Generating a 3D Training Image with Low-Rank Tensor Completion: A Multi-Stage Application

Maiko V. Sell

Degree of Master of Engineering

Department of Mining and Materials Engineering, Faculty of Engineering

McGill University

Montreal, Quebec, Canada

November 20, 2014

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Master of Engineering

DEDICATION

I dedicate this to my mother and to Gabrielle.

ACKNOWLEDGEMENTS

I thank Professor Dimitrakopoulos for the opportunity and financial support (NSERC CRD grant 411270) offered to me. I am equally indebted to Professor Psaromiligkos for his encouragement and technical feedback and to Dr. Machuca-Mory for thorough comments and suggestions. I thank Professors Laura Nilson, In-Ho Jung, and Stephen Yue for helping me through a tough spot.

I am particularly thankful to Gabrielle Ciquier for her editorial revisions, her patience, her unerring devotion, and love; she has been my truest companion in this endeavour. I am grateful to my mother, Maria-Cristina Grossi, who never doubts in me even when I am at my lowest. She is my first teacher. I thank my friend Mark Pelechaty for his support and his comments and my uncle J. Grossi for his meticulous eye.

Suarez Miranda continues to enlighten me. He is as mythical as the stories he tells.

ABSTRACT

A training image (TI) is an important tool in the context of multiple-point (MPS) and high-order (HOS) statistic spatial simulations. TIs represent the repository of patterns and spatial statistics required for drawing values at each grid node in a multiple-point spatial simulation. In turn, spatial simulations provide the framework for the construction of orebody and petroleum reservoir risk and characterization models. Methods of constructing 3D continuous data TIs from adversarial sampling schemes (i.e. non-random) that respect the original sample statistics are important yet uncommon in the literature. Methods for image reconstruction, such as Multi-Stage Matrix Completion (MSMC), work on 2D and with randomly located samples. In the present thesis, drillhole samples of continuous porosity values representing 2% of the exhaustive dataset were used to reconstruct a 3D continuous TI directly from strings of non-randomly located samples using a new method called Multi-Stage Tensor Completion (MSTC). The methodology employs the Low-Rank Tensor Completion (LRTC) algorithm to help reconstruct missing values at the original resolution scale. The MSTC methodology is tested on two benchmark continuous porosity datasets obtained from an exhaustive TI of petroleum reservoir channels (sandstones and clays). The application of MSTC was also applied to an actual mining deposit. Due to confidentiality clauses, only the results obtained from the petroleum reservoir dataset will be discussed in this thesis. In all cases tested, the MSTC method has been found to be effective at reproducing the data statistics. In cases with a very low number of high grade samples, their proportion is not reconstructed as effectively when compared to results obtained using the benchmark cases. The results demonstrate that this method is able to generate continuous data TIs that respect the high order spatial cumulant statistics of the original samples, up to at least the 4th order, all the while maintaining visually reasonable results. The present findings contribute to the development of data-driven methods using MPS and HOS spatial simulation algorithms when modeling continuous attributes of deposits that have complex (nonlinear), non-Gaussian, spatial patterns that are sampled with mining-specific (not random) drill patterns.

RÉSUMÉ

Une image d'entraînement (TI) est un outil fondamental dans le domaine des simulations à points multiples (MPS) et des simulations statistiques spatiales d'ordre élevé (HOS). Les TI représentent une base de données statistiques et de configurations spatiales. Celles-ci sont requises pour réaliser la construction de distributions conditionnelles utilisées afin de générer un point simulé à chaque nœud sur une grille de simulation. À leur tour, les simulations spatiales créent l'architecture sur laquelle est basée la construction des modèles de caractérisation et de risque concernant les réservoirs de pétrole et les gisements de minerais. Cependant, les méthodes 3D utilisées pour construire des TI, basées d'une part sur des données continues et d'autre part sur des données qui n'ont pas été obtenues par des méthodes d'échantillonnage aléatoire, sont très rares dans la littérature. Les méthodes de reconstruction d'image, telles que la méthode de complétion 2D de matrices à multiples étapes (MSMC), s'appliquent à des échantillons situés de façon aléatoire. Ce mémoire propose une méthode de complétion de tenseurs à multiples étapes (MSTC). Cette méthode permet la reconstruction d'une TI en 3D provenant directement de chaines d'échantillons - situées de façon non-aléatoire - et ne représentant que 2% de toute la TI. La méthode MSTC reconstruit des TI de la même résolution que celle des données originales en utilisant un algorithme de complétion de tenseur nommé LRTC. La fiabilité de la méthode MSTC est d'abord testée à partir de deux TI provenant d'un réservoir de pétrole artificiel (composé de grès et d'argile). Cette méthode a également été appliquée à un gisement de minerais réel. En raison de clauses de confidentialité, seul les résultats obtenus à partir de l'ensemble des données artificielles seront présentés dans le cadre de ce mémoire. Dans tous les cas testés, la méthode MSTC est capable de reconstruire les statistiques provenant d'échantillons. Quant aux cas composés d'un faible nombre d'échantillons de haute teneur, les proportions de ces échantillons ne semblent être reconstruites aussi efficacement lorsqu'on les compare avec les résultats obtenus avec les cas de références. Les résultats de cette analyse démontrent que d'une part, la méthode MSTC génère des TIs de données continues qui sont visuellement acceptables, et d'autre part, cette méthode est capable de reconstruire les

statistiques d'ordre élevé, jusqu'au 4^{ième} ordre. Les résultats de ce mémoire contribuent au développement des algorithmes de simulations MPS et HOS qui permettent tous deux de reconstruire des configurations spatiales complexes (non-linéaires) avec des données continues échantillonnées de façon non-aléatoire, tout en utilisant des motifs de forage spécifiques.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

ALM	Augmented Lagrangian Method (optimization algorithm used in matrix completion)
APGL	Accelerated Proximal Gradient with Line Search (optimization algorithm used in matrix completion)
BCD	Block Coordinate Descent (optimization algorithm used in tensor completion)
CCDF	Conditional Cumulative Distribution Function
CVX	A convex optimization Matlab toolbox (optimization algorithm used in matrix completion)
GROUSE	Grassmann Rank-One Update Subspace Estimation (optimization algorithm used in matrix completion)
HOS	High-Order Statistics
HOSIM	High-Order Simulation (HOS simulation algorithm)
LRTC	Low-Rank Tensor Completion (optimization algorithm used in tensor completion)
MPS	Multiple-Point Statistics
MSMC	Multi-Stage Matrix Completion (reconstruction algorithm used with SVT algorithm)
MSTC	Multi-Stage Tensor Completion (tensor reconstruction algorithm used with LRTC algorithm)
RF	Random Function
SSM	Spatial Similarity Method (MPS simulation algorithm)
SET	Subspace Evolution and Transfer (optimization algorithm used in matrix completion)
SNESIM	Single Normal Equation Simulation (MPS simulation algorithm)
SVD	Singular Value Decomposition
SVT	Singular Value Thresholding (optimization algorithm used in matrix completion)
TI	Training Image

CHAPTER 1 INTRODUCTION

A ubiquitous problem encountered in the field of mining engineering is how to effectively use sparse geological datasets to construct orebody models that lead to accurate assessments of their potential to be mined. In reality, economical and physical constraints dictate the fact that an orebody cannot be sampled fully before making the decision to mine it. Mineable orebodies will typically be sampled by repeated exploration, evaluation, and production drilling campaigns that aim to map out its economic value based on local recoverable metal (or mineral) – these samples usually represent less than 1% of the entire orebody. From this data, mineral deposits are represented by "block models" that consist of equally sized volumetric units (blocks) in 3D space. Traditional orebody modeling techniques use estimation methods to gather samples into a single "average type" block model that fails to account for geological uncertainty. It has been shown that ignoring this uncertainty in mining deposits is the major cause of failure or underperformance in mining ventures (Vallee, 2000). In light of this situation, the geostatistical community has developed stochastic spatial simulation methods over the last 40 years (for more information on these methods, see Journel, 1974; Chilès & Delfiner, 1999; David, 1988; Goovaerts, 1997; Journel & Huijbregts, 1978; Remy, Boucher, Wu, 2009; Remy, 2004). Stochastic simulation is the process of generating multiple realizations of the joint distribution of attribute values in space. Conditional spatial simulation methods aim to capture geological heterogeneity by reproducing the data statistics conditional to the observed values; geological heterogeneity is the term used to describe variations in geology, such as mineral grades, rock porosity, grain size, and lithologic texture. When multiple, equally probable spatial simulations are available, it becomes possible to quantify the spatial and rock property uncertainties that are inherent in orebody models. Through access to quantified geological uncertainty obtained using spatial simulation methods, decision-makers are better equipped to avoid highly probable negative financial scenarios and to focus their attention in areas with a higher likelihood of success (Dimitrakopoulos, Farrelly, & Godoy, 2002; Godoy, 2003). In fact, robust stochastic optimization mine planning techniques are being developed to handle geological uncertainty and to

provide mine planners with more practical and profitable mine production schedules over the life of a mine (Benndorf & Dimitrakopoulos, 2013; Consuegra & Dimitrakopoulos, 2010; Dimitrakopoulos, 2011; Godoy, 2003; Goodfellow & Dimitrakopoulos, 2013; Menabde et al., 2004; Montiel & Dimitrakopoulos, 2013; Ramazan & Dimitrakopoulos, 2013). Yet these latest stochastic optimization techniques still depend on the quality of the input orebody models being considered. Traditionally in geostatistics, sequential Gaussian and indicator approaches have been the main simulation methods to generate models of continuous and categorical values, respectively (Deutsch, 2002; Deutsch & Journel, 1998). However, geological datasets do not generally conform well to parametric Gaussian distributions, as it has become apparent that these two-point-based simulation algorithms cannot reproduce the non-linearities that are common to most geological structures (Guardiano & Srivastava, 1993; Strebelle, 2000). Such models are based on the underlying assumption that the spatial variability of attribute values is fully characterized by a covariance function (Goovaerts, 1997). In addition, Gaussian models have a maximum entropy property, which breaks down the spatial continuity of extreme values (Journel & Deutsch, 1993). Indicator simulation approaches have commonly been used to generate class-specific patterns that maintain spatial continuity. However, these methods are also hindered by two-point covariance models that are time consuming, and generate unnatural “patch-like” patterns (Machuca-Mory, Ortiz, Deutsch, 2008).

A practical representation of the connectivity of values in space is critically important (Allard, 1993; Journel & Alabert, 1989). Connectivity is here loosely defined as the probability that two points that are separated by a lag along a certain direction will be connected by similar values (Renard & Allard, 2013). In a mining context, there is an important economic aspect associated with the spatial connectivity of grades, particularly high grade values driving the optimization process, that must be considered in order to derive feasible exploitation schedules under a set of technical constraints; this connectivity ultimately drives the optimization process and results in production schedules and a project’s net present value (NPV) (Dunham & Vann, 2007; Godoy & Dimitrakopoulos, 2011).

Since the development of the first sequential spatial simulation models (Journel, 1974; Journel & Alabert, 1989; Journel & Huijbregts, 1978), researchers beginning with Guardiano and Srivastava (1993), have contributed a range of advanced stochastic imaging methods known as multiple-point statistic (MPS) and high-order statistic (HOS) simulation algorithms. Examples of these algorithms include SNESIM (Strebelle, 2000; Strebelle, 2002), SIMPAT (Arpat, 2005), FILTERSIM (Wu, Zhang, & Journel, 2008; Zhang, 2006), DISPAT (Honarkhah & Caers, 2010; Honarkhah & Caers, 2012), IMPALA (Straubhaar, Renard, Mariethoz, Froidevaux, & Besson, 2011), Direct Sampling (Mariethoz, Renard, & Straubhaar, 2010; Mariethoz & Renard, 2010), HOSIM (Mustapha & Dimitrakopoulos, 2010) and WAVESIM (Chatterjee et al., 2012). These algorithms exist collectively within the field of multiple-point statistics, which is to say, statistics that are calculated from more than two points. MPS and HOS simulation algorithms aim to reproduce complex connectivity patterns. However, these algorithms cannot consistently calculate such multiple-point or high-order statistics without the additional data input from an auxiliary model known as a training image (TI). The focus of this thesis will be on the generation of such a TI.

Recent advances in sparse signal reconstruction have opened the door to a new field known as compressed sensing (Candès & Wakin, 2008). The application of compressed sensing theory (Candès, Romberg, & Tao , 2006a; Candès, Romberg, & Tao , 2006b; Candès & Tao, 2006; Candès & Tao, 2010; Donoho, 2006) to 2D datasets has led to extremely useful reconstruction techniques known as matrix completion (Cai, Candès, & Shen, 2010; Keshavan, Montanari, Oh, 2010) and, by its extension to 3D datasets, tensor completion (Gandy, Recht, & Yamada, 2011; Liu, Musialski, Wonka, & Ye, 2009, 2013; Romera-Paredes & Pontil, 2013; Signoretto, Dinh, De Lathauwer, & Suykens, 2013; Signoretto, Van de Plas, De Moor, & Suykens, 2011). These tools are used to reconstruct sparse datasets by taking advantage of a signal's compressibility (i.e. redundancy of structure).

The scope of this thesis, as highlighted by Figure 1, is to present a method to build a 3D continuous TI (C) through the application of a tensor completion algorithm (B) using limited drilling information (A). The motivation for such work is to provide the continuous

data TI, required by MPS and HOS simulation algorithms (D), in such a way that the MPS simulations (E) generated become entirely data-driven. The reconstruction (i.e. the TI) at unobserved sample locations is guided by the 3D global structure of the original samples, up to the HOS; the specific meaning of "global structure" is explained in Section 3.1. Step B in Figure 1 is accomplished through a method, developed in this thesis, called Multi-Stage Tensor Completion (MSTC). In a related work (Yahya, 2011), a matrix completion algorithm was used to reconstruct 2D continuous TIs using a stepped approach called Multi-Stage Matrix Completion (MSMC). That work is reviewed in Section 3.1.4 to provide a basis for the development of MSTC in this thesis.

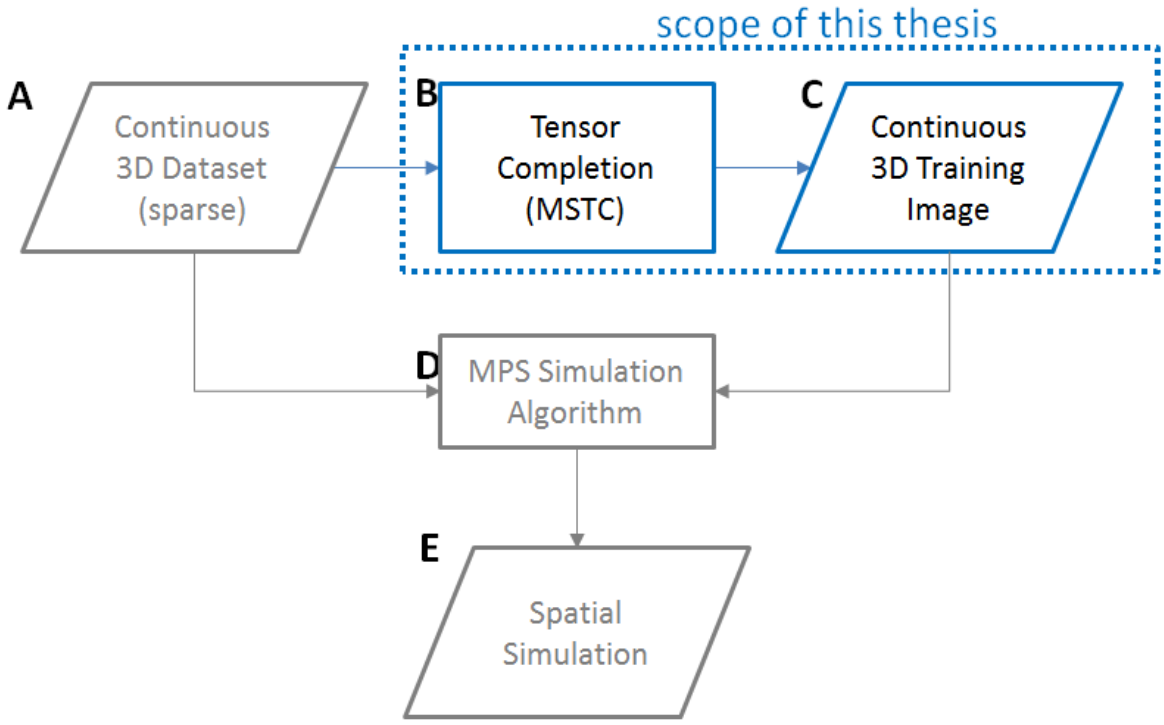

Figure 1. Scope of this thesis: to build a 3D training image (C) from sparse continuous data drill samples (A) using MSTC (B). Training images are a necessary input into MPS algorithms (D), which are used to build spatial simulations (E) of the geological phenomena.

1.1 Goal and Objectives

The goal of this thesis is to introduce the development of an exhaustive (i.e. no missing values) 3D continuous data TI that is built using MSTC and a sparse reconstruction algorithm called LRTC (Liu et al., 2009) in such a way that it captures (to the extent

possible) the spatial HOS of the observed samples (or entries, both terms are used interchangeably in this thesis).

The main objectives of this thesis are the following:

1. Extend the 2D methodology of generating continuous TIs using MSMC (Yahya, 2011; Yahya, Dimitrakopoulos, & Psaromiligkos, 2012) to 3D datasets, all the while respecting the original data and their spatial high-order relations. In this thesis, this objective is achieved through the application of the LRTC algorithm (Liu et al., 2009; Liu, Musialski, Wonka, & Ye, 2009, 2013) within a method defined as MSTC. Spatial HOS of the reconstructions are evaluated against those of the samples and the original TI, using spatial cumulants (Dimitrakopoulos, Mustapha, & Gloaguen, 2010; Mustapha, Dimitrakopoulos, & Chatterjee, 2011).
2. Test variants of MSTC on continuous porosity samples from two synthetic benchmark examples of a petroleum reservoir model known as Stanford V (Mao & Journel, 1999).

The remainder of this thesis is organized as follows: Chapter 2 begins with a literature review describing the MPS simulation framework, which uses TIs. Why and how MPS simulation algorithms use TIs is described through an example of a popular MPS algorithm called SNESIM (Strebelle, 2000). Chapter 2 also includes a review of various methods available to build TIs for MPS simulations. Chapter 3 specifically describes how matrix completion was used to construct 2D TIs via the MSMC method (Yahya, 2011; Yahya et al., 2012). Chapter 4 introduces the notion of tensors and describes the LRTC algorithm (Liu et al., 2009; Liu et al., 2013) based on low-rank matrix completion. Chapter 4 presents the method of MSTC based on the LRTC algorithm, which is applied on two synthetic (benchmark) petroleum reservoir datasets. Results from each of the benchmark cases are presented and analysed in Chapter 5. The method was further tested on a confidential dataset of an orebody for which general results are discussed but details cannot be shown. The final chapter, Chapter 6, starts by summarizing the work presented in this thesis, reflects on the advantages and drawbacks of the MSTC methodology, and concludes with suggestions for further work.

CHAPTER 2 LITERATURE REVIEW

This chapter reviews the method by which MPS algorithms use TIs to generate spatial simulations. A description of the classic MPS algorithm SNESIM (Strebelle, 2000) is included, together with a brief review of other well-known MPS algorithms. Commonly available methods for acquiring TIs are reviewed starting from section 2.2.

Geostatistical datasets are often presented in one of two forms: categorical or continuous. Categorical data result from the grouping of individual point values into groups or categories according to a common attribute. For example, sandstone, limestone, and shale may all be placed under the category label “sedimentary” rocks. By contrast, continuous values are described along the real number line. Metal grades are usually reported in continuous terms, as for example 1.0% copper or 16.49g/ton of gold. Continuous values may be grouped into categories and thus redefined under categorical labels. Categorical values generally cannot be converted to continuous values.

2.1 The Multiple-Point Simulation Paradigm

The MPS simulation paradigm was introduced by Guardiano and Srivastava (1993). Strebelle (2000) implemented the SNESIM algorithm as the first computationally practical application of MPS simulation for the reproduction of complex categorical patterns through the sequential use of conditional distributions calculated from a TI. The application of MPS simulation to continuous variables was done later through the FILTERSIM (Zhang, 2006) and Direct Sampling algorithms (Mariethoz & Renard, 2010). A TI (discussed in Section 2.2) is the term given to a geological model of reality that contains, as far as possible, the spatial complexity that needs to be simulated. The MPS simulation approach was developed to address the shortcomings of two-point simulation methods (Journel, 1974; Chilès & Delfiner, 1999; David, 1988; Goovaerts, 1997; Journel & Huijbregts, 1978), which are not capable of reproducing curvilinear features (e.g. meandering river channels) and are prone to order relations problems. Order relations describe transition probabilities between categories, or between ranges of continuous values; problems in order relations may occur from insufficient data (Deutsch & Journel,

1992; Goovaerts, 1997). MPS and HOS simulations have the ability to capture high-order spatial structures by reproducing the spatial connectivity of extreme values that is important in petroleum flow and mine scheduling problems. Figure 2 presents a binary example (Caers & Zhang, 2004) that clearly shows how variograms, which represent models of connectivity that measure the variance of the difference between attribute values at two different locations (Cressie, 1993), cannot distinguish between the relatively simple, yet obviously different, connectivity patterns exhibited by the three binary TIs because they all share similar two-point statistics (i.e. the statistical relation of two points). A similar comparison is made in Figure 3, in which the histogram and variogram show a close match between two different simulations based on two different algorithms, though their spatial structure is visibly different. The Direct Sampling simulation on the left (Mariethoz, Renard, & Straubhaar, 2010) is able to generate more realistic geological connectivity structures while the Sequential Gaussian Simulation on the right displays the typical Gaussian-type high entropy pattern (discussed in Section 2.2). The importance of connectivity is well known in reservoir characterization and hydrogeology, as it impacts on reservoir flow and solvent transport (Renard and Allard, 2013). In mining, spatial variability can have serious economic implications in the extraction of deposits (Dimitrakopoulos et al., 2002; Dunham & Vann, 2007; Godoy & Dimitrakopoulos, 2011; Vallee, 2000).

The next section reviews the sequential simulation framework that MPS simulations are based on. The MPS simulation paradigm was developed to reproduce complex features by detecting differences in geological heterogeneity, such as shown in the images of Figure 2.

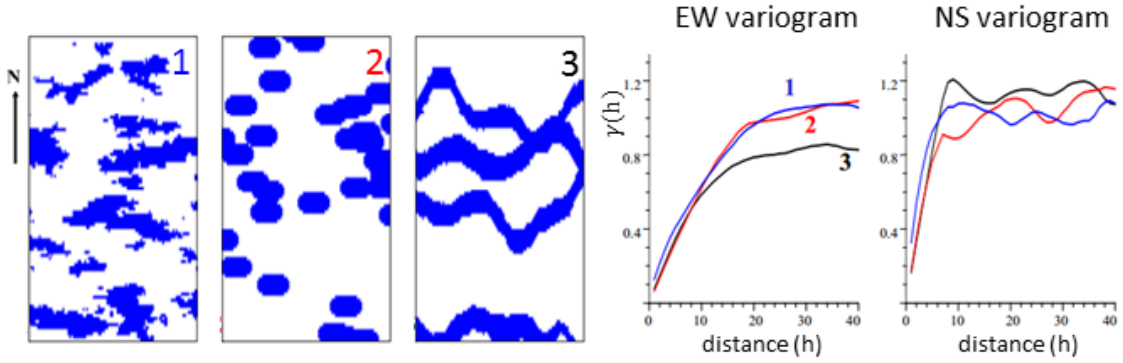

Figure 2. Comparison of variograms for different TIs (Caers & Zhang, 2004).

