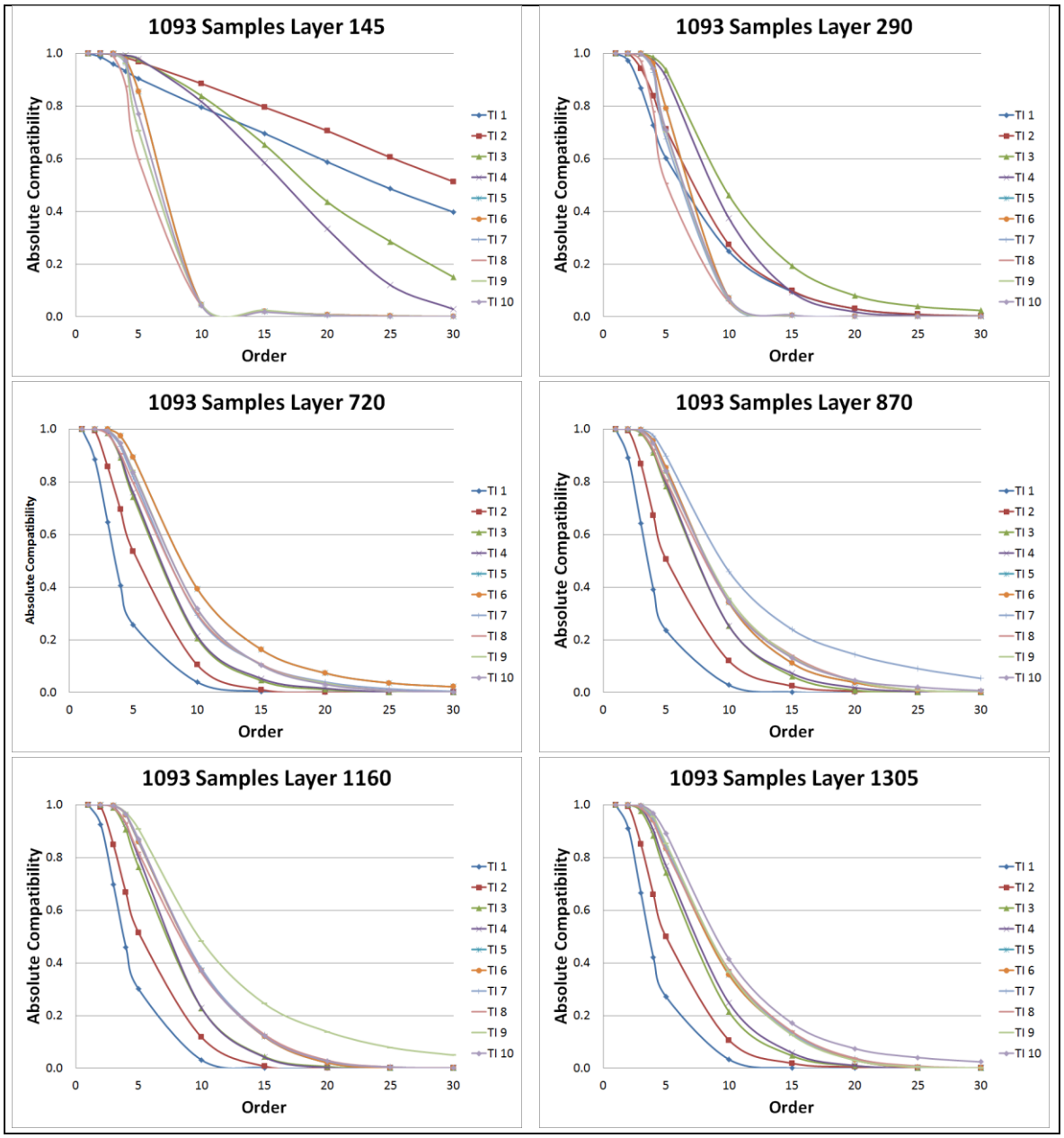

Figure 53: Absolute compatibilities, 1093 samples case study 3