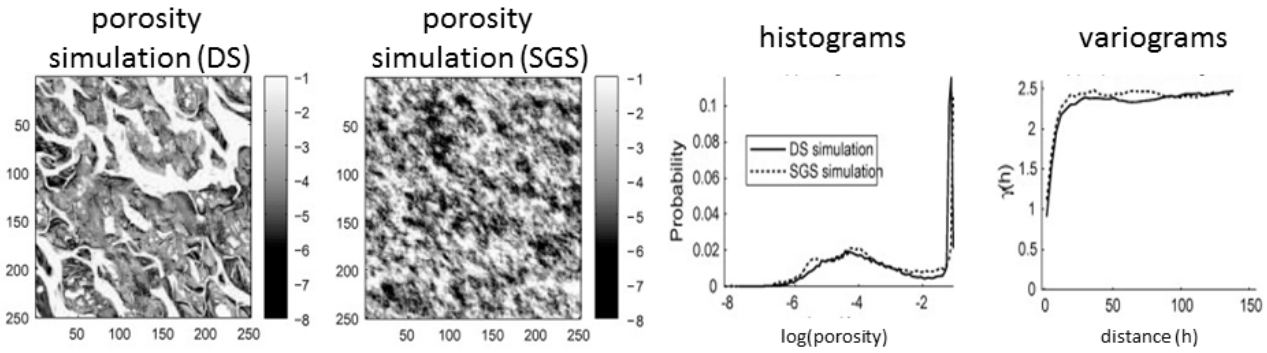

Figure 3. Two different images, based on the Direct Sampling MPS algorithm and the Sequential Gaussian Simulation algorithm, have the same two-point statistics (histogram, variogram) (Renard & Allard, 2013).

2.1.1 The Sequential Simulation Framework

The sequential simulation method (Journel 1989; Deutsch and Journel, 1998; Goovaerts, 1997) is used to generate multiple realizations, each having the spatial statistics of interest (mean and variance only). Consider a set of N random variables $Z(\boldsymbol{u}_\alpha)$, $\alpha = 1, \ldots, \mathrm{N}$ defined at N different locations $\boldsymbol{u}_\alpha$. In a conditional sequential simulation, a joint realization $z(\boldsymbol{u}_\alpha)$ of the N random variables is generated such that the properties of a given multivariate conditional cumulative distribution function (CCDF) **2.1**, conditional to n available data, are reproduced.

$$F(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N; z_1, \ldots, z_N | (n)) = P\{Z(\boldsymbol{u}_1) \leq z_1, Z(\boldsymbol{u}_2) \leq z_2, \ldots, Z(\boldsymbol{u}_N) \leq z_N | (n)\}, \qquad (\mathbf{2.1}).$$

Through a recursive application of Bayes' Law, the CCDF in **2.1** may be decomposed into the product of N univariate CCDFs (Ripley, 2009; Rosenblatt, 1985), given the initial n conditioning data $\Lambda_0 = \{z(\boldsymbol{u}_\alpha), \alpha = 1, \ldots n\}$:

$$\begin{aligned} F(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N; z_1, \ldots, z_N | \Lambda_0) \\ = F\{\boldsymbol{u}_N; z_N | \Lambda_{N-1}\} \\ \cdot F\{\boldsymbol{u}_{N-1}; z_{N-1} | \Lambda_{N-2}\} \\ \cdot \ldots \cdot F\{\boldsymbol{u}_2; z_2 | \Lambda_1\} \cdot F\{\boldsymbol{u}_1; z_1 | \Lambda_0\}. \end{aligned} \qquad (\mathbf{2.2}).$$

In words, **2.2** states that the CCDF of the random function $Z(\boldsymbol{u})$ is the product of N random variable CCDFs. Each successive simulated location becomes part of the conditioning dataset. For example, Λ_1 represents the original sample data Λ_0 plus the first simulated point and Λ_{N-1} is comprised of the original samples and the $(\mathrm{N}-1)$ previously simulated nodes. A realization of the CCDF in **2.1** is generated through N sequential steps, summarized by:

1. Model the CCDF at an unsampled location based on the available data, i.e., samples plus any previously simulated nodes. The first CCDF is given by

$$F(\boldsymbol{u}_1; z_1|\Lambda_0) = P\{Z(\boldsymbol{u}_1) < z_1|\Lambda_0\}. \quad (\mathbf{2.3}).$$

   Note that the specific uncertainty model used is dependent on the simulation method adopted. For example, under the traditional multiGaussian hypothesis the sequential CCDFs in **2.2** are obtained from a normal distribution whose parameters (mean and variance) are derived from kriging (Deutsch & Journel, 1998; Goovaerts, 1997; Journel, 2005). In a sequential indicator simulation, the CCDF is built from solving a number of indicator kriging systems (Goovaerts, 1997). The MPS uncertainty model is inferred from the TI (discussed in section 2.1.2.)
2. Draw at random from **2.3** the first realization $z(\boldsymbol{u}_1)$. Add this value to the set of conditioning data.
3. Randomly move to another unsampled grid node, visiting each unsampled node only once.
4. Repeat Steps 1-3 until all nodes are simulated.

2.1.2 Measuring Multiple-Point Statistics from a TI

Consider an attribute Z with K categories z_k, for $k = 1, \ldots, K$, or if Z is continuous $(K-1)$ thresholds z_k, for $k = 1, \ldots, (K-1)$. A multiple-point event is represented in Figure 4, where it is defined by n values $z(\boldsymbol{u} + \boldsymbol{h}_\alpha)$ separated from $z(\boldsymbol{u})$ by lag vectors $\boldsymbol{h}_\alpha$, $\alpha = 1, \ldots n$. For simplicity, let $z(\boldsymbol{u}_\alpha) \equiv z(\boldsymbol{u} + \boldsymbol{h}_\alpha)$. The multiple-point event is decomposed into a data event d_n and the value to be simulated $z(\boldsymbol{u})$. Under the

decision of model stationarity, and given the scarcity of available samples in real datasets (<2%), a multiple-point spatial inference CCDF is developed by directly scanning an exhaustive TI that represents the geological features of the physical phenomena (Guardiano & Srivastava, 1993; Journel, 1993; Strebelle, 2000). This framework is exemplified using the SNESIM algorithm in the section that follows.

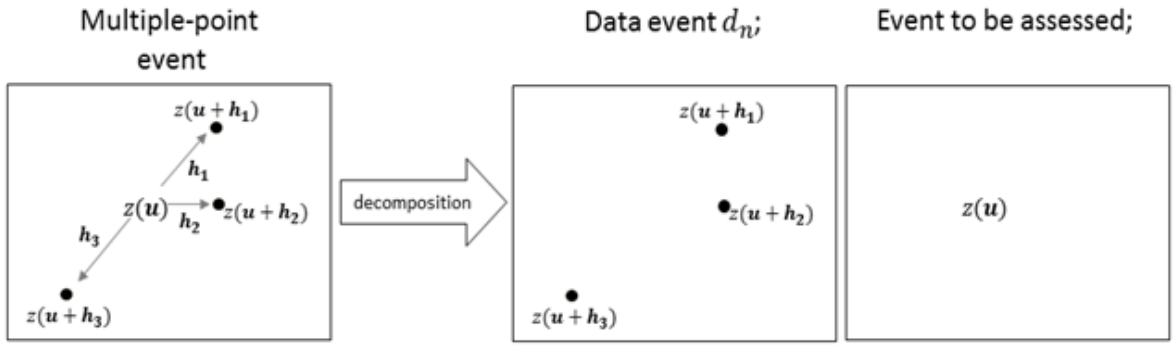

Figure 4. Decomposition of an arbitrary multiple-point event into a data event d_n and $z(\boldsymbol{u})$, the value to be simulated based on d_n.

2.1.3 An Example: The SNESIM Algorithm

The SNESIM algorithm (Strebelle, 2000, 2002) was the first computationally practical MPS simulation algorithm after the initial work by (Guardiano & Srivastava, 1993). It is a pixel-based algorithm designed for categorical data with the ability to condition to data. The algorithm requires a TI to build the CCDF. A description of the methodology (Strebelle, 2002) follows. As shown in Figure 5, a data template τ_n consists of a predefined geometry of $(n+1)$ nodes, centered at location $\boldsymbol{u}$, with the ability to capture any $(1,\dots,n)$-point data event. Data events d_{τ_n} are associated with the data template selected. That is, a data event cannot include a value outside the boundary of the data template – see right side of Figure 5. Pattern frequencies captured by the data template as it scans the image are stored in $c(d_n)$. The number of times the location $\boldsymbol{u}$ belongs to class k is stored in $c_k(d_n)$. The general steps of the SNESIM algorithm are based on sequential simulation. Figure 6 is a useful reference to follow each step.

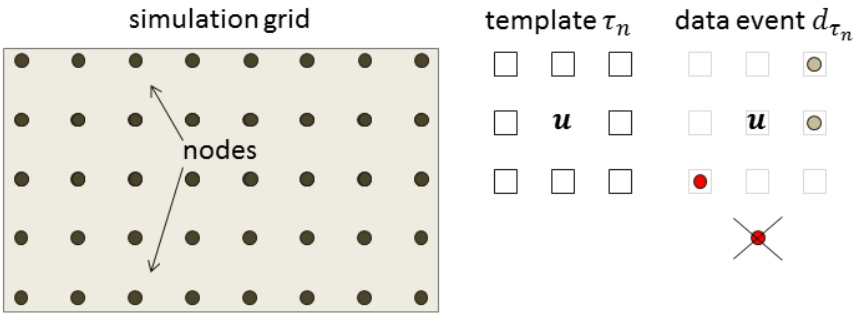

Figure 5. Simulation grid of N=40 nodes (left). A predefined template τ_n is used to scan the TI. A data event d_{τ_n} represents any conditioning data (red/grey circles) screened by the template for a point to be simulated $\boldsymbol{u}$. Points outside of the template are omitted from the calculation.

A. Scan the TI in (1) using the n-point data template τ_n in (2). Store the patterns captured by τ_n along with their counts $c(d_{\tau_n})$ into a search tree. During the scan, the simulation point must lie within the dashed boundary shown over the TI in (1) in Figure 6, otherwise the search template will have search nodes outside the TI.

B. Store the number of times a data event d_{τ_n} has a central value (at $\boldsymbol{u}$) belonging to class k as $c_k(d_{\tau_n})$ into a search tree (3) and calculate the respective multiple-point histograms (4).

C. Infer the CDF $F(\boldsymbol{u}; z_k|d_{\tau_n})$ conditioned on the different data events by substituting the values from Steps A and B into Bayes relationship in **2.4**:

$$F(\boldsymbol{u}; z_k|d_{\tau_n}) = P\{Z(\boldsymbol{u}) = z_k|Z(\boldsymbol{u}_a) = z_{k_a}\} \cong \frac{c_k(d_{\tau_n})}{c(d_{\tau_n})} = \frac{\begin{array}{c}number\ of\ times\ \boldsymbol{u}\\ is\ of\ class\ k\ for\\ a\ given\ pattern\ d_{\tau_n}\end{array}}{\begin{array}{c}number\ of\ times\\ pattern\ d_{\tau_n}\ is\ found\ in\ the\ TI\end{array}}, \qquad \textbf{(2.4)}$$

D. Define a random path on the simulation grid and simulate each node along the path as follows:

   i. Obtain all data events comprised of the surrounding data and any previously simulated nodes,

   ii. Derive the conditional density function using Step C.

   iii. Generate a simulated value by drawing a random value from the CCDF in **2.4** and adding it to the grid.

E. Repeat steps A-D to generate a new realization.

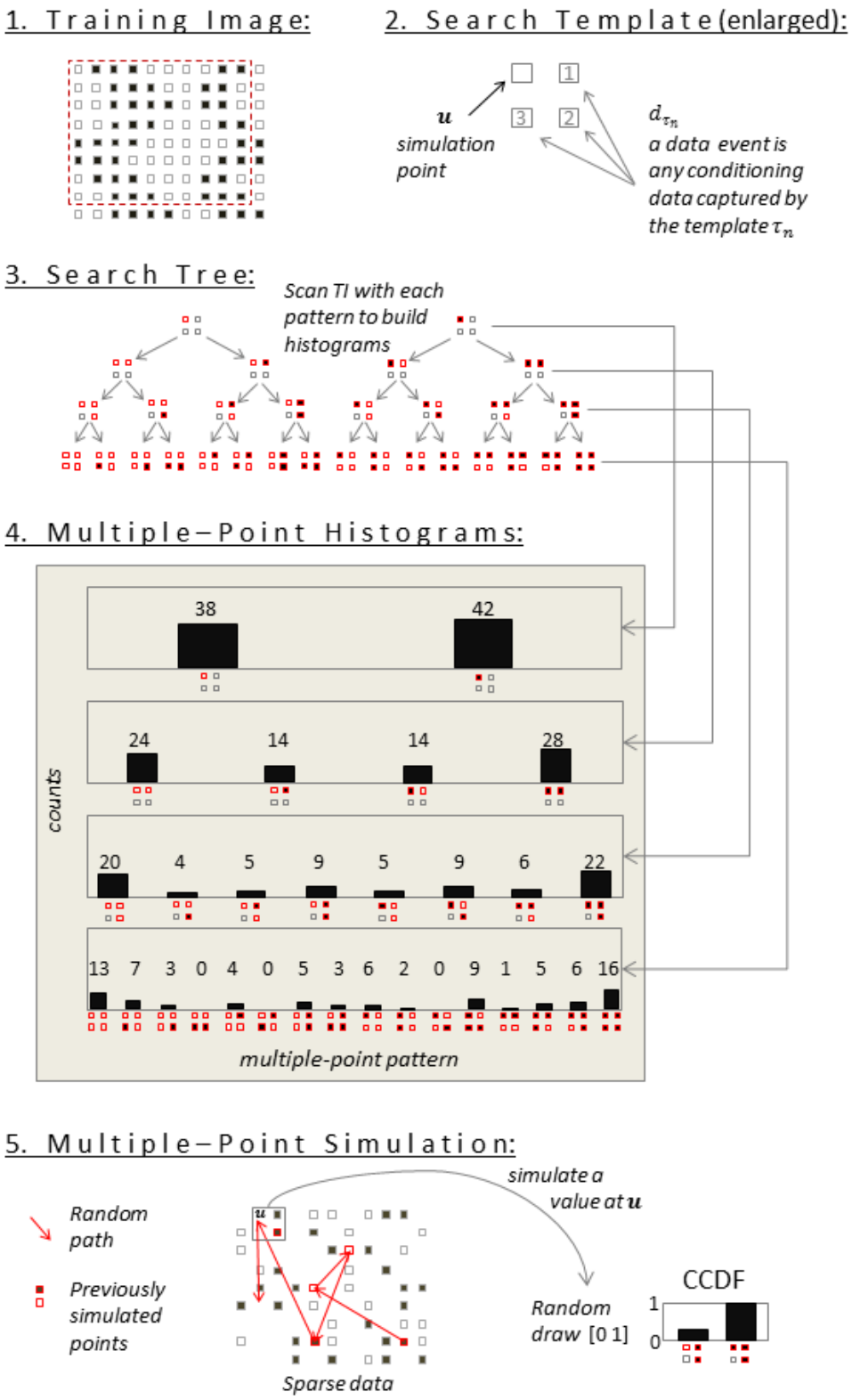

Figure 6. Basic steps in a SNESIM simulation. To simplify the schematic, the simulation point $\boldsymbol{u}$ is chosen *off-centre* in the template. In the search tree, active nodes from the search template are outlined in red.

2.1.4 Other MPS-Type Approaches

A number of multiple-point statistics (MPS) algorithms have been developed since SNESIM. The IMPALA algorithm (Straubhaar et al., 2011) is based on mapping the conditional distributions into a list structure rather than a search tree. It gains a big computational advantage by parallelizing the computation of the CCDF in each level of the search tree shown in Figure 6. Many MPS algorithms, such as FILTERSIM (Zhang, 2006), SIMPAT (Arpat, 2005), DISPAT (Honarkhah & Caers, 2010), Direct Sampling (Mariethoz et al., 2010), and WAVESIM (Chatterjee, Dimitrakopoulos, & Mustapha, 2012) use *scores* to help categorize data events captured by the search template as it scans the TI. A *distance* measure is defined to calculate score differences and thereby select the most appropriate pattern or pixel to paste into the simulation grid based on a minimum value. An example of a distance measure is given by the Manhattan distance between a data event d_{τ_n} centered at $(\mathbf{u})$ and a pattern $pat_{\tau_n}^k$ for a given template geometry τ_n (Arpat, 2005):

$$dist\left(d_{\tau_n}(\mathbf{u}), pat_{\tau_n}^k\right) = \sum_{k=1}^{n} \left| d_{\tau_n}(\mathbf{u}_\mathrm{k}) - pat_{\tau_n}^k \right| \qquad \textbf{(2.5)}$$

In **2.5**, the score values are defined by the absolute value difference between collocated points of $d_{\tau_n}(\mathbf{u})$ and a pattern $pat_{\tau_n}^k$, taken one at a time, across an n-point data template event τ_n for a set of K patterns. The set K may be defined by grouping patterns into clusters, or categories. A common way of grouping these patterns is through K-means clustering (Ding & He, 2004; Hartigan & Wong, 1979; MacQueen, 1967). Distance-based MPS simulation algorithms select a cluster class, or a single prototype pattern, to represent the class that leads to the smallest distance. They are generally fast due to the dimensional reduction achieved by working in the score-space, in some cases up to 10 times faster for simulations of 3D datasets (Wu et al., 2008). Most distance-based MPS algorithms generate a simulated point, or pattern, by randomly drawing from the minimum-distance class rather than by sampling a CCDF built from patterns in the class. This distinction is addressed by the WAVESIM algorithm (Chatterjee et al., 2012) which applies a wavelet decomposition to template patterns

and uses the K-means clustering algorithm to group the resulting approximate sub-bands (i.e. wavelet coefficient scores) for measuring distances. Recently, an application of distance measures was used to quantify the worthiness of different MPS simulation algorithms by jointly capturing (and maximizing) the variability between multiple realizations together with the variability inherent within each realization (Xiaojin, Tahmasebi, Caers, 2014).

The High-Order SIMulation algorithm HOSIM (Mustapha & Dimitrakopoulos, 2010; Mustapha & Dimitrakopoulos, 2011) constitutes a different approach to MPS simulation by generating the conditional cumulative distribution function from spatial HOS, namely moments or cumulants. An intrinsic property of spatial HOS is their ability to capture directional multiple-point connectivity - including the connectivity and periodicity of extreme values - to define geometric characteristics in 2D and 3D datasets. In HOSIM, the CCDF is approximated using Legendre polynomials, whose coefficients are calculated as combinations of high-order moments. HOS are calculated from the template using a TI and all available data. Hence, the need for a TI free of data conflicts at the sample locations that accurately reflects the spatial HOS of the data.

In any application, the specific MPS simulation algorithm used will induce the assumption of a stationary random function (RF) model, where stationarity implies that the RF has equivalent statistical properties throughout. For example, the SNESIM algorithm samples the CCDFs built during the TI scan (Figure 6) assuming that patterns detected in one or more parts of the TI are applicable at any other location on the simulation grid. Natural systems are often best represented by non-stationary RFs that cannot be reproduced accurately with MPS simulation algorithms that assume a stationary model – see Figure 7. This issue has been successively addressed in various ways since the initial MPS algorithms were introduced (Strebelle, 2000). The concepts of data event rotations and pattern-scaling (affinity) to find a matching pattern in the TI were introduced into algorithms (Caers & Zhang, 2004; Wu, 2007; Zhang, Bombarde, Strebelle, & Oatney, 2006) to transform parts of the TI depending on the simulation node location. Honarkhah and Caers (2012) have incorporated location information into their Spatial Similarity Method (SSM) by weighing patterns according to their Euclidean distance from the location of the simulation node. This algorithm is suitable for

categorical and continuous data but it remains arbitrary with respect to the definition of suitable weights (i.e. the algorithm uses pattern weights and location weights that are user-defined parameters which may be difficult to decide on).

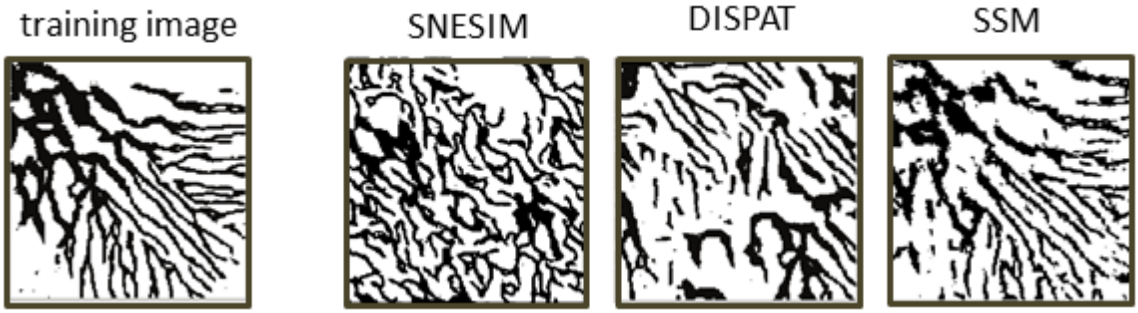

Figure 7. Simulation of a fluvial fan deposit using three different MPS simulation algorithms that assume a stationary model. In this example, only the last one accounts for a location-dependent pattern database. (Summary of results from Caers and Zhang, 2004; Honarkhah and Caers, 2012; Michael et al., 2010).

The requirement for relevant TIs to simulate the desired geological heterogeneity is well known (Arpat, 2005; Boucher, Gupta, Caers, & Satija, 2010; Chatterjee et al., 2012; Deutsch & Journel, 1992; Hu & Chugunova, 2008; Maharaja, 2008; Mustapha & Dimitrakopoulos, 2010; Pyrcz, Boisvert, & Deutsch, 2008; Strebelle, 2002). All MPS and HOS simulation algorithms rely on a TI to access and reproduce the MPS or HOS. However, as a recent paper explains, TIs used in MPS simulation algorithms are not substitutes for a geological random field model (Emery & Lantuejoul, 2014) since they face limitations related to a) the size of the TI available with respect to the template size used to scan for patterns (e.g. smaller templates provide less accuracy) and b) the spatial distribution of the random field (i.e. whether there is a sufficiently high template probability of occurrence in order to reproduce the desired complexities in the simulation grid). From this perspective, MPS simulations deviate from a geostatistical reproduction of the input statistics when they rely on a TI that does not respect the same statistics. In contrast, the method used to generate TIs in this thesis promotes the reproduction of the original data statistics.

Methods presented in the literature for building TIs are discussed next.

2.2 Training Images

In this section, a number of methods that have been used to build TIs are reviewed. A *training image* is the term used to describe an exhaustive (no missing entries, or samples) model of reality. In geostatistics, this 'image' is generally represented by a uniform grid spanned by a matrix in two dimensions, or a third order tensor in three dimensions. The image can also be multi-dimensional (Mariethoz et al., 2010). A TI is important as it represents a necessary repository of statistically relevant spatial structures (i.e. repeatability of patterns), used by MPS simulation algorithms, to impart on a realization (Guardiano & Srivastava, 1993; Strebelle, 2000; Zhang, 2006). Structural features at various scales need to represent the geological heterogeneities that may exist. Compared to a variogram, it is easier to visually inspect, accept, or reject, a TI based on its resemblance to a geological reality – see Figure 2 and text in Section 2.1.

In ore mining as in reservoir characterization, the spatial variability of values is an important property. In mining the spatial connectivity of extreme grade values ultimately defines an ore extraction schedule and its associated Net Present Value (Dimitrakopoulos, 2011; Godoy & Dimitrakopoulos, 2011). In reservoir models, the spatial connectivity of porous media (i.e. permeability) is crucial to flow simulation studies. In both cases, the spatial connectivity of values, particularly that of extreme values, that is generated by a two-point simulation method such as the sequential Gaussian simulation, does not reflect the true connectivity of grades since this method is known to produce maximum entropy connectivity patterns (Journel, 2005; Journel & Alabert, 1988; Journel & Deutsch, 1993). Conceptually, this means that the image contains the minimal amount of structure that is necessary to fit the data (see SGS simulation in Figure 3) since high and low values are spread out as uniformly as possible over the simulation grid (Gull & Skilling, 1984; Jaynes, 1989). In practice, such high entropy structures can lead to restricted flow scenarios in reservoir characterization models (dell'Arciprete et al., 2012; Gómez-Hernández & Wen, 1998; Journel & Alabert, 1988) and suboptimal ore extraction schedules in mining sequences (Dimitrakopoulos, 2011; Dimitrakopoulos et al., 2002; Dunham & Vann, 2007). MPS and HOS simulations, which rely on TIs to capture multiple-point spatial connectivity, seek to avoid this. In

MPS simulation algorithms, the relevance of multiple-point structures is defined by their repetitive character or by measures of distance to account for similarity (Boucher et al., 2010; Hu & Chugunova, 2008; Strebelle, 2002). Based on this, the TI should not introduce patterns that are unlikely to pertain to the phenomena in question (Maharaja, 2008). The local accuracy of the TI used is increasingly more important, given new simulation algorithms that keep track of pattern locations (Honarkhah & Caers, 2012, Mariethoz & Kelly, 2011).

The requirement for suitable TIs has grown with the success of TI-based algorithms used in modeling applications; as aforesaid, these methods rely on a TI to extract some or all of the spatial complexity. However, as will be discussed in the following sections, the methods used to obtain TIs, particularly continuous TIs, have a limited range in the literature (Boisvert, Pyrcz, & Deutsch, 2007; Boucher et al., 2010; Chugunova & Hu, 2008; Hu & Chugunova, 2008; Maharaja, 2008). According to Boucher et al. (2010), algorithms used to generate TIs are still incomplete and unsatisfactory in terms of their applications to the mining industry (Lantuéjoul, 2002). Most of them are aimed to represent fluvial environments. In the mining industry, methods for building TIs have been addressed most often by "borrowing" information from more exhaustive datasets originating from a nearby source (Goodfellow et al., 2012; Osterholt & Dimitrakopoulos, 2007) or by spatially-domained Gaussian simulations (Machuca-Mory & Dimitrakopoulos, 2011). In the former case, an assumption is made about the statistics of the data. In the latter case, second-order spatial restrictions are placed on the connectivity structure of the data.

The main methodologies for generating TIs are *object-based*, *process-based*, and expertly-derived or previously mined geological analogs. Examples of each type are presented in Figure 9 on page 22, and reviewed in the subsections that follow. However, object- and process-based models have limited use for generating TIs in a mining context. An extensive review by Koltermann and Gorelick (1996) presented various methodologies for generating TIs. However, these methods were limited to generating models of aquifer heterogeneity in sedimentary environments.

The methodology for constructing 2D continuous TIs via matrix completion and MSMC (Yahya, 2011; Yahya et al., 2012) is thoroughly reviewed in Chapter 3.

2.2.1 Current Training Image Generation Approaches

2.2.1.1 Object-based Methods

Object-based methods are intended to model simple categorical geometries and as such cannot be used to model the spatial continuity of continuous grade values. The method is reviewed briefly for completeness.

Object-based methods are sometimes called *marked point* or *Boolean techniques*. The literature on marked point process models for geometric objects is covered by (Stoyan, 1987). The method was first applied to simulate shales (Haldorsen & Lake, 1984) using a simulation algorithm developed earlier (Hastings, 1970). A marked point process is a collection of randomly placed points in space with additional information available at each point (e.g. grade, or porosity). Boolean techniques refer to models created through interactions of specified geometries sampled from simulated distributions. These categorical methods are normally used when the geological heterogeneity is controlled by relatively simple parameterizable geometries (Haldorsen & Chang, 1986; Shmaryan & Deutsch, 1999) that are specific to a geological environment (Tetzlaff, 1989) (e.g. a sinusoidal fluvial channel). Object-based TIs are primarily built via unconditional realizations – that is, simulations that sample the correct data distribution but are not constrained to the location of known values. Object-based modelling entails two steps: 1) the assignment of geometric parameters by randomly sampling a distribution, for example a distribution of sphere diameters observed in the field, and 2) the random placement of each geometry (including its orientation) within the model. The inference of these parameters and the conditioning of models to data remain a challenge for this method (Boucher et al., 2010; Maharaja, 2008; Ortiz & Deutsch, 2004). Unconditional simulations are fast, however conditioning these models to local data is best when the data are sparse and no trends exist. Otherwise, the model building is slow as geological bodies are built through trial-and-error leading to higher CPU times (Allard et al., 2005; Michael, Boucher, Sun, Caers, & Gorelick, 2010).

Object-based models are not common in the mining literature as they are generally aimed at fluvial systems. Example simulators include FLUVSIM (Deutsch & Tran, 2002; Deutsch & Wang, 1996), LOBESIM (Pyrcz & Deutsch, 2004), and ELLIPSIM (Deutsch & Journel, 1992). More recent object-based TI generators include TiGenerator (Maharaja,

2008) and Tetris (Boucher et al., 2010). Tetris is useful to construct complex categorical shapes.

Reservoir modeling applications with very sparse data (wells every few hundred metres) are better suited to conditional object-based simulation techniques (Strebelle, 2005) compared to the dense drill datasets available in ore mining. In a mining scenario, it is not possible to use a categorical object-based TI to capture the spatial variability of continuous grades in an ore deposit. As stated in the beginning of this section, the application of object-based methods is ill suited for generating TIs that capture the connectivity of values of a continuous variable (e.g. grade). The method is likely to generate models that conflict with data and fails to reproduce the high-order spatial relations required in the TI.

2.2.1.2 Process-based Methods

A *forward* model is the term given to an outcome generated through the application of theoretical principles of physics using mathematical equations. In this sense, *process-based* methods may be used to generate continuous TIs of geological heterogeneity by forward simulations. These techniques are generally restricted to either subsurface fluid flow and transport, or the geologic processes governing sedimentary basin formation and filling (Koltermann & Gorelick, 1996). A process-based method for generating vein formations was presented by Boisvert et al. (2008). The method mimics the flow of ore-bearing fluids to build a TI. The motivation originated in the fact that MPS algorithms in the mining industry are often limited to using TIs based on more densely sampled, or mined, areas. However, finding such datasets with statistics similar to the deposit of interest is difficult in most cases. Briefly stated, the methodology used consists in the application of a flow simulator to a fracture model. TIs are subsequently generated by linking high flow zones with the location of veins and mineral deposition. Four main steps are used: 1) a fracture model is built or otherwise acquired, 2) the permeability across the model is calculated, 3) a flow simulation is initiated, constrained by permeability values, and 4) high flow threshold values are used to mark the location of veins. One drawback of this method is its reliance on a fracture model. However,

depending on the nature of the deposit it may be possible to obtain this information indirectly (e.g. geophysical surveys).

Process-based methods may produce more realistic TIs compared to object-based simulations (Cojan, Fouché, Lopéz, & Rivoirard, 2005; Maharaja, 2008; Pyrcz & Strebelle, 2006). However, these methods are generally used to build reservoir models, including deep water fans, deltas, large scale landscapes, fluvial meandering channels, and deep water meandering channels (Michael et al., 2010; Miller et al., 2008) – see Figure 9-2. In drilling-constrained environments, or even where soft data are available (e.g. seismic), it remains difficult to restrict the process-based models at the conditioning data locations (Hu & Chugunova, 2008). Stochastic approaches to process-based models are possible by changing the initial model parameters. However, the different model outcomes may not be statistically representative of the phenomena to be simulated.

Through forward modeling programs, process-based models use laws of physics to describe fluid flow velocity and particle movement in addition to tracking model topography, suspended sediment composition, and grain size distributions to calculate and store the erosional and depositional characteristics of a physical model in time. Actual physical models (Paola et al., 2001), as well as numerical models, are built to obtain boundary parameters such as turbulence velocity and spatial variability. Empirical equations are developed from such models to describe the erosional and depositional processes (Michael et al., 2010). Process-based models are initiated at a source and rely on rules to form a path through the model that naturally respects source, topographic, and erodability constraints. Many depositional environment architectures are possible through multiple paths, for example, braided fluvial channels or distributary lobes and splays. A process-based model is created by sequentially positioning source control nodes from source to terminus along a desired path. A continuous spline function is fitted to these nodes. Flow energy rules are used to dictate the transition from erosional channels to deposition lobes. Model construction ends when specific user-defined thresholds are met. Process-based models preserve the information generated throughout the construction process. This information enables the measurement of detailed facies proportions over time. Process-based models are

not guaranteed to reproduce the input statistics, as they tend to produce emergent features resulting from the interaction between rules and between rules and model constraints. An example is shown in Figure 8, where a model is built on a grid with regular X-Y cell size spacing but varying vertical cell dimensions. A simulation is developed from initial conditions by creating and removing cells as deposition and erosion occur. Each cell maintains a history log as the model proceeds forward in time (Miller et al., 2008). A very recent review and application of a process-based method for generating TIs was coupled with the Direct Sampling algorithm (Mariethoz et al., 2010) to better understand the hydrogeological connectivity of an aquifer (Comunian, Jha, Giambastiani, Mariethoz, Kelly, 2014). Despite their ability to construct continuous data models, process-based models have limited application in the modeling of mineral deposits. They do not condition well to dense datasets and cannot consistently reproduce the sample statistics.

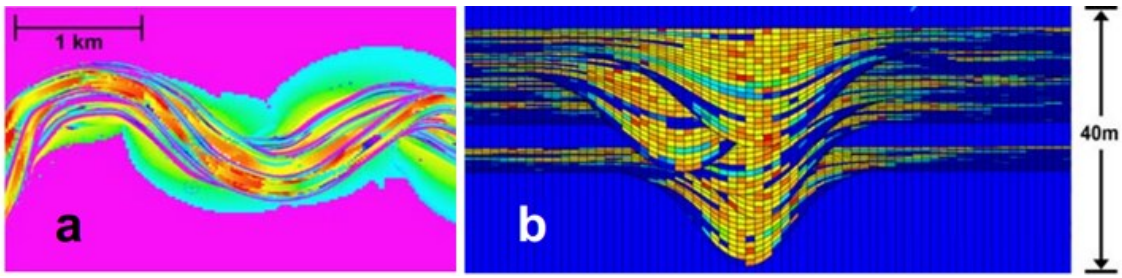

Figure 8. Process based model of a set of deepwater channels. Top view (a) and cross-section (b) showing variable size gridding. Colors indicate grain size: cool = fine, warm = large (Miller et al., 2008).

2.2.1.3 Geological Analogs and Other Methods of Building a TI

Expert-derived geological analogs are tacit models constructed by or with the aid of expert geologists. They may also be geophysical analogs (Gloaguen & Dimitrakopoulos, 2009; Mao & Journel, 1999), or excerpts from adjacent or similar deposits or reservoirs (Goodfellow et al., 2012; Osterholt & Dimitrakopoulos, 2007) – see Figures 9-4 and 9-5. It is often easier to use another part of the deposit as a TI to model connectivity and volumetric uncertainty assuming that the same geological units, with the same statistics, are captured.

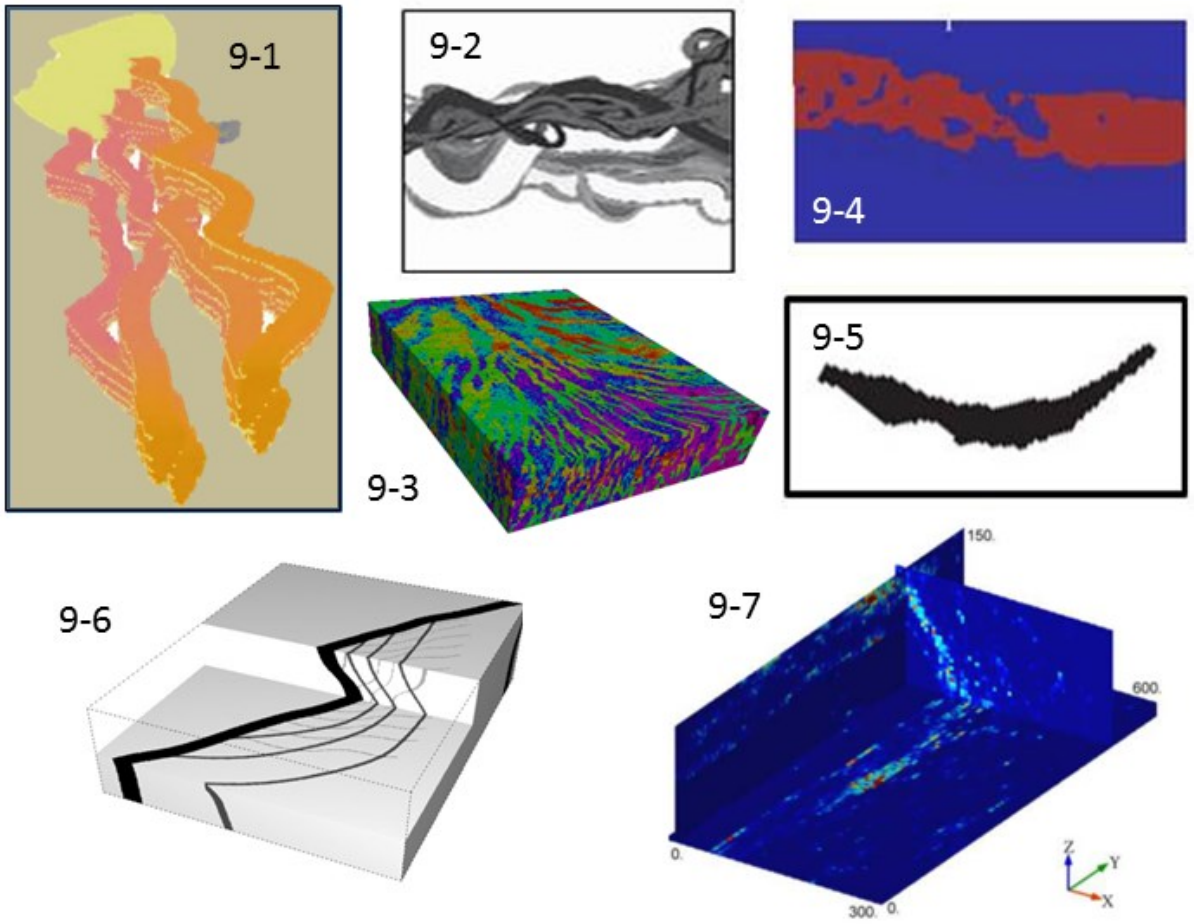

Figure 9. Example of TIs from the MPS literature: -1 is an object-based TI for a series of turbidite flows (Strebelle, 2002), -2 is a process-based TI of a delta channel (Pyrcz et al., 2009); -3 is complex TI from (Honarkhah and Caers, 2012); -4 and -5 are orebody analogs used to model the geometry of an iron ore deposit (Goodfellow et al., 2012; Osterholt et al., 2007), -6 categorical TI of the Apensu deposit built using drillhole information (Jones et al., 2013), 9-7 continuous TI of the Apensu deposit built using sequential Gaussian simulations (Machuca-Mory, et al., 2011).

A categorical TI for the Apensu deposit exists (Jones, Douglas, Jewbali, 2013). Such a TI image was put together using the available drillhole data and identifying three main structures within the data: 1) the main mineralization domain within a 50m wide shear zone, 2) 12-15m curvilinear structures splaying off of the main shear zone and 3) 3-5m extensional fractures primarily located between the second order structures. A fourth category is used to mark whatever remains - see Figure 9-6. It is unclear how the authors connected the data to make one exhaustive TI, however it is reasonable to assume that some expert judgement was required.

A continuous TI for the Apensu deposit was built by Machuca-Mory et al. (2011). That image, shown in Figure 9-7, was built by simulating continuous gold values from drillhole data within each of the categories in the TI of Figure 9-6. While this image is initially appealing, second-order spatial restrictions are placed on the connectivity structure of the data due to the nature of the sequential Gaussian simulation algorithm used, which is the issue to be resolved. Another drawback of this technique is that it generates unnatural discontinuities at the category boundaries since each category domain is simulated separately from the others.

Mariethoz & Kelly (2011) developed a novel multiple-point simulation algorithm in which they use small and simple 3D elemental TI units instead of a global conceptual geological model. Complex simulated structures are possible by applying (locally) random rotations and/or affinity to the units while transform-invariant distances are used to control the reproduction of specific characteristics of the data.

CHAPTER 3 TRAINING IMAGES BUILT WITH MATRIX COMPLETION

This chapter introduces the theory behind low-rank matrix completion. Sections 3.1.3 and 3.1.4 briefly present the Singular Value Thresholding algorithm (SVT) used by Yahya (2011) and by Yahya et al. (2012) to generate 2D TIs using the methodology of MSMC. MSTC is based on the approach of MSMC.

3.1 Low-Rank Matrix Completion

The matrix completion problem is one in which a matrix $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$ is only partially complete; i.e., a percentage of the entries are missing. Matrix completion generates values for missing entries based on information from all available entries. Many methods to solve problems with missing values exist. They generally differ in the assumptions made about the missing entries. For example, interpolation methods (reviewed by Yahya, 2011) will use surrounding data to infer the value of a missing entry while spatial simulation methods (Journel, 1974; Chilès & Delfiner, 1999; David, 1988; Goovaerts, 1997; Journel & Huijbregts, 1978; Remy, 2009) generate the value of a missing entry by sampling from a distribution. In the present thesis, matrix and tensor completion refer specifically to low-rank solution methods.

Low-rank matrix completion is based on the assumption of an underlying global data structure captured from a set Ω of observed entry locations, i.e., location $(i,j) \in \Omega$ if the entry $\mathbf{M}_{ij}$ is observed, which can be leveraged to infer the missing information. This structure is assumed to be of low rank, meaning that there exists a linear dependence between columns/rows that enables the reconstruction of missing entries of the matrix. From this standpoint, the low-rank matrix completion problem statement is to find a unique matrix $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$ with minimum rank satisfying the condition that randomly observed values $\mathbf{M}_\Omega$ are consistent with $\mathbf{X}_\Omega$. That is,

$$\begin{aligned} &\text{minimize} \quad rank\,(\mathbf{X}) \\ &\text{subject to} \quad \mathbf{X}_\Omega = \mathbf{M}_\Omega \end{aligned} \tag{3.1}$$

However, the rank minimization **3.1** is an NP-hard problem for which the only available algorithms require doubly exponential time in the dimension (Cai et al., 2010; Candès & Recht, 2008). To address the non-convexity of the matrix rank function, the optimization problem **3.1** is approximated by the minimization of the nuclear norm $\|\mathbf{X}\|_*$ of the matrix $\mathbf{X}$ with rank r given in **3.2** (Candès and Recht, 2009). That is,

$$\|\mathbf{X}\|_* = \sum_{n=1}^{r} \sigma_n(\mathbf{X}) \qquad \textbf{(3.2)}$$

where $\sigma_n(\mathbf{X})$ is the nth singular value of $\mathbf{X}$ (i.e. the root of the nth eigenvalue of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$). The matrix nuclear norm is the nearest convex relaxation of the matrix rank function (Fazel et al., 2001). Using this approximation, **3.1** becomes

$$\begin{aligned} &\text{minimize} \quad \|\mathbf{X}\|_* \\ &\text{subject to} \quad \mathbf{X}_\Omega = \mathbf{M}_\Omega \end{aligned} \qquad (\textbf{3.3})$$

A number of algorithms exist for solving **3.3**, Michenková (2011) reviews a number of solvers based on two main categories: 1) constrained convex solvers based on Lagrangian, penalty, and the augmented Lagrangian method such as CVX (Grant & Boyd, 2011), SVT: Singular Value Thresholding (Cai, Candès, & Shen, 2010), APGL: Accelerated Proximal Gradient with Line Search (Toh & Yun, 2010), and ALM: Augmented Langrangian Method (Lin, Chen, & Ma, 2010), and 2) algorithms based on the minimization of a function on a Grassmanian manifold such as OptSpace (Keshavan & Oh, 2009), SET: Subspace Evolution and Transfer (Dai, Milenkovic, & Kerman, 2011), and GROUSE: Grassman Rank-One Update Subspace Estimation (Balzano, Nowak, & Recht, 2010). In particular, the SVT algorithm is reviewed in Section 3.1.3 because it has been used to construct 2D TIs from sparse samples (Yahya, 2011; Yahya et al. 2012).

Compressed sensing theory describes how sparse vector signals may be recovered accurately from fewer measurements than necessary by traditional methods (Candès et al., 2006a; Candès et al., 2006b; Candès & Tao, 2006; Candès & Tao, 2010; Candès & Wakin, 2008; Donoho, 2006); traditional methods are based on the Nyquist frequency,

which states that in order to perfectly reconstruct a signal from its samples, the sampling rate must be at least twice the maximum frequency present in the signal.

Two key concepts form the backbone of the compressed sensing framework, which extends to matrix and tensor completion: sparsity and incoherence. In compressed sensing and matrix completion, these concepts lead to provable guarantees of signal reconstruction under certain conditions discussed next. To the best of the author's knowledge, no reconstruction guarantees have been extended to the tensor case.

3.1.1 Sparsity

A vector $\boldsymbol{x}$ is called r-sparse if at most r of its entries are non-zero. The term sparsity is used to capture the notion of a signal's compressibility over a concise vector basis. This notion extends similarly to matrices through the number of degrees of freedom, which is given by $f = n_1 r + (n_2 - r)r$ for a matrix $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$ of rank r. A matrix is low-rank when $r < \min(n_1, n_2)$. For a low-rank matrix, f is considerably lower than $n_1 n_2$. Given a square matrix (i.e., $n = n_1 = n_2$), the degrees of freedom approximate to $2nr \ll n^2$. In other words, it is said that when n is large compared to r the signal information rate is much smaller than the ambient dimension of the matrix (Candès & Recht, 2009; Candès & Wakin, 2008). It is this compressibility property which enables the matrix's recovery from $m \ll n_1 n_2$ entries.

The concept of sparsity may also be expressed through the singular values of a matrix. Consider the singular value decomposition theorem (Golub & Van Loan, 1996) which states that a matrix $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$ can be decomposed into three component matrices as

$$\mathbf{M} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^{\mathrm{T}}. \tag{3.4}$$

The middle factor $\mathbf{\Sigma}$ is a diagonal matrix of singular values given by $\sigma(\mathbf{M}) = diag(\mathbf{\Sigma}) = [\sigma_1, \sigma_2, \dots, \sigma_{\min(n_1,n_2)}]$, where $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min(n_1,n_2)}$. Matrices $\mathbf{U}$ and $\mathbf{V}$ are composed of orthogonal left and right singular vectors $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_r] \in \mathbb{R}^{n_1 \times r}$ and $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_r] \in \mathbb{R}^{n_2 \times r}$, respectively, arranged into the same order as the singular values in $\mathbf{\Sigma}$. With the

SVD, $\mathbf{U}$ and $\mathbf{V}$ form vector bases for the row and column spaces of $\mathbf{M}$, respectively, which act together through the singular values to represent the matrix using $r \leq \min(n_1, n_2)$ basis vectors, where r is the rank of the matrix. Since each column vector in $\mathbf{U}$ and $\mathbf{V}$ is orthogonal to all other vectors in the set ($\mathbf{U}$ and $\mathbf{V}$ form a set of basis vectors), the interaction of their respective i^{th} row vectors generates a different structure in the image whose intensity is controlled by the singular values. Figure 10 illustrates this point.

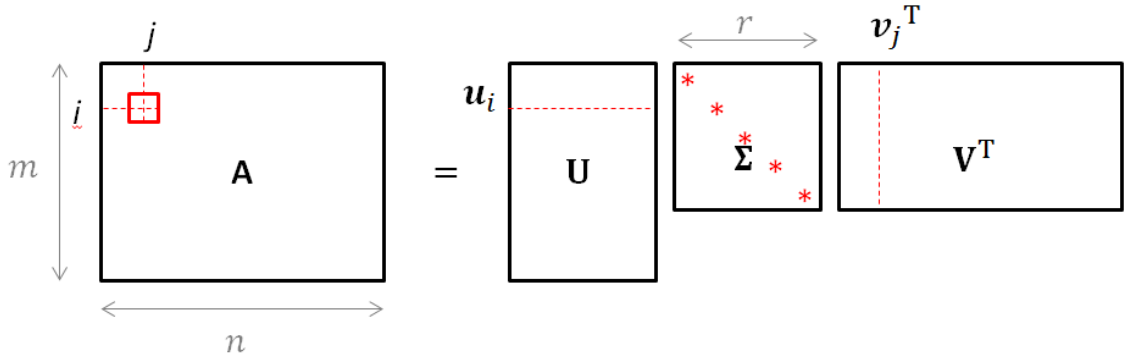

Figure 10. Schematization of the singular value decomposition of a sparse matrix. A point value in image $\mathbf{A}$ is constructed from the interaction of row vectors from $\mathbf{U}$ and $\mathbf{V}$ with the set of singular values.