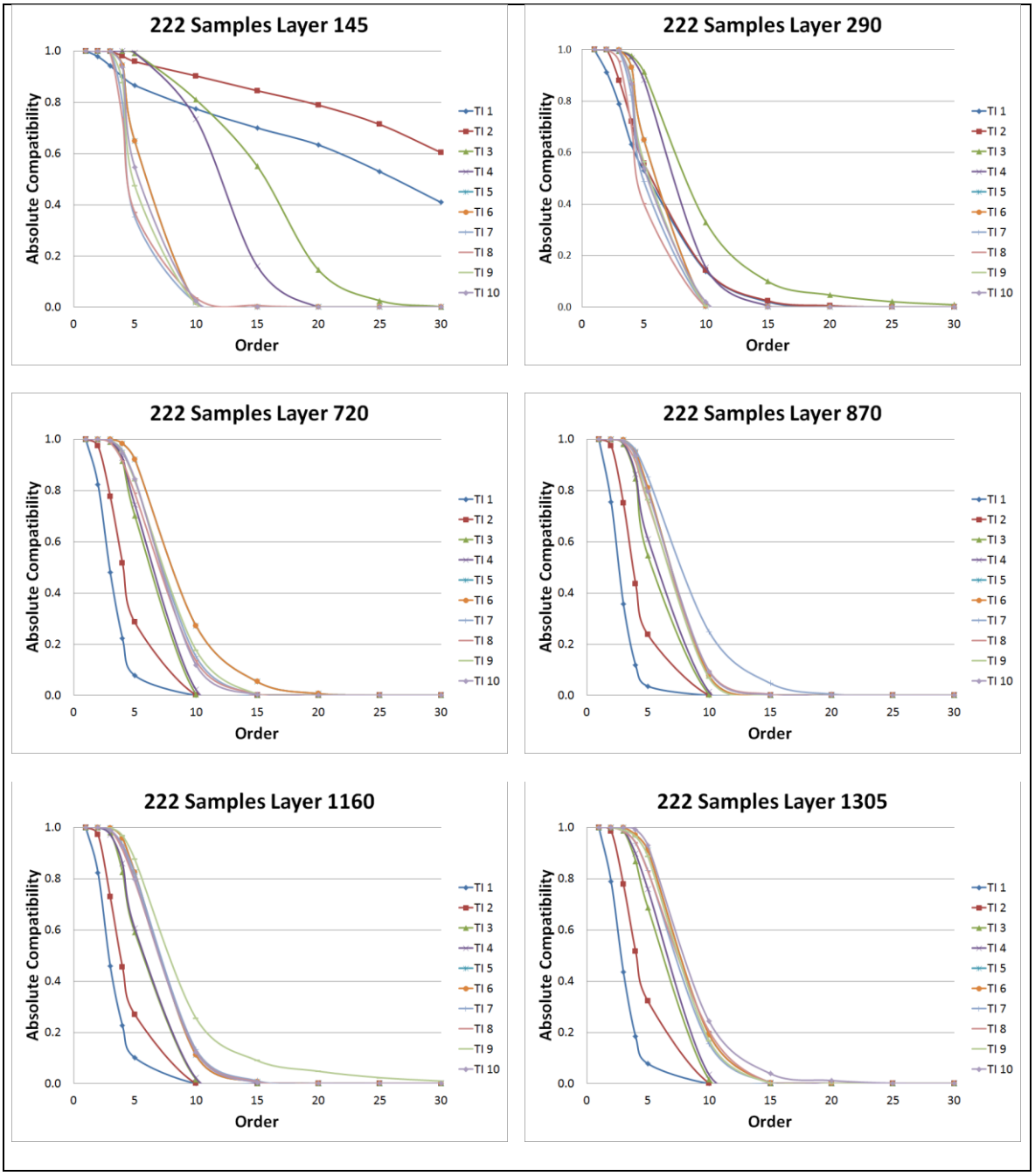

Figure 54: Absolute compatibilities, 222 samples case study 3