The relative magnitude of the singular values may be used to further compress the signal by truncating the values that do not significantly contribute to signal strength, albeit with some loss of information (i.e. setting their value to zero does not significantly affect the reconstruction of the original matrix). Figure 11 illustrates the spectrum of singular values of three (60x60) images and the utility of the rank as a measure of their complexity. The sparsity of each image is captured by the number of non-zero singular values. The decay profiles in Figure 11 indicate that both the chessboard (rank 2) and categorical (rank 38) images are low rank while the continuous image is very nearly full rank. Low-rank matrix and tensor completion algorithms reconstruct missing values based on the assumption that the singular values vector is r-sparse or compressible (Candès & Recht, 2009). An image is qualified as compressible when a large number of its singular values may be zeroed, or truncated, with little information loss.

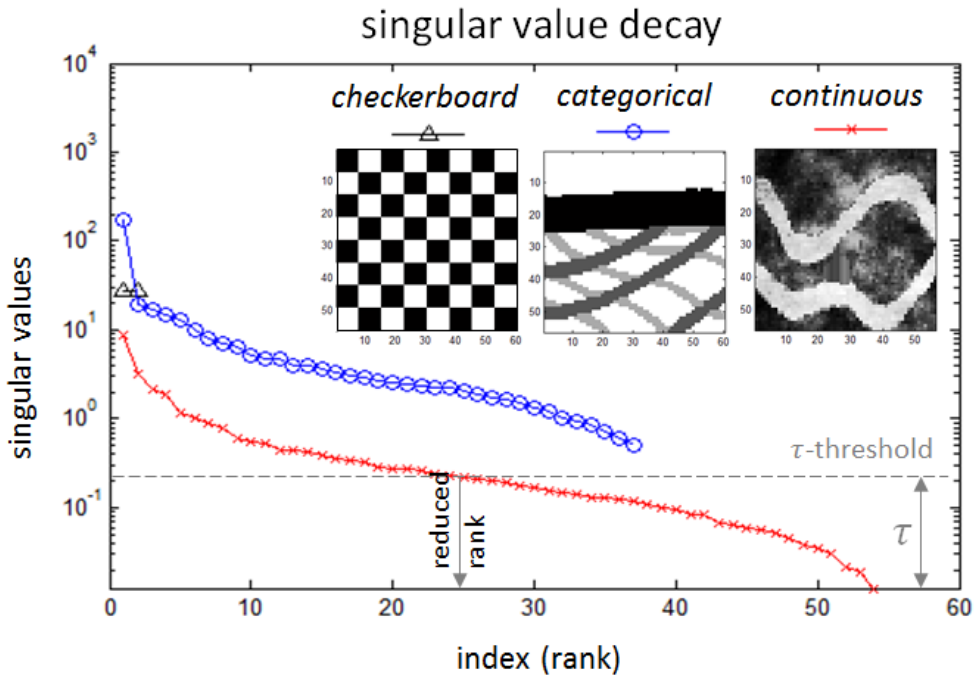

Figure 11. The three singular value profiles capture the sparsity of three successively more complex images. The SVT algorithm (discussed in Section 3.1.3) applies a τ-thresholding operator that reconstructs images based on a reduced rank.

3.1.2 Incoherence

An incoherence property μ describes how difficult it may be to accurately recover a matrix $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$, from m entries that have been drawn uniformly at random from $\mathbf{M}$, by measuring the correlation between its singular vectors $\mathbf{U}, \mathbf{V}$ and the standard basis vectors $\mathbf{e}_i \in \mathbb{R}^n$, for $i = 1, \ldots, n$. The standard basis vector $\mathbf{e}_i$ is a vector of zeroes except for a 1 at the index location i, e.g. $\mathbf{e}_3 = [0\ 0\ 1\ 0]^{\mathrm{T}}$ for $n = 4$. Consider the brief comparative example below of rank 2 matrices $\mathbf{A}$, $\mathbf{B}$, for any arbitrary positive singular values $diag(\mathbf{\Sigma})$ (e.g. $\sigma_1 = \sigma_2 = 1$) where $n = 4$:

$$\mathbf{A} = \mathbf{U_A \Sigma_A V_A}^{\mathrm{T}} = \sum_{k_A=1}^{2} \sigma_{k_A} \boldsymbol{u}_{k_A} \boldsymbol{v}_{k_A}^{\mathrm{T}} \qquad \mathbf{B} = \mathbf{U_B \Sigma_B V_B}^{\mathrm{T}} = \sum_{k_B=1}^{2} \sigma_{k_B} \boldsymbol{u}_{k_B} \boldsymbol{v}_{k_B}^{\mathrm{T}}$$

$$\boldsymbol{u}_{1_{\mathrm{A}}} = \boldsymbol{v}_{1_A} = {\mathbf{e}_1}/{\sqrt{n}}$$

$$\boldsymbol{u}_{2_{\mathrm{A}}} = \boldsymbol{v}_{2_A} = {\mathbf{e}_2}/{\sqrt{n}}$$

$$\mathbf{A} = \begin{bmatrix} {1}/{n} & 0 & 0 & 0 \\ 0 & {1}/{n} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

$$\boldsymbol{u}_{1_{\mathrm{B}}} = \boldsymbol{v}_{1_{\mathrm{B}}} = [\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}]$$

$$\boldsymbol{u}_{2_{\mathrm{B}}} = \boldsymbol{v}_{2_{\mathrm{B}}} = [\frac{1}{\sqrt{n}}, \frac{-1}{\sqrt{n}}, \frac{1}{\sqrt{n}}, \frac{-1}{\sqrt{n}}]$$

$$\mathbf{B} = \begin{bmatrix} {2}/{n} & 0 & {2}/{n} & 0 \\ 0 & {2}/{n} & 0 & {2}/{n} \\ {2}/{n} & 0 & {2}/{n} & 0 \\ 0 & {2}/{n} & 0 & {2}/{n} \end{bmatrix}.$$

Unless most of the values from $\mathbf{A}$ are sampled, such matrices as constructed from sparse singular vectors, such as $\boldsymbol{u}_{1_{\mathrm{A}}}$ and $\boldsymbol{u}_{2_{\mathrm{A}}}$ in the previous example, are difficult to recover accurately in spite of being very low-rank. By contrast, low-rank matrices such as $\mathbf{B}$, constructed from singular vectors that have a more uniform distribution of values (e.g. $\boldsymbol{u}_{1_{\mathrm{B}}}$ and $\boldsymbol{u}_{2_{\mathrm{B}}}$), can be reconstructed from fewer observations.

An incoherence property with parameter $\mu \in [1, \frac{n}{r}]$ is said to exists if

$$\begin{aligned} \|\mathbf{U}^{\mathrm{T}}\boldsymbol{e}_i\|^2 &= \frac{\mu r}{n_1}, \forall\, i, \\ \|\mathbf{V}^{\mathrm{T}}\boldsymbol{e}_i\|^2 &= \frac{\mu r}{n_2}, \ \forall\, i. \end{aligned} \tag{3.5}$$

It has been shown that matrices of rank r sampled uniformly at random that obey the incoherence property may be reconstructed exactly with $\mathrm{O}(nr \log n)$ entries.

3.1.3 An Example: The Singular Value Thresholding Algorithm

The SVT algorithm is a simple two-step iterative algorithm used to reconstruct missing values of a low-rank matrix (Cai et al., 2010). This algorithm was used previously to build 2D Tls (Yahya, 2011; Yahya, et. al., 2012). It is described here to facilitate the exposition of the LRTC algorithm presented in Chapter 4, which is closely related in its approach.

Both the SVT and LRTC algorithms rely on a soft thresholding, or singular value shrinkage, operator $\mathcal{D}(\cdot)$ that acts directly on the singular values of $\mathbf{M}$. For the case of the SVT algorithm, consider $\mathcal{D}_\tau(\mathbf{M}) = \mathbf{U}\mathcal{D}_\tau(\mathbf{\Sigma})\mathbf{V}^{\mathrm{T}}$, where $\mathcal{D}_\tau(\mathbf{\Sigma}) = \{(\sigma_i(\mathbf{M}) - \tau)_+\}_{i=1}^r$ and $\tau > 0$ is called the thresholding parameter. The operator $(\cdot)_+$ is used to keep the positive part of $(\cdot)$. Note that $\mathcal{D}_\tau(\cdot)$ is simply a set of values. When $\mathbf{M}$ has singular values below τ, the rank of $\mathcal{D}_\tau(\mathbf{M})$ will be lower than that of $\mathbf{M}$ (see Figure 11). For all $\tau > 0$ it has been shown (Cai et al., 2010) that the thresholding operator applied on $\mathbf{M}$ is the solution to the dual of the nuclear norm minimization problem shown in **3.3**. That is,

$$\mathcal{D}_\tau(\mathbf{M}) = \arg\ \min_{\mathbf{X}} \{\tau\|\mathbf{X}\|_* + \tfrac{1}{2}\|\mathbf{X} - \mathbf{M}\|_{\mathrm{F}}^2\ \}. \tag{3.6}$$

This leads to the iterative SVT algorithm, which is used to recover a low-rank matrix from a subsample of entries. The SVT algorithm, shown next, has two main steps: a thresholding operation (2.i.) and an update step (2.iii).

Algorithm 1: SVT

Input: Data matrix $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$, auxiliary matrix $\mathbf{Y} \in \mathbb{R}^{n_1 \times n_2}$, sample location indices Ω, step size δ, tolerance ϵ, singular value thresholding parameter τ, maximum number of iterations k_{max}.

Output: Reconstructed matrix $\mathbf{X}^{(k)} \in \mathbb{R}^{n_1 \times n_2}$ at the k^{th} iteration.

1. Set $\mathbf{Y}^{(0)} := \mathbf{0}$.
2. While $\epsilon(k)$ is greater than ϵ and $k < k_{max}$, repeat:
   i. Threshold the singular value decomposition of $\mathbf{Y}^{(k)}$ into the low-rank approximation $\mathbf{X}^{(k)}$:
   $$\mathbf{X}^{(k)} = \mathcal{D}_\tau\left(\mathbf{Y}^{(k)}\right) = \mathbf{U}^{(k)}\mathcal{D}_\tau(\mathbf{\Sigma}^{(k)})\mathbf{V}^{(k)\,\mathrm{T}}$$
   ii. Calculate the tolerance error $\epsilon(k)$:
   $$\epsilon(k) = \frac{\left\|(\mathbf{X}_{\mathbf{\Omega}}^{(k)} - \mathbf{M}_{\mathbf{\Omega}})\right\|_{\mathrm{F}}}{\|\mathbf{M}_{\mathbf{\Omega}}\|_{\mathrm{F}}}$$
   iii. Update $\mathbf{Y}^{(k+1)}$:
   $$\mathbf{Y}^{(k+1)} = \mathbf{Y}^{(k)} + \delta_k(\mathbf{M}_\Omega - \mathbf{X}_\Omega^{(k)})$$

3.1.4 Training Images Built with Low-Rank Matrix Completion

In the work by Yahya (2011) and Yahya et al. (2012), a data-driven HOS simulation method (Mustapha & Dimitrakopoulos, 2010) was combined with the soft thresholding algorithm SVT (Cai et al., 2010), reviewed in Section 3.1.3, to build a 2D TI using very sparse data (<5%). Compared to other matrix completion algorithms, such as SET (Dai et al., 2011) and OptSpace (Keshavan & Oh, 2009), SVT is a powerful and fast solver that can handle datasets with dimensions larger than 100 x 100. In order for the SVT algorithm to work, sparsity and incoherence conditions must be met (Sections 3.1.1 and 3.1.2). Additionally, the sparse matrix cannot have empty rows or columns and must have a minimum number of entries sampled uniformly at random relative to the dimensions of the matrix. The work by Yahya (2011) and Yahya et al. (2012) implemented the SVT algorithm in an approach known as MSMC, which reconstructs sparse matrices in which there are empty rows and columns placed at regular intervals. The main steps of MSMC are outlined below with a schematic example shown in Figure 12:

1. Downscale a sparse matrix by removing empty rows and columns and thereby "compressing" the size of the matrix into a smaller but higher sample density matrix;
2. Apply matrix completion using the SVT algorithm on the downscaled matrix;
3. Re-grid the matrix by reinserting missing rows and columns removed in Step 1;
4. Randomly shift a subset of the reconstructed and original entries into adjacent empty positions according to a uniform distribution. Note that a value need not necessarily shift since there is an equal probability of remaining in the same position;
5. Apply matrix completion using the SVT algorithm on the re-gridded matrix from Step 4;
6. Repeat Steps 3-5 for the number of consecutive empty rows and columns removed in Step 1.

The example in Figure 12 only has one stage since there is only one empty row and one empty column between original entries. A two-stage problem would imply two empty rows and two empty columns between original entries.

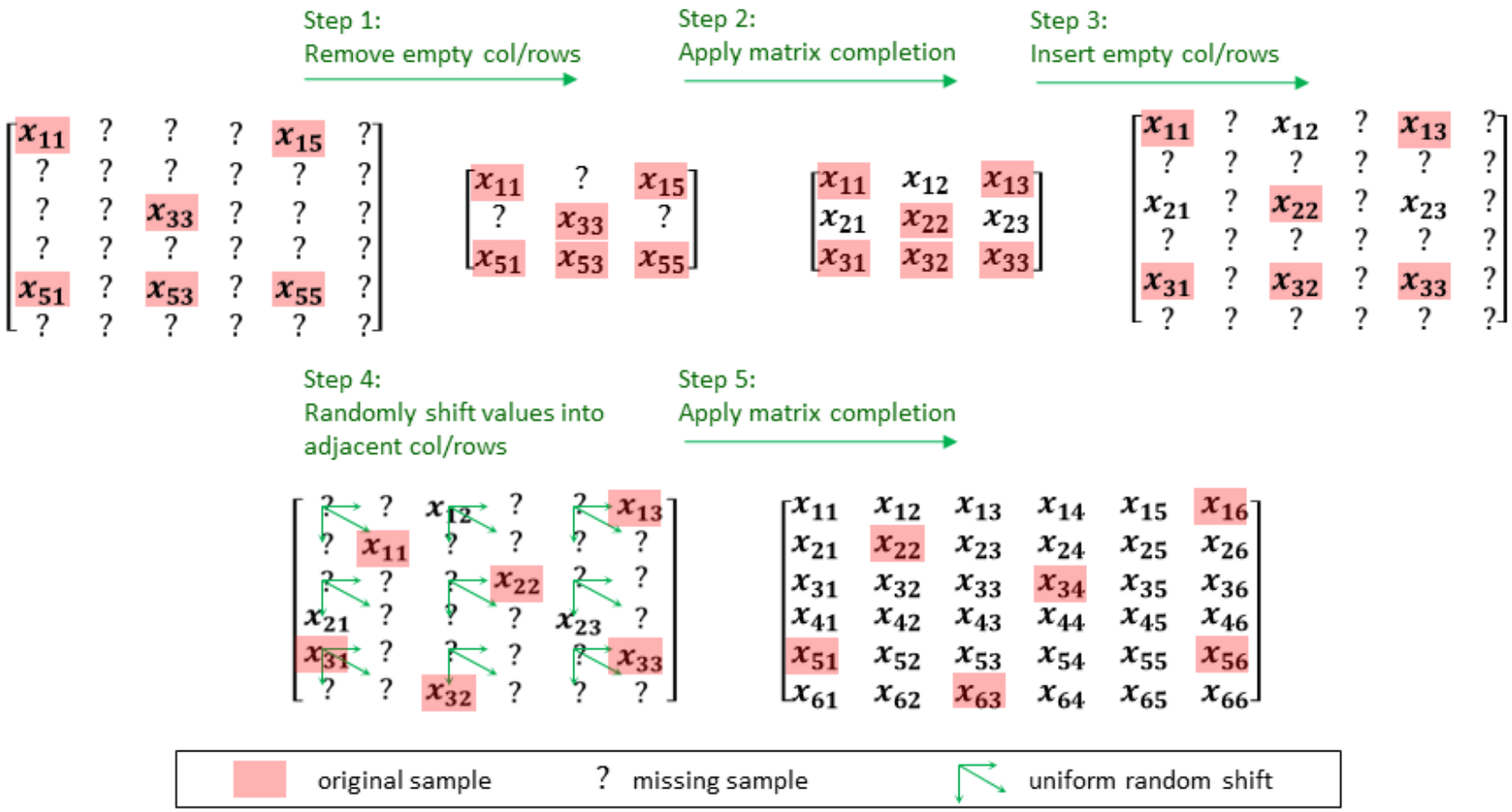

Figure 12. Main steps in MSMC using a 1-stage MSMC example.

MSMC was previously tested (Yahya, 2011; Yahya et al., 2012) on several images of varying rank: a) a chessboard, b) a woman's face (Lenna), c) a brain MRI, d) a building and e) a 2D slice taken from a 3D synthetic porosity model of reservoir channels (Mao & Journel, 1999). Images b) to d), shown in Figure 13, were compressed to a rank of 20 before testing. MSMC was able to reconstruct each of these images from 1% to 5% sample densities with very low reconstruction error when compared to a single application of matrix completion using the SVT algorithm. Table 1 lists the average root mean square error (RMSE) obtained from these experiments.

Figure 14 depicts the MSMC reconstruction of the "channels" TI using 2%, 3%, and 4% random sampling schemes. Compared to the high quality reconstructions seen in Figure 13, the lower quality reconstruction of the channels image is explained by the lower sampling density but also, and more importantly, by examining the cumulative singular values of each image. Figure 15 shows the cumulative singular values for each of the images in Figure 13, each normalized by their maximum value. The graph shows that the cumulative singular value profile gradients are steeper for 'Lenna' and 'building' than for "channels", which means that they are more compressible since they require fewer singular values to reconstruct the images. In other words, in the "channels" image the total weight of the information is spread out more evenly across singular values and thresholding these values results in greater information loss.

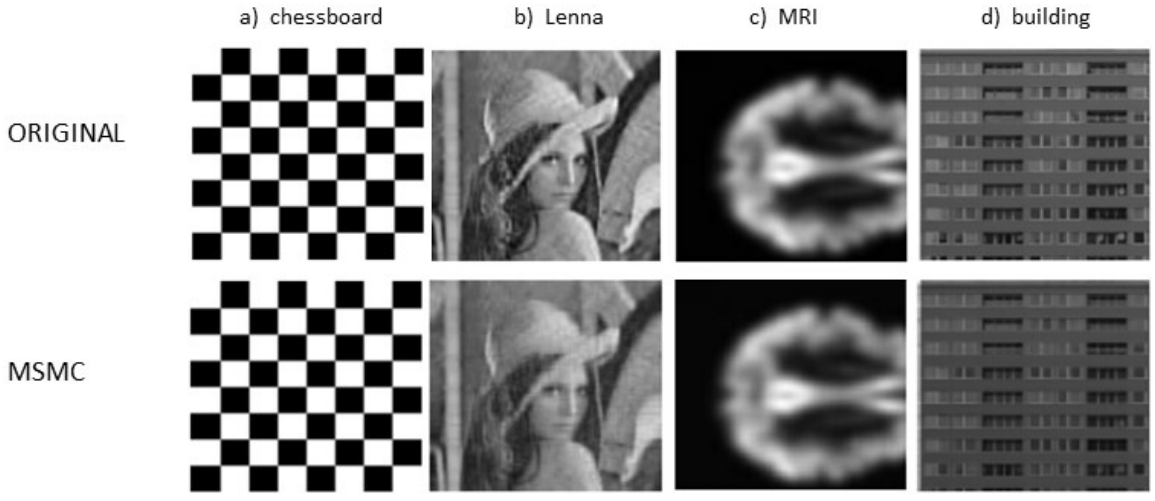

Figure 13. MSMC reconstruction of several images from 5% uniform sampling at random (Yahya, 2011).

Table 1. RMSE measured for MSMC and SSMC.

Image	Original dims.	Down-scaled dims.	Rank	Number of entries	Original image sample density	Down-scaled image sample density	SSMC error	MSMC error
Lenna, MRI, Building	240x240	60x60	20	576 – 2880	1 - 5%	16 – 80%	0.96	0.23
Channels	100x100	25x25	100	200 - 400	2 - 4%	32 – 64%	Not available	

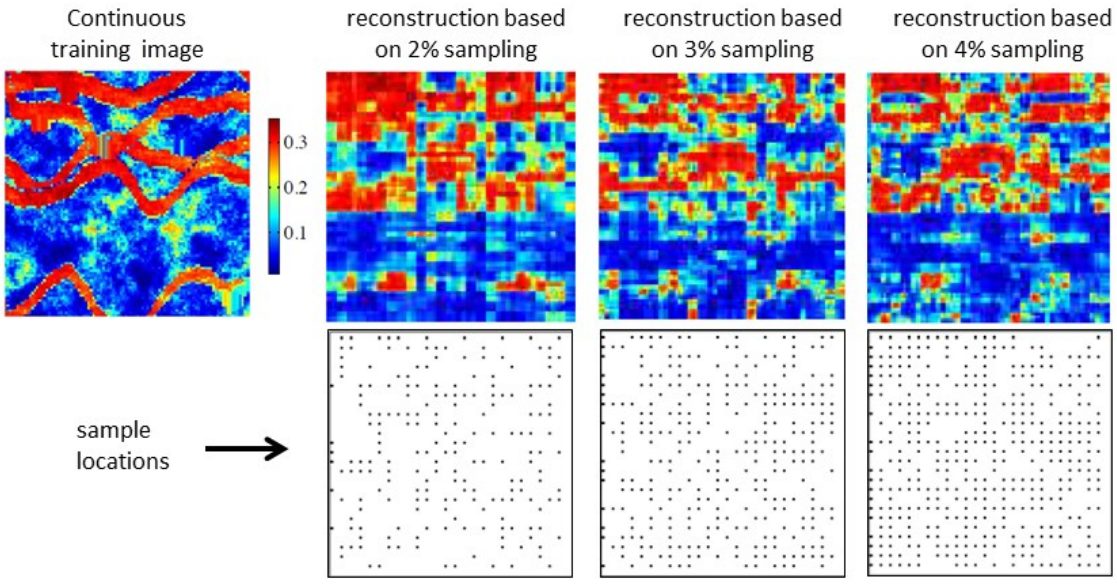

Figure 14. MSMC reconstruction of a continuous TI of reservoir channels from 2%, 3%, and 4% uniform sampling at random (Yahya, 2011).

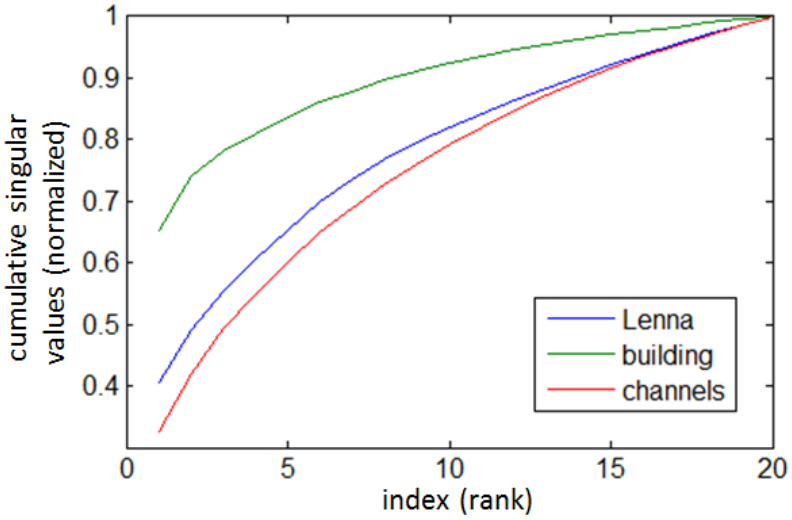

Figure 15. Normalized cumulative ratio of total singular values for three images: Lenna, buildings, and channels.

The sparse entries used in MSMC are obtained by uniform random sampling a downscaled version of the benchmark image - see Steps 1 to 3 in Figure 16. This sparse image is subsequently re-gridded by regularly inserting two empty columns and two empty rows between each column and row of the downscaled image. Table 2 summarizes the sample densities used in Yahya (2011) to reconstruct the channels TI. Figure 17 shows what a 5% randomly sampled image looks like on a downscaled version of itself. In MSMC (Yahya, 2011), the sample densities reported are based on the re-gridded image versions that include empty columns/rows. This approach works well when, in addition to the assumption that the difference between entries is small (i.e. spatial connectivity), there is an additional assumption that the number of consecutive missing entries is also small (at most 2 in Yahya (2011)).

In the current work, the idea of MSMC is modified for 3D using a tensor completion algorithm known as LRTC (Liu et al., 2013). As will be shown in Chapter 4, the MSMC approach is adapted to situations where the sampling density is less than 2%, the distances between entries are irregular, large sections of information are missing, and the TIs are both three dimensional, continuous and heterogeneously complex.

Table 2. The sampling scheme used in MSMC.

	Original TI	Downscaled TI
Dimensions	100 x 100	25 x 25
Total number of entries	10000	625
200 entries represents:	2%	32%
300 entries represents:	3%	48%
400 entries represents:	4%	64%

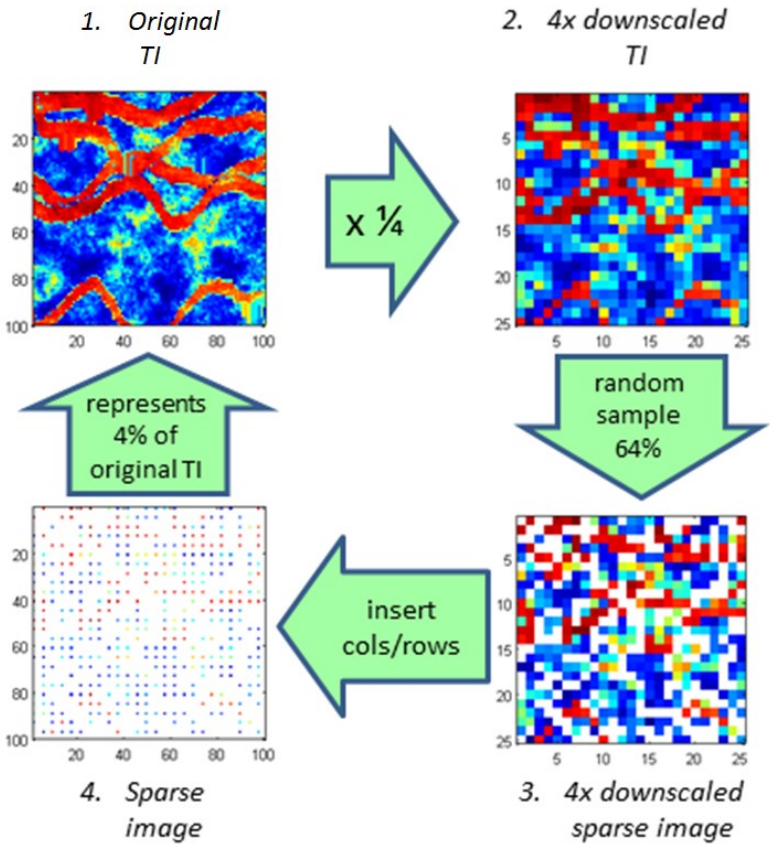

Figure 16. Schematization of the subsampling scheme devised by Yahya (2011).

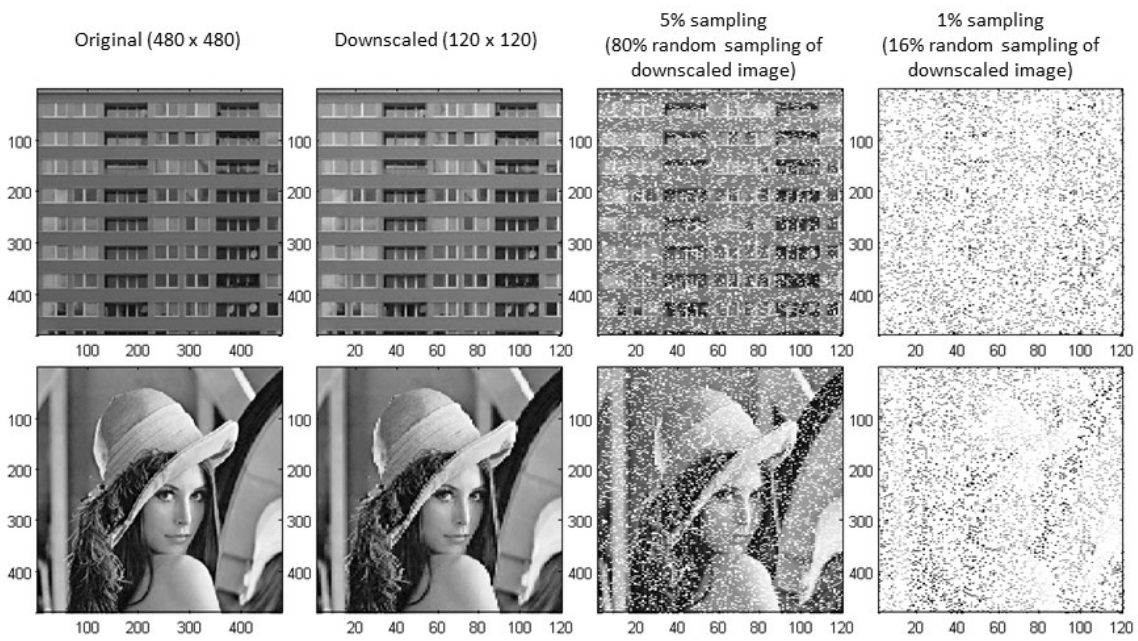

Figure 17. Random sampling of downscaled images as per the scheme used in Yahya (2011).

CHAPTER 4 TRAINING IMAGES BUILT WITH TENSOR COMPLETION

Tensor objects, tensor notation, and relevant tensor operations and properties are described at the beginning of this chapter. Section 4.2 describes the LRTC algorithm (Liu et al., 2009) used in this thesis to build 3D TIs via a method, described in Section 4.3, known as MSTC.

4.1 Tensors

Tensors are used to denote the generalization of scalars, vectors, and matrices to higher dimensions. A scalar is a zero-order tensor. Vectors and matrices are first- and second-order tensors, respectively. Volumes or "boxes" are represented by third order tensors and so on, see Figure 18.

4.1.1 Notation

A tensor with N dimensions is called an Nth-order tensor and is henceforth denoted by script notation $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, with capitalized superscript indices $I_1 \times I_2 \times \ldots I_n \times \ldots \times I_N$ to indicate the extent I_n of the tensor along each dimension n from $1, \ldots, N$. Tensors of order 3 or higher are called higher order tensors. An element of $\boldsymbol{\mathcal{T}}$ is denoted by $t_{i_1, i_2, \ldots, i_N}$ where each subscript index i_n varies from 1 to I_n. Each dimension of a tensor is known as a *mode*. A mode-n slice is generated by taking all elements along all other modes while keeping mode-n fixed. For example, there are I_3 mode-3 slices in $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, each of which is an $I_1 \times I_2$ matrix. The notation $\|\boldsymbol{\mathcal{T}}\|_F := \sqrt{\left(\sum_{i_1, i_2, \ldots, i_N} \left| t_{i_1, i_2, \ldots, i_N} \right|^2\right)}$ defines the Frobenius norm of tensor $\boldsymbol{\mathcal{T}}$**,** which is used in Chapter 5 to measure the accuracy of the TI reconstructions. This thesis only deals with 2nd and 3rd order tensors of real valued numbers.

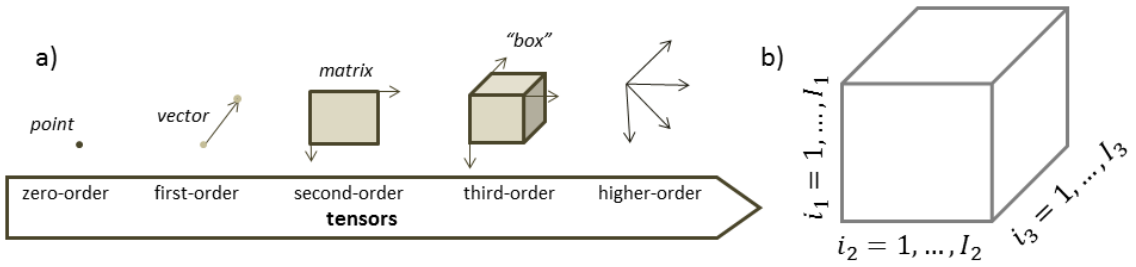

Figure 18. a) Schematic representation of tensors of different orders, b) Index representation for a third order tensor.

4.1.2 Tensor Unfolding and Folding Operations

Tensor *unfolding,* or *matricization,* is a common operation used to convert a higher order tensor into a matrix. The mode-n matricization transforms the Nth order tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ into a matrix $\mathbf{T}_{(n)} \in \mathbb{R}^{I_n \times J}$ where $J = \prod_{\substack{i=1 \\ i \neq n}}^{i=N} I_i$. This operation has an element-wise expression given by $(\mathbf{T}_{(n)})_{i_n j} = t_{i_1 i_2 \ldots i_N}$ where $j = 1 + \sum_{\substack{m=1 \\ m \neq n}}^{N} \left[(i_m - 1) \prod_{\substack{m'=1 \\ m' \neq n}}^{m} I_{m'} \right]$ indicates the column index of the matrix $\mathbf{T}_{(n)}$ (Kolda and Bader, 2009). Figure 19 illustrates this point with a 3rd order tensor. For each of the three unfoldings (mode-I, mode-K, mode-J) the two red cells in tensor $\boldsymbol{\mathcal{T}}$ map to different locations in each of the matrices depending on the unfolding direction. Alternate definitions for the mode-n matricization are possible (De Lathauwer et al., 2000; Kiers, 2000). However, the concept of changing a tensor into a matrix remains the same and tensor-matrix operations remain unaffected within a choice of convention (Kolda and Bader, 2009; Tomioka et al., 2011).

The crucial point to retain is that an Nth order tensor will have N different ways of unfolding it, each of which results in a matrix with a different arrangement of the data.

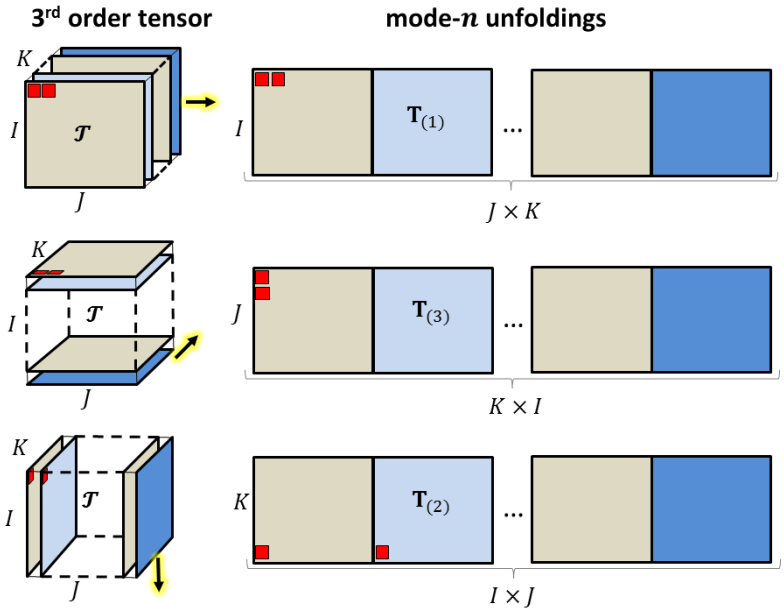

Figure 19. Matricizations of a 3rd order tensor into its three mode-n unfoldings (mode-I, mode-J, mode-K): Tensor slices are "pulled" (→) next to each other as per the colored slices.

The advantage of a matricization is that linear algebraic methods used to define structural properties of the data, such as the singular value decomposition (see **3.4** in Section 3.1.1), may be applied to unfolded tensors. However, it is well known that unfolding a tensor breaks the inherent structure and correlation of the original data, removing redundancies and higher order structures and leading to a misinterpretation of the component matrices (Kolda & Bader, 2009). For this reason, it would be useful to build a TI directly from the original dimensions of the data.

The opposite operation to matricization involves *folding* a matrix into a tensor structure. If the matrix originated from a tensor, care is required to fold the matrix using the same convention, but in reverse, as was used to unfold the tensor.

4.1.3 Ranks of a Tensor

The notion of matrix rank (i.e. the number of linearly independent column/row vectors) does not uniquely extend to tensors. Several definitions exist (Kolda & Bader, 2009) for which two relevant ones are reviewed here: the $rank(\mathcal{T})$ and the $n\text{-}ranks(\mathcal{T})$ of tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$.

The strictest analogy to the matrix rank is obtained by the minimum sum of r rank-one tensors that add to $\mathcal{T}$, where a rank-one tensor consists of the outer product $\circ$ of r vectors $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_R$ taken form N factor matrices. That is,

$$\mathcal{T} = \sum_{i=1}^{r} \boldsymbol{u}_i{}^{(1)} \circ \boldsymbol{u}_i{}^{(2)} \circ \ldots \circ \boldsymbol{u}_i{}^{(N)}. \tag{4.1}$$

where the subscript indicates the ith column vector from each of the N matrices $\mathbf{U}^{(n)} = [\boldsymbol{u}_1^{(n)}, \boldsymbol{u}_2^{(n)}, \ldots, \boldsymbol{u}_N^{(n)}]$. In this case, $rank(\mathcal{T}) = r$. This sum is easy to see for a third order tensor as shown in Figure 20. However finding $rank(\mathcal{T}) = r$ is NP-hard as there is no known algorithm to compute the decomposition in **4.1** for higher order tensors (Håstad, 1990; Kolda & Bader, 2009; Kruskal, 1989); for second-order tensors that decomposition is akin to the SVD where $\mathbf{U}^{(n)} = \{\mathbf{U}^{(1)} = \mathbf{U}, \mathbf{U}^{(2)} = \mathbf{V}\}$ for $N = 2$, such that $\mathbf{U}^{(1)}$ is the left singular matrix and $\mathbf{U}^{(2)}$ is the right singular matrix.

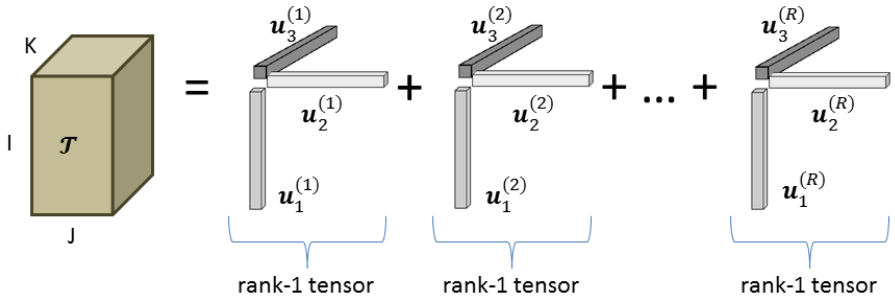

Figure 20. Decomposition of a tensor into a linear sum of rank-1 tensors. The outer product of vectors $\boldsymbol{u}_1^{(n)}, \boldsymbol{u}_2^{(n)}, \boldsymbol{u}_3^{(n)}$ is the n^{th} rank-1 tensor. The minimum number of rank-1 tensors that can produce $\mathcal{T}$ defines $rank(\mathcal{T})$.

A much simpler rank to calculate is obtained by the mode-n unfoldings of tensor $\mathcal{T}$. That is, $n\text{-}rank(\mathcal{T}) = [rank(\mathbf{T}_{(1)}), \ldots, rank(\mathbf{T}_{(n)}), \ldots, rank(\mathbf{T}_{(N)})]$ where $rank(\mathbf{T}_{(n)})$ is simply the number of linearly independent columns of the mode-n unfolded tensor $\mathbf{T}_{(n)}$ – see Figure 19. This second notion of rank is used in the LRTC algorithm (Liu et al., 2013).

4.2 The LRTC Algorithm

Tensor completion has evolved from the matrix completion literature (Candès & Recht, 2009). LRTC attempts to recover an N-dimensional tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_N}$ from

$m < n_1 n_2 \ldots n_N$ random entries, by nuclear norm minimization (see **3.3**). The algorithm is developed over three successive steps, following the work of Liu et al. (2013):

step 1: A definition of the nuclear norm for a tensor $\boldsymbol{\mathcal{X}}$,

step 2: The disambiguation of inter-connected mode-n matrices of the tensor $\boldsymbol{\mathcal{X}}$, and

step 3: The formulation of the global optimal problem based on nuclear norm minimization of all mode-n matrices of $\boldsymbol{\mathcal{X}}$.

Similarly to the matrix completion problem **3.1**, it is reasonable to consider low-rank tensor completion as the problem:

$$\begin{aligned} &\text{minimize} \quad \|\boldsymbol{\mathcal{X}}\|_* \\ &\text{subject to} \quad \boldsymbol{\mathcal{X}}_\Omega = \boldsymbol{\mathcal{T}}_\Omega \end{aligned} \tag{4.2}$$

Liu et al. (2009) define the nuclear norm of a tensor $\|\boldsymbol{\mathcal{X}}\|_*$ based on the n-ranks of the mode-n unfoldings of $\boldsymbol{\mathcal{X}}$ (see Sections 4.1.2 and 4.1.3). That is,

$$\begin{aligned} \|\boldsymbol{\mathcal{X}}\|_* &= \sum_{i=1}^{N} \alpha_i \left\|\mathbf{X}_{(i)}\right\|_* \\ \|\boldsymbol{\mathcal{X}}\|_* &= \sum_{i=1}^{N} \alpha_i \sum_{k=1}^{r} \sigma_{k_{(i)}} \left(\left\|\mathbf{X}_{(i)}\right\|_*\right) \end{aligned} \tag{4.3}$$

where $\sigma_{k_{(i)}} \left(\left\|\mathbf{X}_{(i)}\right\|_*\right)$ is the k^{th} singular value of the mode-i unfolded matrix $\mathbf{X}_{(i)}$ with rank r and α_i are constant weights that satisfy $\alpha_i \geq 0$ and $\sum_{i=1}^{N} \alpha_i = 1$, for $i = 1, \ldots, N$. Note that the tensor nuclear norm definition **4.3** encompasses the nuclear norm of a matrix given by **3.2**, i.e. $\|\boldsymbol{\mathcal{X}}\|_* = \|\mathbf{X}\|_*$ when $N = 2$ because the nuclear norm of a matrix is the same as that of its transpose. Thus, **4.2** becomes:

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{N} \alpha_i \left\|\mathbf{X}_{(i)}\right\|_* \\ &\text{subject to} \quad \boldsymbol{\mathcal{X}}_\Omega = \boldsymbol{\mathcal{T}}_\Omega \end{aligned} \tag{4.4}$$

Unlike the matrix case, in which the nuclear norm is proved to be the closest convex envelop to the matrix rank (Fazel et al., 2001), LRTC does not strictly define the tensor nuclear norm as the nearest convex envelope to the tensor rank (Romera-Paredes & Pontil, 2013). Furthermore, finding the rank of any higher order tensor remains an NP-hard problem, as there is no explicit expression for it (see Section 4.1.3) and direct minimization of the tensor rank is not possible.

Recall from Section 3.1.3 that the matrix nuclear norm minimization is solved by tackling its dual form in **3.6**. However, in the tensor case, the difficulty in optimizing problem **4.4** directly through its proximity operator (i.e. the thresholding operator) stems from the inter-dependency of the N matrix nuclear norms. That is, each mode-n unfolded matrix of $\mathcal{T}$ cannot be minimized independently of the others since they all share the same tensor elements (see Figure 19). To address this interdependency, consider **4.5** which is based on a new set of matrices $\mathbf{M}_i \cong \boldsymbol{\mathcal{X}}_{(i)}$ such that $\left\|\boldsymbol{\mathcal{X}}_{(i)} - \mathbf{M}_i\right\|_{\mathrm{F}}^2 \leq \mathrm{d}_i$:

$$\begin{aligned} &\underset{\boldsymbol{\mathcal{X}}, \mathbf{M}_i}{\text{minimize}} \sum_{i=1}^{n} \alpha_i \|\mathbf{M}_i\|_* \\ &\text{subject to} \quad \left\|\boldsymbol{\mathcal{X}}_{(i)} - \mathbf{M}_i\right\|_{\mathrm{F}}^2 \leq \mathrm{d}_i \\ &\mathrm{d}_i \geq 0 \end{aligned} \tag{4.5}$$

The convex dual formulation to **4.5** is given by

$$\begin{aligned} &\underset{\boldsymbol{\mathcal{X}}, \mathbf{M}_i}{\text{minimize}} \sum_{i=1}^{n} \alpha_i \|\mathbf{M}_i\|_* + \frac{\beta_i}{2} \left\|\boldsymbol{\mathcal{X}}_{(i)} - \mathbf{M}_i\right\|_{\mathrm{F}}^2 \\ &subject\ to \quad \boldsymbol{\mathcal{X}}_{\Omega} = \boldsymbol{\mathcal{T}}_{\Omega} \end{aligned} \tag{4.6}$$

In order to obtain a globally optimal solution, the formulation **4.6** is addressed via a block coordinate descent (BCD) optimization algorithm (Hildreth, 2006); BCD is a method that iteratively optimizes a block (a group) of variables while keeping the remaining blocks of variables fixed (Polak, 1971; Warga, 1963). In this case, the tensor

$\boldsymbol{\mathcal{X}}$ and n $\mathbf{M}_i$ matrices constitute $(n+1)$ optimization blocks, which are solved through the following two sub-problems:

Optimization of $\boldsymbol{\mathcal{X}}$

$$\underset{\boldsymbol{\mathcal{X}}}{\text{minimize}} \sum_{i=1}^{n} \frac{\beta_i}{2} \left\| \boldsymbol{\mathcal{X}}_{(i)} - \mathbf{M}_i \right\|_{\mathrm{F}}^2 \quad \text{subject to} \quad \boldsymbol{\mathcal{X}}_{\Omega} = \boldsymbol{\mathcal{T}}_{\Omega} \tag{4.7}$$

Optimization of $\mathbf{M}_i$'s

$$\underset{\mathbf{M}_i}{\text{minimize}} \sum_{i=1}^{n} \frac{\beta_i}{2} \left\| \boldsymbol{\mathcal{X}}_{(i)} - \mathbf{M}_i \right\|_{\mathrm{F}}^2 + \|\mathbf{M}_i\|_* \equiv \left\| \boldsymbol{\mathcal{X}}_{(i)} - \mathbf{M}_i \right\|_{\mathrm{F}}^2 + \frac{\alpha_i}{\beta_i} \|\mathbf{M}_i\|_* \tag{4.8}$$

Whereby the solution to (**4.7**) at the unobserved entry locations $\overline{\Omega}$ is given by

$$\boldsymbol{\mathcal{X}}_{\overline{\Omega}} = \frac{\sum_{i=1}^{N} \beta_i fold(\mathbf{M}_i)}{\sum_{i=1}^{N} \beta_i} \tag{4.9}$$

It has been shown (Cai et al., 2010) that the optimal solution to **4.8** is given by $D_\tau\left(\boldsymbol{\mathcal{X}}_{(i)}\right)$, where $\tau = \frac{\alpha_i}{\beta_i}$, which is the thresholding operator reviewed in Section 3.1.3 and the dual to the nuclear norm minimization problem **3.6** used to solve low-rank matrix completion using the SVT algorithm. The LRTC algorithm is summarized next.

Algorithm 2: LRTC

Input: Data tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, singular value thresholding parameter τ, maximum number of iterations k_{max}.

Output: Reconstructed tensor $\boldsymbol{\mathcal{X}}^{(k)} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ at the k^{th} iteration.

While $k < k_{max}$, repeat:

1. For each mode $i = 1, \ldots, N$:
   a. Calculate the mode-i unfolded matrix $\mathbf{M}_i^{(k)}$ as $\mathbf{M}_i^{(k)} = D_\tau\left(\boldsymbol{\mathcal{X}}_{(i)}^{(k)}\right)$ where $\mathbf{M}_i^{(k)}$ is thresholded into the low-rank approximation $\boldsymbol{\mathcal{X}}_{(i)}^{(k)}$:

$$\mathbf{M}_i^{(k)} = D_\tau\left(\boldsymbol{\mathcal{X}}_{(i)}^{(k)}\right) = \mathbf{U}_i^{(k)} \mathcal{D}_\tau(\boldsymbol{\Sigma}_i^{(k)}) \mathbf{V}_i^{(k)\,\mathrm{T}}$$

2. Update $\boldsymbol{\mathcal{X}}^{(k)}$ at all unobserved entry locations
   a. For each mode $i = 1, \ldots, N$:

$$\boldsymbol{\mathcal{X}}_{\overline{\Omega}}^{(k)} = \frac{\sum_{i=1}^{N} \beta_i fold(\mathbf{M}_i^{(k)})}{\sum_{i=1}^{N} \beta_i}$$

4.3 The Method of MSTC Using LRTC

This section presents the method of MSTC, which is used to reconstruct 3D continuous TIs of entries from a drill pattern that is non-uniform and non-random. The method relies on the LRTC algorithm (Liu et al., 2009) for tensor reconstruction at each stage.

Naturally, physical limitations in geological drilling make it impossible to sample the earth at depth uniformly and at random. A drill hole from surface, or underground, will be a contiguous series of entries in a straight line. From the Earth's surface this will be represented as a vertical or angled hole, while from an underground position (e.g. in an underground mine setting) it may also be horizontally directed. As a result, a drilling campaign over a deposit will exhibit many large gaps between sample points in some areas as well as highly concentrated sampling in other areas. Figure 21 presents a simplified example.

The MSTC method presents a solution for the reconstruction of 3D continuous TIs with very low and adversarial (i.e. non-random) sampling schemes where large sections of the tensor are not represented. MSTC begins with the same premise as MSMC (Yahya, 2011). That is, the reconstructed tensor is assumed to be spatially connected and initial entries are used to build the final connectivity map. MSMC removes empty columns and rows to generate less sparse matrices of varying sizes at different stages - first by removing all empty rows and columns to produce a highly sampled matrix at the first stage, and then re-introducing columns and rows previously removed to build the "multi-stage" matrices until the original matrix resolution is reached (Figure 12 on page 32). However, in that work, the sparse matrices used to test the MSMC methodology are synthetically built with exactly two empty columns and two empty rows between each represented rows and columns. That is, the number of empty rows and columns located between adjacent, represented rows and columns is always the same. In the current work, MSTC is applied to a real drilling pattern to generate the entries used for reconstruction. In real drilling patterns, it is not feasible to remove two empty slices at "every other" slice, mode-by-mode. Rotating and back rotating the dataset to remove unsampled rows and columns prior to reconstruction is prone to "edge" effects at corner locations, where no entries are present (Figure 21).

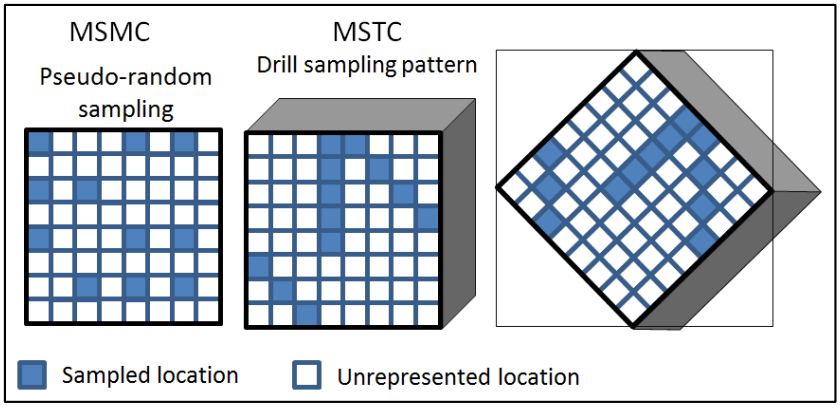

Figure 21. In MSMC (left) large gaps exist as columns and rows that can be removed. In MSTC (middle), the drilling pattern precludes the possibility of removing slices from the data as easily as in MSMC, despite large gaps. Rotating the dataset (right) to remove empty columns/rows would be prone to edge effects at corner locations.

The main steps used in MSTC are outlined next. In each step, the notation $\boldsymbol{\mathcal{X}} \xrightarrow{f} \boldsymbol{\mathcal{Y}}$ indicates the function f is used to map $\boldsymbol{\mathcal{X}}$ to $\boldsymbol{\mathcal{Y}}$. This concise notation is used in Chapter 5 to describe the different variants of MSTC implemented.

For simplicity of notation, also let

$$I' = \frac{I}{2^{S-l}}, \qquad J' = \frac{J}{2^{S-l}}, \qquad K' = \frac{K}{2^{S-l}} \tag{4.10}$$

and

$$I'' = \frac{I}{2^{S-(l+1)}}, \qquad J'' = \frac{J}{2^{S-(l+1)}}, \qquad K'' = \frac{K}{2^{S-(l+1)}} \tag{4.11}$$

4.10 and **4.11** are used to define scale transformations between tensors. Figure 22 presents a pictorial summary of MSTC. A sparse tensor $\boldsymbol{\mathcal{T}}$ is regridded to different tensors $\boldsymbol{\mathcal{G}}_0, \boldsymbol{\mathcal{G}}_1, \boldsymbol{\mathcal{G}}_2, \ldots$, each representing a different level $l = 0,1,2, \ldots$ of resolution. The regridding process is based on a random selection of subset values linked to the $\boldsymbol{\mathcal{G}}_l$ tensor. One option for a reconstruction is as follows, at $l = 1$ the tensor $\boldsymbol{\mathcal{G}}_1$ is reconstructed as $\boldsymbol{\mathcal{R}}_1$ using the LRTC algorithm. It is subsequently upsized to the next level as $\boldsymbol{\mathcal{R}}_2^{up}$, where it samples $\boldsymbol{\mathcal{G}}_2$ to fill in as many of its vacant entries as possible (without overwriting existing entries) to become $\boldsymbol{\mathcal{R}}_2^{upsample}$. Another LRTC reconstruction

is applied to $\boldsymbol{\mathcal{R}}_{\mathbf{2}}^{upsample}$ to generate $\boldsymbol{\mathcal{R}}_2$. This sequence continues until the desired resolution is reached. Examples of each function used are given in the Appendix. For reference, see summary in Table 3.

Table 3. Summary of MSTC functions.

Function	Purpose
$\boldsymbol{\mathcal{X}} \xrightarrow{\text{DOWNGRID}} \boldsymbol{\mathcal{Y}}$	Regrid $\boldsymbol{\mathcal{X}}$ to the (lower) resolution of $\boldsymbol{\mathcal{Y}}$.
$\boldsymbol{\mathcal{X}} \xrightarrow{\text{LRTC}} \boldsymbol{\mathcal{Y}}$	$\boldsymbol{\mathcal{Y}}$ is the LRTC reconstruction of $\boldsymbol{\mathcal{X}}$.
$\boldsymbol{\mathcal{X}} \xrightarrow{\text{UPSIZE}} \boldsymbol{\mathcal{Y}}$	Regrid $\boldsymbol{\mathcal{X}}$ to the (higher) resolution of $\boldsymbol{\mathcal{Y}}$ by inserting empty slices along each mode.
$\boldsymbol{\mathcal{X}} \xrightarrow{\text{RANDUP}} \boldsymbol{\mathcal{Y}}$	Regrid $\boldsymbol{\mathcal{X}}$ to the (higher) resolution of $\boldsymbol{\mathcal{Y}}$ by inserting empty slices along each mode and randomly shifting values.
$\boldsymbol{\mathcal{X}} \xrightarrow{\text{CORRELUP}} \boldsymbol{\mathcal{Y}}$	Regrid $\boldsymbol{\mathcal{X}}$ to the (higher) resolution of $\boldsymbol{\mathcal{Y}}$ by inserting empty slices along each mode and shifting values towards other similar values.
$\boldsymbol{\mathcal{X}} \xrightarrow{\text{UPSAMPLE}\,\mathcal{G}_l} \boldsymbol{\mathcal{Y}}$	Infill empty locations in $\boldsymbol{\mathcal{X}}$ by sampling values from $\boldsymbol{\mathcal{G}}_l$.

MSTC steps:

1. Map the original sample data $\boldsymbol{\mathcal{T}}$ into the lower resolution tensors $\boldsymbol{\mathcal{G}}_l$, $l = 0, \ldots, S$, where S defines the number of multi-stages in the reconstruction and $\boldsymbol{\mathcal{G}}_S = \boldsymbol{\mathcal{T}}$.

$$\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I \times J \times K} \xrightarrow{\text{DOWNGRID}} \boldsymbol{\mathcal{G}}_l \in \mathbb{R}^{I' \times J' \times K'} \tag{4.12}$$

2. Stage reconstruction using the LRTC algorithm to reconstruct $\boldsymbol{\mathcal{G}}_l$ as $\boldsymbol{\mathcal{R}}_l$ for $l > 0$ (there is no stage-0 reconstruction):

$$\boldsymbol{\mathcal{G}}_l \in \mathbb{R}^{I' \times J' \times K'} \xrightarrow{\text{LRTC}} \boldsymbol{\mathcal{R}}_l \in \mathbb{R}^{I' \times J' \times K'} \tag{4.13}$$

3. Upsize $\boldsymbol{\mathcal{R}}_l$ to the next stage $(l+1)$ by increasing its resolution in one of three ways:
    a. Insert one empty slice next to each mode-m slice, where $m \in \{1,2,3\}$:

$$\boldsymbol{\mathcal{R}}_l \in \mathbb{R}^{I' \times J' \times K'} \xrightarrow{\text{UPSIZE}} \boldsymbol{\mathcal{R}}_{l+1}^{up} \in \mathbb{R}^{I'' \times J'' \times K''} \quad \textbf{(4.14)}$$

b. Insert one empty slice next to each mode-m slice and randomly shift the values. This step ensures the representation of all slices (i.e. no slices are left empty) along all modes of the tensor.

$$\boldsymbol{\mathcal{R}}_l \in \mathbb{R}^{I' \times J' \times K'} \xrightarrow{\text{RANDUP}} \boldsymbol{\mathcal{R}}_{l+1}^{randup} \in \mathbb{R}^{I'' \times J'' \times K''} \quad \textbf{(4.15)}$$

c. Insert one empty slice next to each mode-m slice and shift the values in a direction that correlates with values from a previous stage that has been $regridded$ to the same resolution - see Step 4.

$$\boldsymbol{\mathcal{R}}_l \in \mathbb{R}^{I' \times J' \times K'} \xrightarrow{\text{CORRELUP}} \boldsymbol{\mathcal{R}}_{l+1}^{correlup} \in \mathbb{R}^{I'' \times J'' \times K''} \quad \textbf{(4.16)}$$

4. For a given stage $l > 0$, sample the $downgridded$ tensor $\boldsymbol{\mathcal{G}}_l$ at locations of $\boldsymbol{\mathcal{R}}_l^{up}$,$\boldsymbol{\mathcal{R}}_l^{randup}$,$\boldsymbol{\mathcal{R}}_l^{correlup}$ with missing entries. Any number of tensors $\boldsymbol{\mathcal{G}}_l$ may be sampled as long as they have been $regridded$ to the dimensions of the current stage:

$$\boldsymbol{\mathcal{R}}_l^{up} \in \mathbb{R}^{I' \times J' \times K'} \xrightarrow{\text{UPSAMPLE}\,\boldsymbol{\mathcal{G}}_l} \boldsymbol{\mathcal{R}}_l^{upsample} \in \mathbb{R}^{I' \times J' \times K'} \quad \textbf{(4.17)}$$

$$\boldsymbol{\mathcal{R}}_l^{randup} \in \mathbb{R}^{I' \times J' \times K'} \xrightarrow{\text{UPSAMPLE}\,\boldsymbol{\mathcal{G}}_l} \boldsymbol{\mathcal{R}}_l^{upsample} \in \mathbb{R}^{I' \times J' \times K'} \quad \textbf{(4.18)}$$

$$\boldsymbol{\mathcal{R}}_l^{correlup} \in \mathbb{R}^{I' \times J' \times K'} \xrightarrow{\text{UPSAMPLE}\,\boldsymbol{\mathcal{G}}_l} \boldsymbol{\mathcal{R}}_l^{upsample} \in \mathbb{R}^{I' \times J' \times K'} \quad \textbf{(4.19)}$$

5. Repeat steps 3 to 5 up to $l \leq S$.

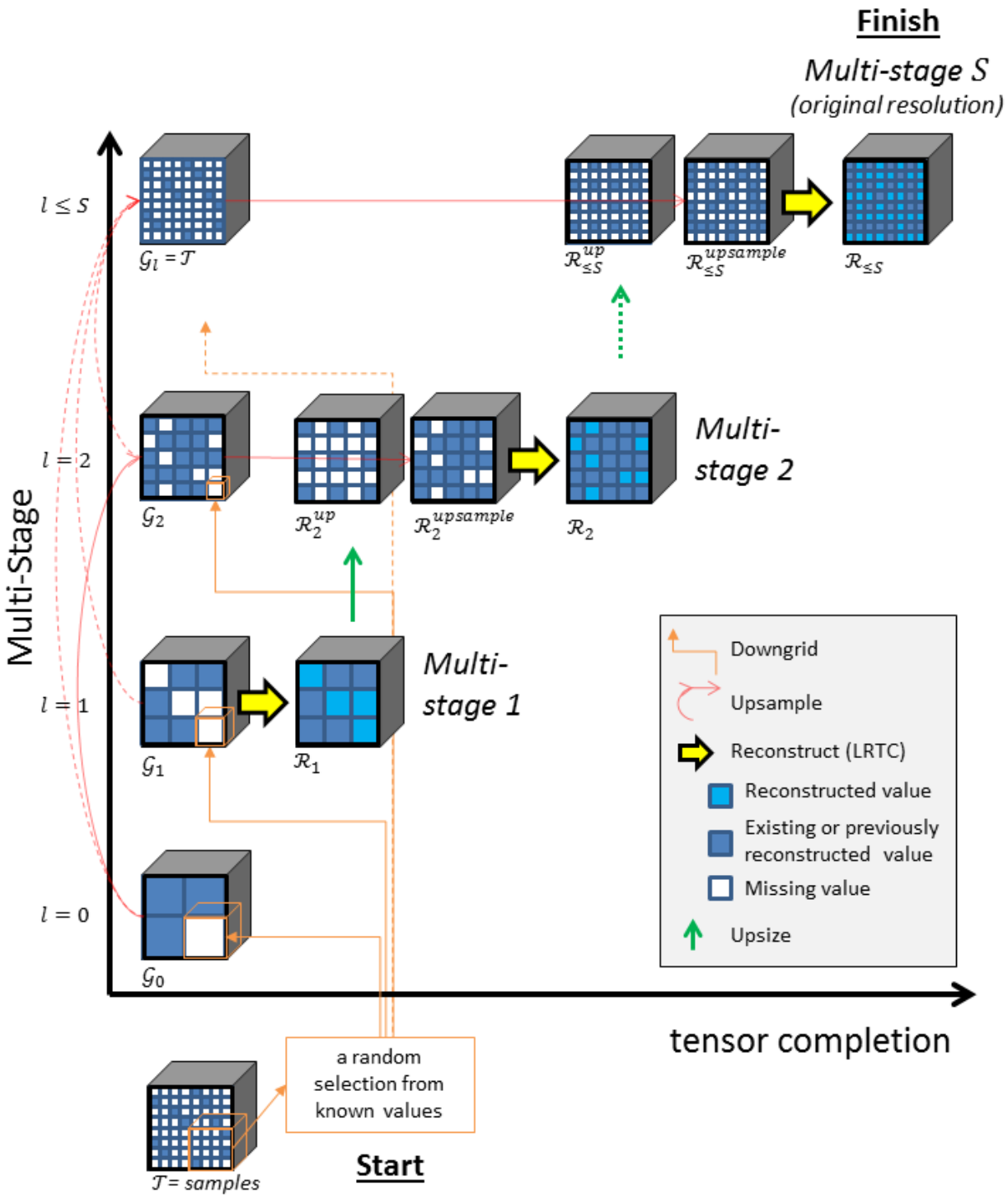

Figure 22. Method of MSTC showing the progressive reconstruction of a sparse tensor $\mathcal{T}$ up to multi-stage $l \leq S$.

4.3.1 Number of Stages

Similarly to MSMC (Yahya, 2011), the number of stages S is defined by the number of times a reconstruction occurs. While the number of stages in MSMC is defined by the number of consecutive empty rows/columns between represented rows/columns, the number of stages in MSTC is a user-defined parameter based on the desired "coarseness" and accuracy of the reconstruction.

4.3.2 The MSTC Algorithm

The previous section provided the building blocks for the MSTC algorithm. The pseudocode for MSTC is shown in Algorithm 3.

Algorithm 3: MSTC

Input: Sparse tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, number of multi-stages S, $regridding$ parameter vector $uscaleVec$ (refer to Appendix), LRTC parameters τ and k_{max}.

Output: Multi-stage reconstruction $\boldsymbol{\mathcal{R}}_{\leq S} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ at original resolution, or lower.

1. For $l = 0\ to\ S$
    a. Build low resolution tensors: $\boldsymbol{\mathcal{T}} \xrightarrow{\text{DOWNGRID}} \boldsymbol{\mathcal{G}}_l$
2. For $l = 1\ to\ S$
    a. Do a reconstruction: $\boldsymbol{\mathcal{G}}_l \xrightarrow{\text{LRTC}(\tau, k_{max})} \boldsymbol{\mathcal{R}}_l$ using Algorithm 2
    b. Upsize the reconstruction by one of three methods:
        i. $\boldsymbol{\mathcal{R}}_l \xrightarrow{\text{UPSIZE}(uscaleVec)} \boldsymbol{\mathcal{R}}^{up}_{l=l+1}$
        ii. $\boldsymbol{\mathcal{R}}_l \xrightarrow{\text{RANDUP}(uscaleVec)} \boldsymbol{\mathcal{R}}^{up}_{l=l+1}$
        iii. $\boldsymbol{\mathcal{R}}_l \xrightarrow{\text{CORRELUP}(uscaleVec)} \boldsymbol{\mathcal{R}}^{up}_{l=l+1}$
    c. Sample information from another tensor (optional)
        $\boldsymbol{\mathcal{R}}_l^{(method\ used\ in\ Step\ 2)} \xrightarrow{\text{UPSAMPLE}\boldsymbol{\mathcal{G}}_l} \boldsymbol{\mathcal{R}}_l^{upsample}$

By only allowing certain modes to $upsize$ or by introducing an arbitrary number of empty slices per mode during the $upsizing$ (Step 3), a different MSTC strategy is possible. Variants of the MSTC method based on combinations of functions listed in Steps 3 and 4 (Section 4.3) are presented in Chapter 5 using Algorithm 3 and two benchmark TIs.

CHAPTER 5 MSTC IMPLEMENTATION, RESULTS AND ANALYSIS

Implementation details and results obtained from the application of MSTC on two benchmark sets are presented and discussed in this section. The benchmark TIs pertain to a 3D dataset known as Stanford V. The Stanford V reservoir model is an exhaustive synthetic dataset for general purpose testing of various aspects pertaining to petroleum reservoir modeling and characterization (Mao and Journel, 1999), including the application of MPS and HOS simulation algorithms. The model is populated with various co-located continuous petrophysical properties, including porosity. A reconstruction relative standard error $RSE = \frac{\|\mathcal{R}_{\leq S} - \mathcal{B}\|_F}{\|\mathcal{B}\|_F}$, where $\mathcal{R}_{\leq S}$ is the reconstructed TI and $\mathcal{B}$ is the exhaustive benchmark TI is used to measure the quality of the TI reconstructions. Low RSE results from the application of MSTC on the two benchmark models could be used to choose the most suitable MSTC method to apply on the reconstruction of a TI for a real mining dataset. A description of the benchmark models follows, before specifying the particulars of the MSTC implementation.

5.1 Benchmark Model Testing

5.1.1 Description of Continuous Benchmark Models

The 3D benchmark data used are continuous porosity values arranged in sets of sinusoidal channels. The dataset has been separated into two main categories to reflect the complexity of the channels and to test the MSTC methodology against that complexity: one set for simple geometries and a second for complicated arrangements of sinusoidal channels. Each set has channels running mostly orthogonally to the other set. Figures 23 and 24 present these two porosity datasets. Both figures have been arranged into five panels (A, B, C, D, E) that contain the following information:

1. Each of the insets A, B, and C contain eight X-Y slices of a 3D TI, with XYZ dimensions given by 56 x 112 x 8, constructed from the Stanford V model. The stacking order of the slices within a given inset should be read left-to-right first, then top-to-bottom.

i. A is a model in which each slice is identical – this is the simplest of all models.
ii. B is a model in which every other slice is different, with the remaining in-between slices simply a copy of the previous slice.
iii. C is a model in which each slice is different.

2. Inset D contains the de-clustered drill pattern (slice-by-slice pierce-points) pertaining to a real deposit; this pattern is used to sample each model (A, B, C). The drilling pattern consists of angled holes dipping approximately 50 degrees on a 270^{o} azimuth. The entries shown represent approximately 2% of any 3D TI shown.
3. Inset E contains the distribution of porosities from the data in insets A and D (i.e. where D represents entries from A). By visual inspection of both benchmark sets, the sample distribution curve is a close match to the TI distribution curve, despite the relatively low number of entries. Although not shown, a similar match was noticed for models B and C.

A brief summary of the aforementioned description is listed in Table 4.

Table 4. Description of benchmark 3D TIs in each of two sets.

3D TI Benchmark Sets	Model A	Model B	Model C
Set 1: Simple geometries	All layers in the model are the same.	Intra-pair layers are the same; Inter-pair layers are different.	All layers in the model are different.
Set 2: Complex geometries			

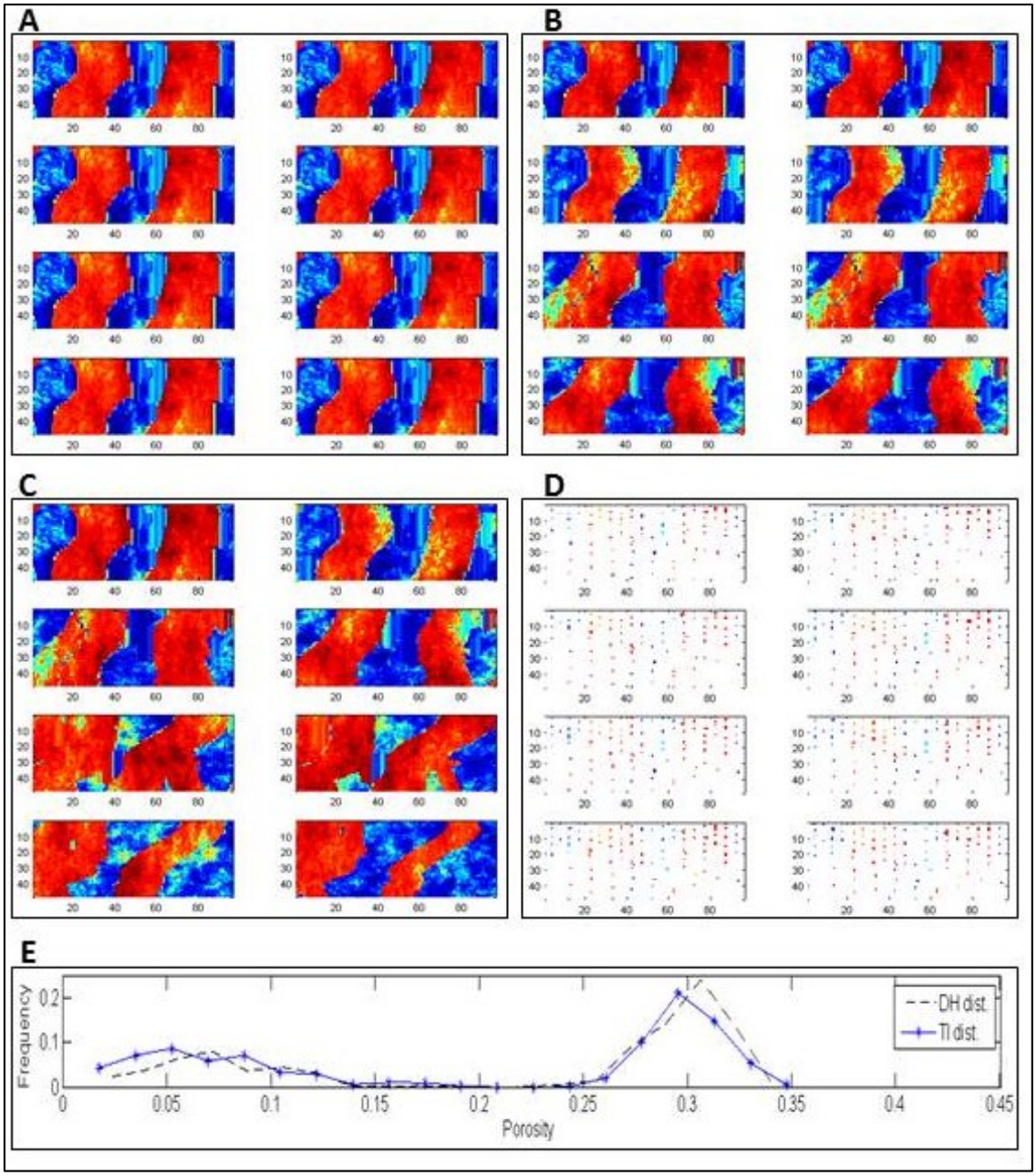

Figure 23. Benchmark 3D TIs in set 1: simple geometries. See Section 5.1.1 for figure description.

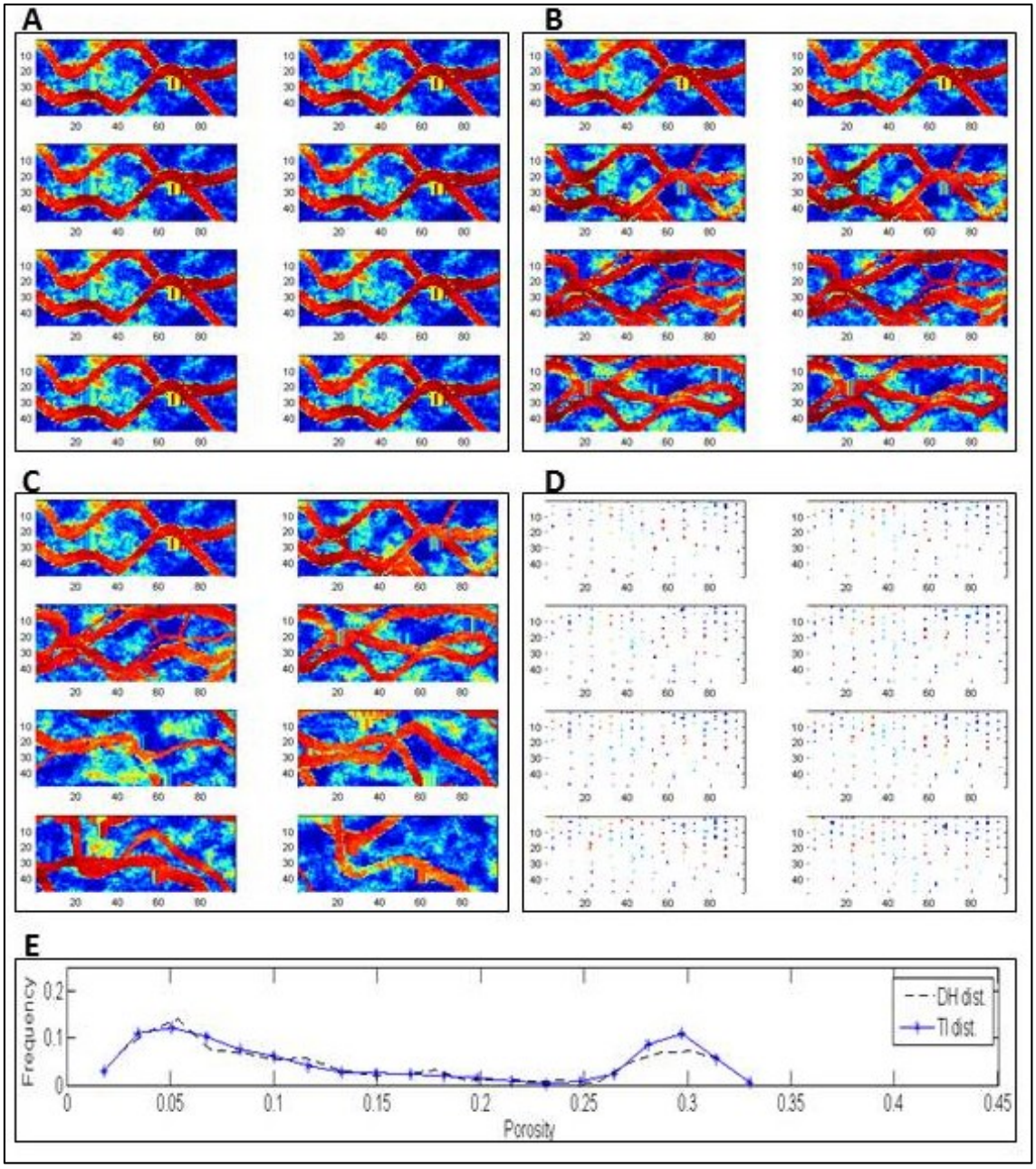

Figure 24. Benchmark 3D TIs in set 2: complex geometries. See Section 5.1.1 for figure description.

5.1.2 Description of the Sampling Pattern Used

From the insets D in Figures 23 and 24, it is clear that large data gaps exist and that sample points are not distributed uniformly throughout the slices; the density of entries is sparser along the bottom left of each slice. By comparison, recall that the MSMC (Yahya, 2011) method relies on a uniformly random pattern with constant empty

rows/columns between entries – see Figure 14. Furthermore, the samples are connected from slice to slice in a drilling pattern to form strings of points.

5.1.3 Implementation of MSTC on Continuous Benchmark Models

In Section 4.3, the basic steps for carrying out MSTC were presented. MSTC may be implemented along different sequences depending on the method used to increase the size of a reconstructed tensor (e.g. $randup$) and the number of tensors $\mathcal{G}_l$ to reference when sampling for additional information (i.e. $upsampling$). Five main variants of MSTC were implemented as per descriptions 1) to 5) below and the notation developed in Section 4.3 (a functional notation is summarized in Table 3 while detailed examples are included in the Appendix). Each variant was run using four stages, i.e. $S = 4$. In each case, the LRTC algorithm parameters were set to $k_{max} = 5000$ and $\tau = 0.1$. These parameters were chosen based on a sensitivity analysis performed over several reconstructions of a sub-tensor of the Stanford V porosity dataset. Results from the sensitivity analysis are shown in Figure 25. The dimensions of the initial sparse tensor $\mathcal{T}$ and its final reconstruction $\mathcal{R}_4$ are the same, given by 56 x 112 x 8.

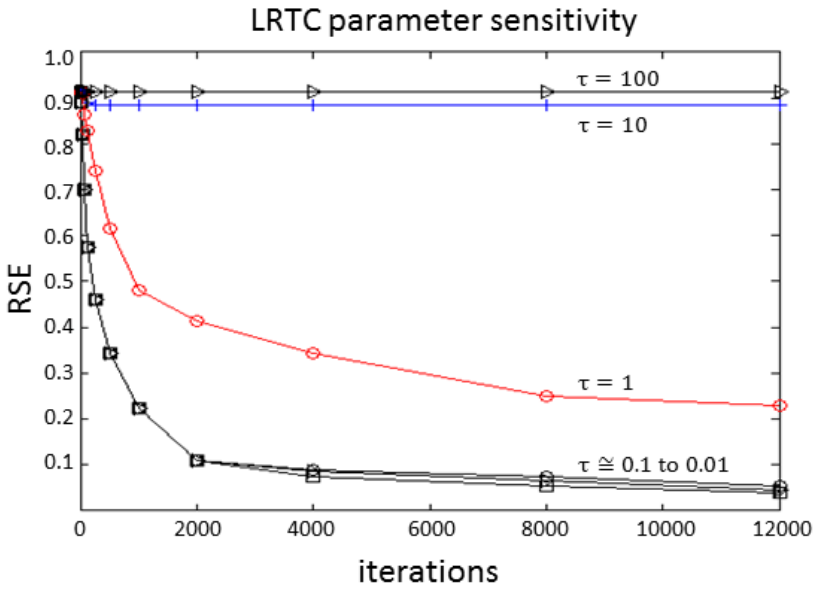

Figure 25. Parameter sensitivity analysis used to choose an appropriate τ value.

The first variant tested, MSTC-1, simply reconstructs tensors which have been upsized. The entire MSTC sequence is captured by **5.1**. The notation used is that which was described in Section 4.3 and through equations **4.12** to **4.19**:

$$\begin{aligned}\boldsymbol{\mathcal{T}} \in \mathbb{R}^{56\times 112\times 8} &\xrightarrow{\text{DOWNGRID}} \boldsymbol{\mathcal{G}}_{l=1} \in \mathbb{R}^{7\times 14\times 1} \xrightarrow{\text{LRTC}} \boldsymbol{\mathcal{R}}_1 \in \mathbb{R}^{7\times 14\times 1} \\ &\xrightarrow{\text{UPSIZE}} \boldsymbol{\mathcal{R}}_2^{up} \in \mathbb{R}^{14\times 28\times 2} \xrightarrow{\text{LRTC}} \boldsymbol{\mathcal{R}}_2 \in \mathbb{R}^{14\times 28\times 2} \\ &\xrightarrow{\text{UPSIZE}} \boldsymbol{\mathcal{R}}_3^{up} \in \mathbb{R}^{28\times 56\times 4} \xrightarrow{\text{LRTC}} \boldsymbol{\mathcal{R}}_3 \in \mathbb{R}^{28\times 56\times 4} \\ &\xrightarrow{\text{UPSIZE}} \boldsymbol{\mathcal{R}}_4^{up} \in \mathbb{R}^{56\times 112\times 8} \xrightarrow{\text{LRTC}} \boldsymbol{\mathcal{R}}_4 \in \mathbb{R}^{56\times 112\times 8}\end{aligned} \quad (\mathbf{5.1})$$

The second variant (MSTC-2) reconstructs tensors that have been upsized and had their values randomly shifted, towards an adjacent node. **5.2** summarizes the MSTC sequence:

$$\begin{aligned}\boldsymbol{\mathcal{T}} \in \mathbb{R}^{56\times 112\times 8} &\xrightarrow{\text{DOWNGRID}} \boldsymbol{\mathcal{G}}_{l=0,1,2} \in \mathbb{R}^{7\times 14\times 1} \xrightarrow{\text{LRTC}} \boldsymbol{\mathcal{R}}_1 \in \mathbb{R}^{7\times 14\times 1} \\ &\xrightarrow{\text{RANDUP}} \boldsymbol{\mathcal{R}}_2^{up} \in \mathbb{R}^{14\times 28\times 2} \xrightarrow{\text{LRTC}} \boldsymbol{\mathcal{R}}_2 \in \mathbb{R}^{14\times 28\times 2} \\ &\xrightarrow{\text{RANDUP}} \boldsymbol{\mathcal{R}}_3^{up} \in \mathbb{R}^{28\times 56\times 4} \xrightarrow{\text{LRTC}} \boldsymbol{\mathcal{R}}_3 \in \mathbb{R}^{28\times 56\times 4} \\ &\xrightarrow{\text{RANDUP}} \boldsymbol{\mathcal{R}}_4^{up} \in \mathbb{R}^{56\times 112\times 8} \xrightarrow{\text{LRTC}} \boldsymbol{\mathcal{R}}_4 \in \mathbb{R}^{56\times 112\times 8}\end{aligned} \quad (\mathbf{5.2})$$

MSTC-3 represents a combination, whereby tensors are first upsized, followed by a sampling of values from another tensor in the multi-stage sequence. These steps are summarized in **5.3**:

$$\begin{aligned}\boldsymbol{\mathcal{T}} \in \mathbb{R}^{56\times 112\times 8} &\xrightarrow{\text{DOWNGRID}} \boldsymbol{\mathcal{G}}_{l=0,1,2} \in \mathbb{R}^{7\times 14\times 1} \xrightarrow{\text{LRTC}} \boldsymbol{\mathcal{R}}_1 \in \mathbb{R}^{7\times 14\times 1} \\ \xrightarrow{\text{UPSIZE}} \boldsymbol{\mathcal{R}}_2^{up} \in \mathbb{R}^{14\times 28\times 2} &\xrightarrow{\text{UPSAMPLE } \boldsymbol{\mathcal{G}}_1} \boldsymbol{\mathcal{R}}_2^{upsample} \in \mathbb{R}^{14\times 28\times 2} \xrightarrow{\text{LRTC}} \boldsymbol{\mathcal{R}}_2 \in \mathbb{R}^{14\times 28\times 2} \\ \xrightarrow{\text{UPSIZE}} \boldsymbol{\mathcal{R}}_3^{up} \in \mathbb{R}^{28\times 56\times 4} &\xrightarrow{\text{UPSAMPLE } \boldsymbol{\mathcal{G}}_2} \boldsymbol{\mathcal{R}}_3^{upsample} \in \mathbb{R}^{28\times 56\times 4} \xrightarrow{\text{LRTC}} \boldsymbol{\mathcal{R}}_3 \in \mathbb{R}^{28\times 56\times 4} \\ \xrightarrow{\text{UPSIZE}} \boldsymbol{\mathcal{R}}_4^{up} \in \mathbb{R}^{56\times 112\times 8} &\xrightarrow{\text{UPSAMPLE } \boldsymbol{\mathcal{G}}_4=\boldsymbol{\mathcal{T}}} \boldsymbol{\mathcal{R}}_4^{upsample} \in \mathbb{R}^{56\times 112\times 8} \xrightarrow{\text{LRTC}} \boldsymbol{\mathcal{R}}_4 \in \mathbb{R}^{56\times 112\times 8}\end{aligned} \quad (\mathbf{5.3})$$

MSTC-4, summarized by **5.4**, is based on MSTC-3 with the exception that an additional tensor is sampled in order to help reduce the number of missing entries prior to LRTC reconstruction. A hidden function $regrid$ (see Appendix) is used to upsize the second tensor to the correct resolution.

$$\begin{aligned}
&\mathcal{T} \in \mathbb{R}^{56\times 112\times 8} \xrightarrow{\text{DOWNGRID}} \mathcal{G}_{l=0,1,2} \in \mathbb{R}^{7\times 14\times 1} \xrightarrow{\text{LRTC}} \mathcal{R}_1 \in \mathbb{R}^{7\times 14\times 1} \\
&\xrightarrow{\text{UPSIZE}} \mathcal{R}_2^{up} \in \mathbb{R}^{14\times 28\times 2} \xrightarrow{\text{UPSAMPLE } \mathcal{G}_1,\mathcal{G}_0} \mathcal{R}_2^{upsample} \in \mathbb{R}^{14\times 28\times 2} \xrightarrow{\text{LRTC}} \mathcal{R}_2 \in \mathbb{R}^{14\times 28\times 2} \\
&\xrightarrow{\text{UPSIZE}} \mathcal{R}_3^{up} \in \mathbb{R}^{28\times 56\times 4} \xrightarrow{\text{UPSAMPLE } \mathcal{G}_2,\mathcal{G}_1} \mathcal{R}_3^{upsample} \in \mathbb{R}^{28\times 56\times 4} \xrightarrow{\text{LRTC}} \mathcal{R}_3 \in \mathbb{R}^{28\times 56\times 4} \\
&\xrightarrow{\text{UPSIZE}} \mathcal{R}_4^{up} \in \mathbb{R}^{56\times 112\times 8} \xrightarrow{\text{UPSAMPLE } \mathcal{G}_4=\mathcal{T},\mathcal{G}_3} \mathcal{R}_4^{upsample} \in \mathbb{R}^{56\times 112\times 8} \xrightarrow{\text{LRTC}} \mathcal{R}_4 \in \mathbb{R}^{56\times 112\times 8}
\end{aligned} \tag{5.4}$$

MSTC-5 represents the last variant tried, summarized in **5.5**, which is based on shifting values, towards an adjacent node which is in a direction of maximum correlation with adjacent entries from a previous stage tensor.

$$\begin{aligned}
&\mathcal{T} \in \mathbb{R}^{56\times 112\times 8} \xrightarrow{\text{DOWNGRID}} \mathcal{G}_{l=0,1,2} \in \mathbb{R}^{7\times 14\times 1} \xrightarrow{\text{LRTC}} \mathcal{R}_1 \in \mathbb{R}^{7\times 14\times 1} \\
&\xrightarrow{\text{CORRUP}} \mathcal{R}_2^{up} \in \mathbb{R}^{14\times 28\times 2} \xrightarrow{\text{UPSAMPLE } \mathcal{G}_1,} \mathcal{R}_2^{upsample} \in \mathbb{R}^{14\times 28\times 2} \xrightarrow{\text{LRTC}} \mathcal{R}_2 \in \mathbb{R}^{14\times 28\times 2} \\
&\xrightarrow{\text{CORRUP}} \mathcal{R}_3^{up} \in \mathbb{R}^{28\times 56\times 4} \xrightarrow{\text{UPSAMPLE } \mathcal{G}_2} \mathcal{R}_3^{upsample} \in \mathbb{R}^{28\times 56\times 4} \xrightarrow{\text{LRTC}} \mathcal{R}_3 \in \mathbb{R}^{28\times 56\times 4} \\
&\xrightarrow{\text{CORRUP}} \mathcal{R}_4^{up} \in \mathbb{R}^{56\times 112\times 8} \xrightarrow{\text{UPSAMPLE } \mathcal{G}_4=\mathcal{T}} \mathcal{R}_4^{upsample} \in \mathbb{R}^{56\times 112\times 8} \xrightarrow{\text{LRTC}} \mathcal{R}_4 \in \mathbb{R}^{56\times 112\times 8}
\end{aligned} \tag{5.5}$$

5.1.4 MSTC Reconstruction Results

Reconstruction error (RSE) results obtained for each MSTC variant applied over all models are summarized in Figure 26. The performance of the MSTC variants applied to the first benchmark set compare similarly to the results obtained in the second set. This is encouraging, as it may suggest that the better MSTC methods are so regardless of model complexity. For models in benchmark set 1 it is easy to see that the reconstruction error is lower for model A and highest for model C, which corresponds with the complexity of each model (Table 4). For the second benchmark set, it's difficult

to see why the reconstruction of model B outperforms that of model A when using MSTC-1, MSTC-2, and MSTC-3; this is unexpected since model A is a simpler model.

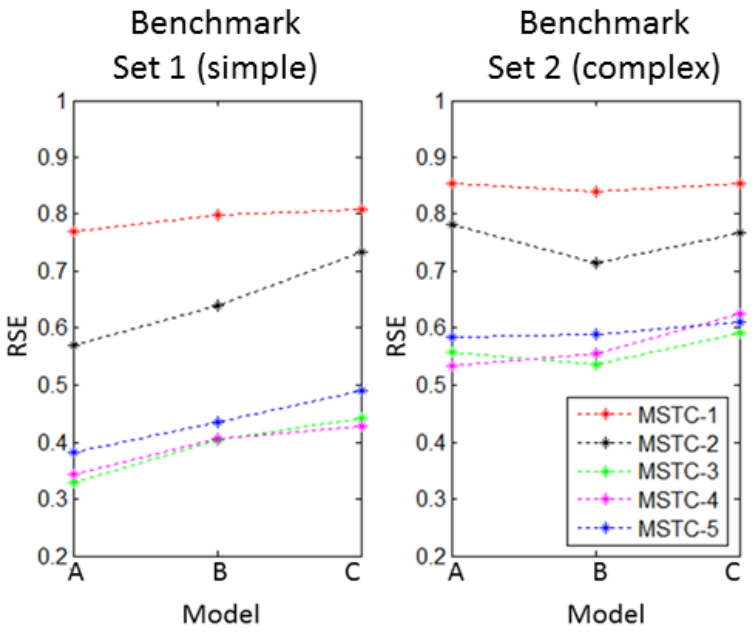

Figure 26. RSE results using benchmark datasets.

The lowest error performance, as measured by RSE and visually, comes from MSTC-4. Step-by-step reconstruction results of model C and A in benchmark sets 1 and 2, using MSTC-4 are shown in Figures 27 and 28, respectively. In each of the two figures, row 1 is the benchmark TI re-gridded to different stage resolutions, row 2 represents the same re-gridding but done on the sampled entries, and in row 3 are the reconstruction results for each stage in the multi-stage sequence. The histograms in row 4 compare the distributions of the data in each of the first three rows (i.e. benchmark model, entries, and reconstruction), while row 5 is a direct representation of the $upsize$ function, before any shifting (random or correlated) or sampling of values. The sampling of values along unobserved entries of images in row 5 is shown in row 6. The data in row 6 represent the input for an LRTC reconstruction, after which the process begins again at the next column to the right. The final reconstruction at the original scale resolution is located at the far right along the third row. Immediately above it are the original entries and the benchmark model.

Figures 29 and 30 present reconstructions for MSTC variants 1-5 (rows a-e) applied to each of the three models (A, B, C) for simple and complex cases, respectively. For simplification, the same slice using the same colour table is always shown in all models. It is interesting to see that the third best reconstruction occurs for MSTC-5, which is generated by shifting values in a direction of maximum correlation with a tensor from a previous stage. The worst reconstruction occurs when simply upsizing and attempting a reconstruction using LRTC. This occurs because the upsizing action introduces too many missing entries, some of which form entire empty slices; over a successive number of multi-stages the effect grows worse. Random shifting of values (MSTC-2) does not work very well in MSTC with a number of stages $S > 2$. That is, the combined effect of upsizing and randomizing values effectively breaks down structural features when performed over several stages.

The noticeable “blockiness” of the reconstruction when using MSTC-4 presents a negative side-effect of sampling a tensor that originates from a lower stage; this effect is not as present in the reconstructions from MSTC-3 and MSTC-5, but these in turn do not respect the data statistics as well as MSTC-4.

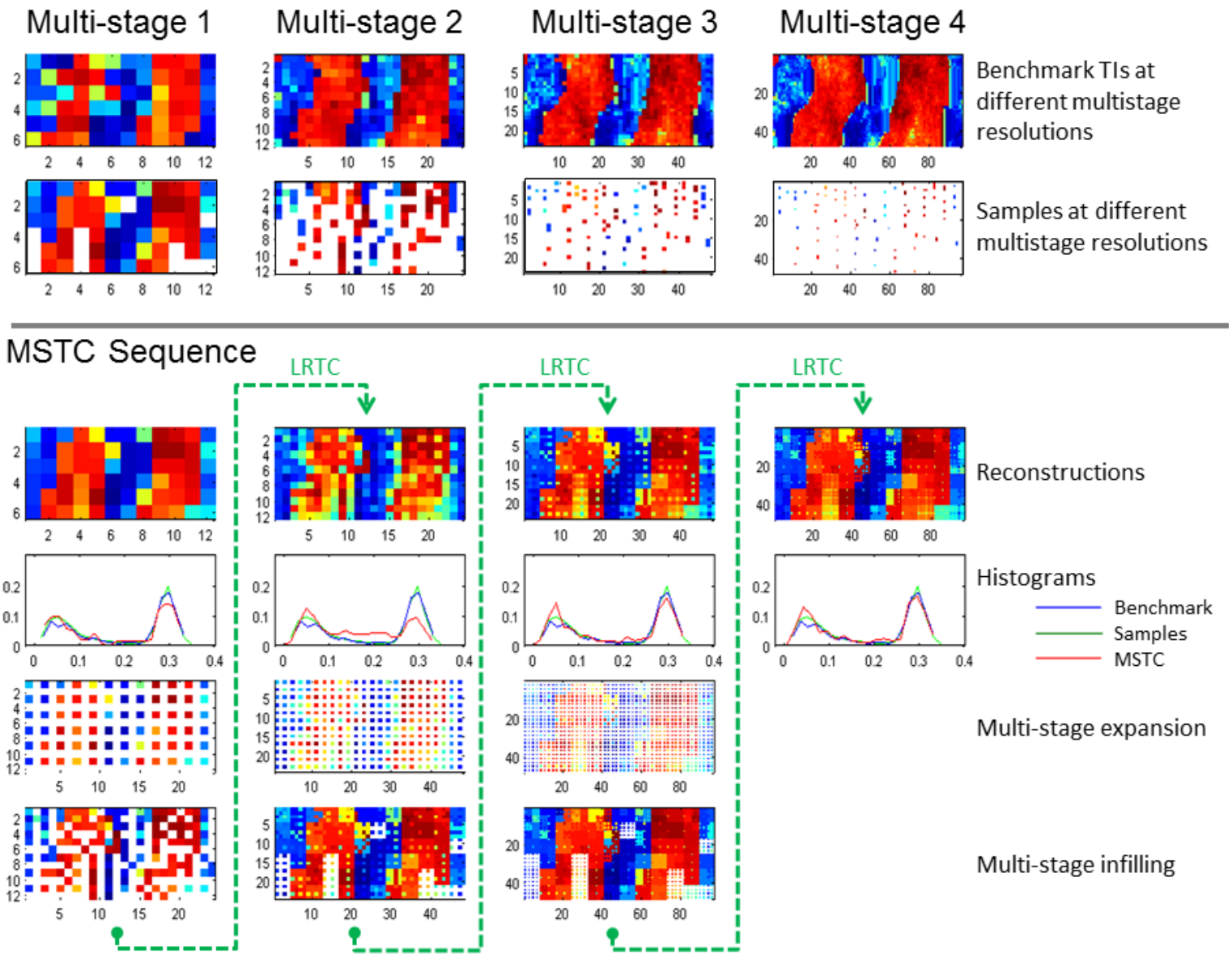

Figure 27. Step-by-step 4-stage MSTC of benchmark set 1, model C using MSTC-4 (only one slice shown).

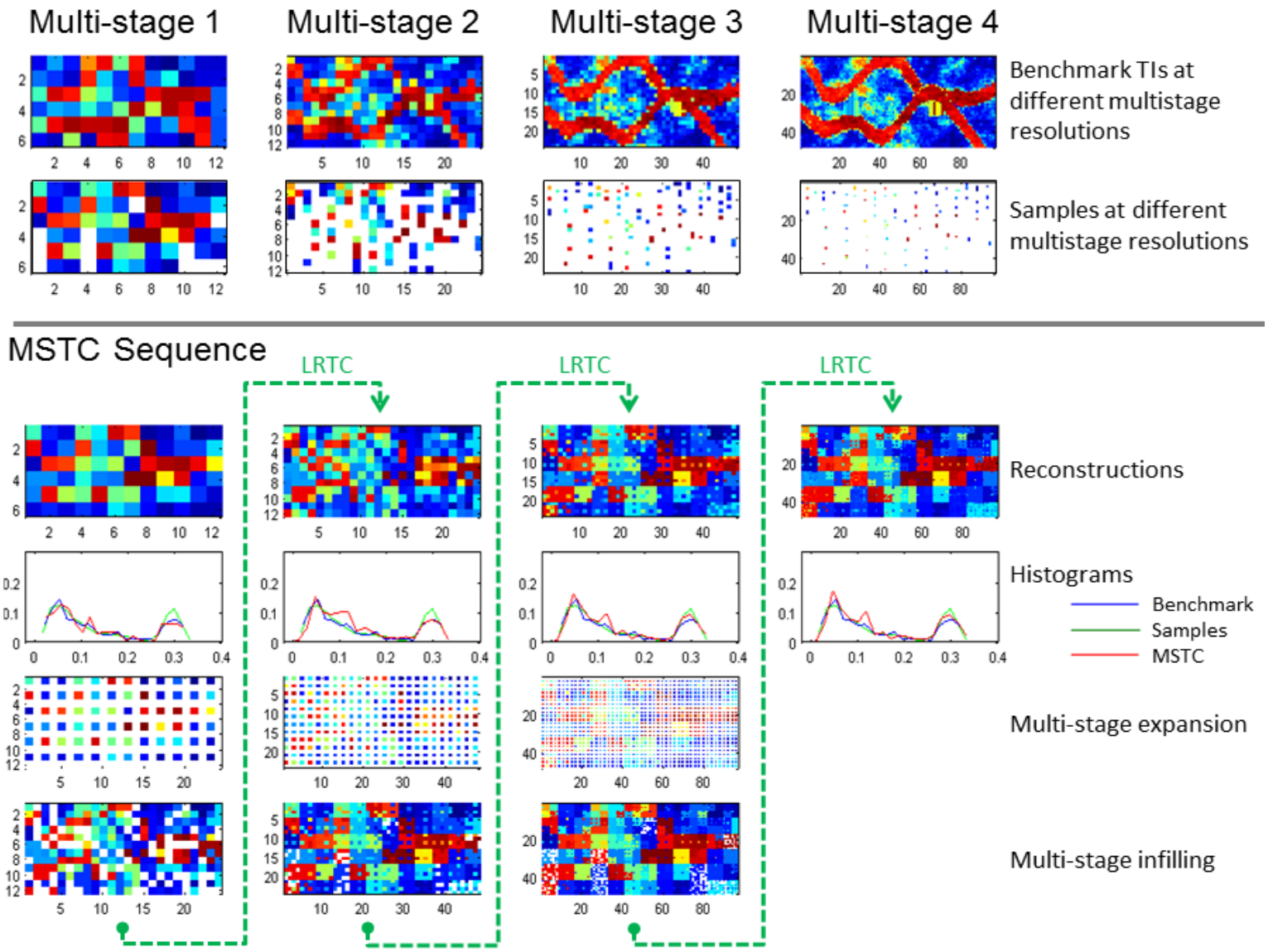

Figure 28. Step-by-step 4-stage MSTC of benchmark set 2, model A, using MSTC-4 (only one slice shown).

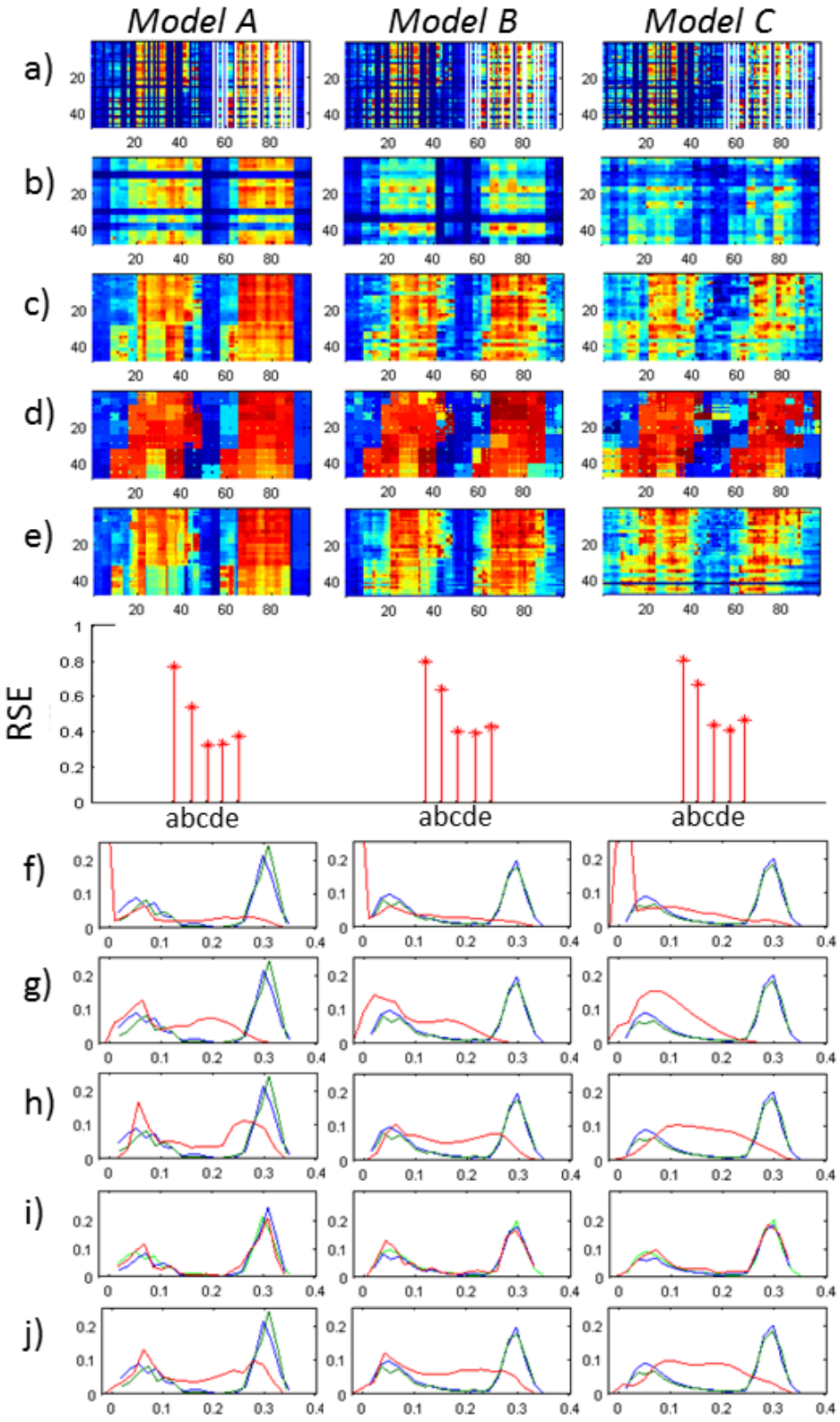

Figure 29. Summary of reconstructions using benchmark set 1.

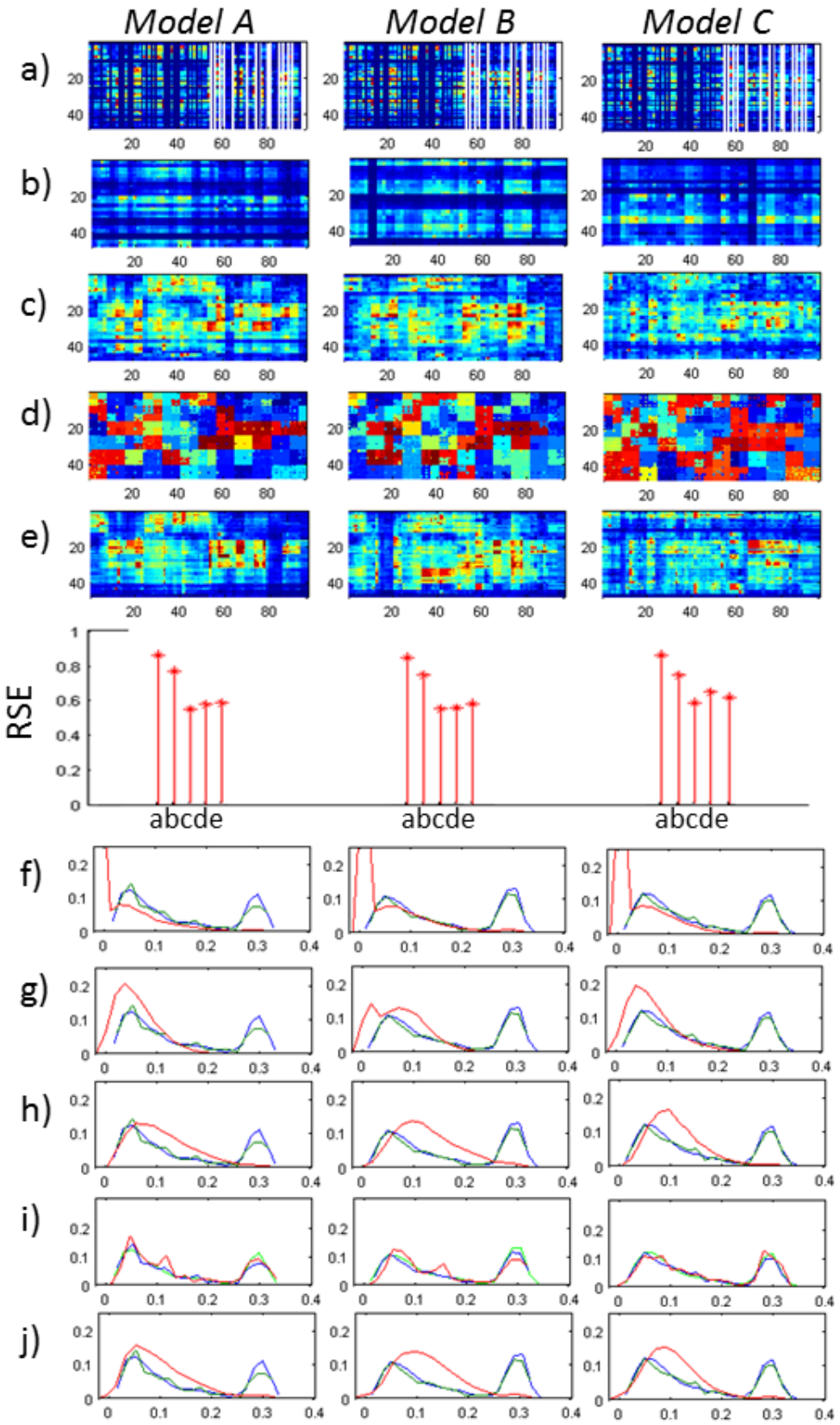

Figure 30. Summary of reconstructions using benchmark set 2.

5.1.5 Comparison of High-Order Statistics

A statistical test for multiple-point connectivity is performed using HOS given by 4th order spatial cumulants (Dimitrakopoulos et al., 2010).

Briefly stated, higher-order cumulants depend on combinations of lower-order moments, such that their reproduction implicitly contains the lower-order statistics. The k^{th} order spatial cumulant of a random function $Z(\boldsymbol{u})$ is stated as:

$$c_k^Z(\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{k-1}) = Cum\{Z(\boldsymbol{u}), Z(\boldsymbol{u}+\mathbf{h}_1), Z(\boldsymbol{u}+\mathbf{h}_2), \dots, Z(\boldsymbol{u}+\mathbf{h}_{k-1})\}, \quad \textbf{(5.1)}$$

where lag separations $\mathbf{h}_\alpha, \alpha = 1, \dots, (k-1)$ are used, as in Figure 31, to define a particular spatial architecture used to measure the cumulants statistics. The relationship between cumulants and moments is given by:

$$c_k^Z(\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{k-1}) = \sum (-1)^{p-1}(p-1)!\, E\left\{\prod_{s\in p} E\left(\prod_{i\in s} Z_i\right)\right\}, \quad \textbf{(5.2)}$$

where p runs through the list of all partitions of $\{1, \dots, k-1\}$, and s runs through the list of all blocks of the partition p. A step-by-step example calculation is included in the Appendix.

For the 4th order spatial cumulants, consider **5.3**, which is the fourth order moment minus combinations of lower order cumulants:

$$
\begin{aligned}
c_4^Z(\mathbf{h}_1,\mathbf{h}_2,\mathbf{h}_3) = {} & E\{Z(\boldsymbol{u})\cdot Z(\boldsymbol{u}+\mathbf{h}_1)\cdot Z(\boldsymbol{u}+\mathbf{h}_2)\cdot Z(\boldsymbol{u}+\mathbf{h}_3)\} \\
& - E\{Z(\boldsymbol{u})\cdot Z(\boldsymbol{u}+\mathbf{h}_1)\}\cdot E\{Z(\boldsymbol{u}+\mathbf{h}_2)\cdot Z(\boldsymbol{u}+\mathbf{h}_3)\} \\
& - E\{Z(\boldsymbol{u})\cdot Z(\boldsymbol{u}+\mathbf{h}_2)\}\cdot E\{Z(\boldsymbol{u}+\mathbf{h}_1)\cdot Z(\boldsymbol{u}+\mathbf{h}_3)\} \\
& - E\{Z(\boldsymbol{u})\cdot Z(\boldsymbol{u}+\mathbf{h}_3)\}\cdot E\{Z(\boldsymbol{u}+\mathbf{h}_1)\cdot Z(\boldsymbol{u}+\mathbf{h}_2)\} \\
& - E\{Z(\boldsymbol{u})\}\cdot E\{Z(\boldsymbol{u}+\mathbf{h}_1)\cdot Z(\boldsymbol{u}+\mathbf{h}_2)\cdot Z(\boldsymbol{u}+\mathbf{h}_3)\} \\
& - E\{Z(\boldsymbol{u})\}\cdot E\{Z(\boldsymbol{u})\cdot Z(\boldsymbol{u}+\mathbf{h}_2)\cdot Z(\boldsymbol{u}+\mathbf{h}_3)\} \\
& - E\{Z(\boldsymbol{u})\}\cdot E\{Z(\boldsymbol{u})\cdot Z(\boldsymbol{u}+\mathbf{h}_1)\cdot Z(\boldsymbol{u}+\mathbf{h}_3)\} \\
& - E\{Z(\boldsymbol{u})\}\cdot E\{Z(\boldsymbol{u})\cdot Z(\boldsymbol{u}+\mathbf{h}_1)\cdot Z(\boldsymbol{u}+\mathbf{h}_2)\} \\
& + 2E\{Z(\boldsymbol{u})\cdot Z(\boldsymbol{u}+\mathbf{h}_1)\}\cdot E\{Z(\boldsymbol{u})\}^2 \\
& + 2E\{Z(\boldsymbol{u})\cdot Z(\boldsymbol{u}+\mathbf{h}_3)\}\cdot E\{Z(\boldsymbol{u})\}^2 \\
& + 2E\{Z(\boldsymbol{u}+\mathbf{h}_1)\cdot Z(\boldsymbol{u}+\mathbf{h}_3)\}\cdot E\{Z(\boldsymbol{u})\}^2 \\
& + 2E\{Z(\boldsymbol{u}+\mathbf{h}_2)\cdot Z(\boldsymbol{u}+\mathbf{h}_3)\}\cdot E\{Z(\boldsymbol{u})\}^2 \\
& + 2E\{Z(\boldsymbol{u}+\mathbf{h}_1)\cdot Z(\boldsymbol{u}+\mathbf{h}_2)\}\cdot E\{Z(\boldsymbol{u})\}^2 \\
& -6E\{Z(u)\}^4
\end{aligned}
\qquad (\mathbf{5.3})
$$

When $Z(\boldsymbol{u})$ is a zero-mean random function, **5.3** reduces to:

$$
\begin{aligned}
c_4^Z(\mathbf{h}_1,\mathbf{h}_2,\mathbf{h}_3) = {} & E\{Z(\boldsymbol{u})\cdot Z(\boldsymbol{u}+\mathbf{h}_1)\cdot Z(\boldsymbol{u}+\mathbf{h}_2)\cdot Z(\boldsymbol{u}+\mathbf{h}_3)\} \\
& - E\{Z(\boldsymbol{u})\cdot Z(\boldsymbol{u}+\mathbf{h}_1)\}\cdot E\{Z(\boldsymbol{u}+\mathbf{h}_2)\cdot Z(\boldsymbol{u}+\mathbf{h}_3)\} \\
& - E\{Z(\boldsymbol{u})\cdot Z(\boldsymbol{u}+\mathbf{h}_2)\}\cdot E\{Z(\boldsymbol{u}+\mathbf{h}_1)\cdot Z(\boldsymbol{u}+\mathbf{h}_3)\} \\
& - E\{Z(\boldsymbol{u})\cdot Z(\boldsymbol{u}+\mathbf{h}_3)\}\cdot E\{Z(\boldsymbol{u}+\mathbf{h}_1)\cdot Z(\boldsymbol{u}+\mathbf{h}_2)\}
\end{aligned}
\qquad (\mathbf{5.4})
$$

From which it is possible to see the implicit two-point covariances $E\{Z(\boldsymbol{u})\cdot Z(\boldsymbol{u}+\mathbf{h}_i)\}$, and $E\{Z(\boldsymbol{u}+\mathbf{h}_i)\cdot Z\left(\boldsymbol{u}+\mathbf{h}_{\substack{j\neq i\\ j>i}}\right)\}$ for $i=1,2,3, j=1,2,3$.

A useful property of cumulants is that they are shift-invariant. That is, for orders 2 and greater, the cumulants of a non-centered random function are the same as for a zero-mean random function.

Experimental 4th order cumulants were calculated using **5.5** based on the orthogonal template $T_4^{\mathbf{h}_1,\mathbf{h}_2,\mathbf{h}_3}$ defined in Figure 31, where x, y, and z are replaced by 1, 2, and 3, respectively.

$$
\begin{aligned}
c^{T_4^{\mathbf{h}_1,\mathbf{h}_2,\mathbf{h}_3}} &= \frac{1}{N_{\mathbf{h}_1,\mathbf{h}_2,\mathbf{h}_3}} \sum_{i=1}^{N_{\mathbf{h}_1,\mathbf{h}_2,\mathbf{h}_3}} Z(\boldsymbol{u_i}) \cdot Z(\boldsymbol{u_i}+\mathbf{h}_1) \cdot Z(\boldsymbol{u_i}+\mathbf{h}_2) \cdot Z(\boldsymbol{u_i}+\mathbf{h}_3) \\
&- \frac{1}{N_{\mathbf{h}_1,\mathbf{h}_2,\mathbf{h}_3}} \left[\sum_{i=1}^{N_{\mathbf{h}_1,\mathbf{h}_2,\mathbf{h}_3}} Z(\boldsymbol{u_i}) \cdot Z(\boldsymbol{u_i}+\mathbf{h}_1)\right]\left[\sum_{i=1}^{N_{\mathbf{h}_1,\mathbf{h}_2,\mathbf{h}_3}} Z(\boldsymbol{u_i}+\mathbf{h}_2) \cdot Z(\boldsymbol{u_i}+\mathbf{h}_3)\right] \\
&- \frac{1}{N_{\mathbf{h}_1,\mathbf{h}_2,\mathbf{h}_3}} \left[\sum_{i=1}^{N_{\mathbf{h}_1,\mathbf{h}_2,\mathbf{h}_3}} Z(\boldsymbol{u_i}) \cdot Z(\boldsymbol{u_i}+\mathbf{h}_2)\right]\left[\sum_{i=1}^{N_{\mathbf{h}_1,\mathbf{h}_2,\mathbf{h}_3}} Z(\boldsymbol{u_i}+\mathbf{h}_1) \cdot Z(\boldsymbol{u_i}+\mathbf{h}_3)\right] \\
&- \frac{1}{N_{\mathbf{h}_1,\mathbf{h}_2,\mathbf{h}_3}} \left[\sum_{i=1}^{N_{\mathbf{h}_1,\mathbf{h}_2,\mathbf{h}_3}} Z(\boldsymbol{u_i}) \cdot Z(\boldsymbol{u_i}+\mathbf{h}_3)\right]\left[\sum_{i=1}^{N_{\mathbf{h}_1,\mathbf{h}_2,\mathbf{h}_3}} Z(\boldsymbol{u_i}+\mathbf{h}_1) \cdot Z(\boldsymbol{u_i}+\mathbf{h}_2)\right]
\end{aligned}
\tag{5.5}
$$

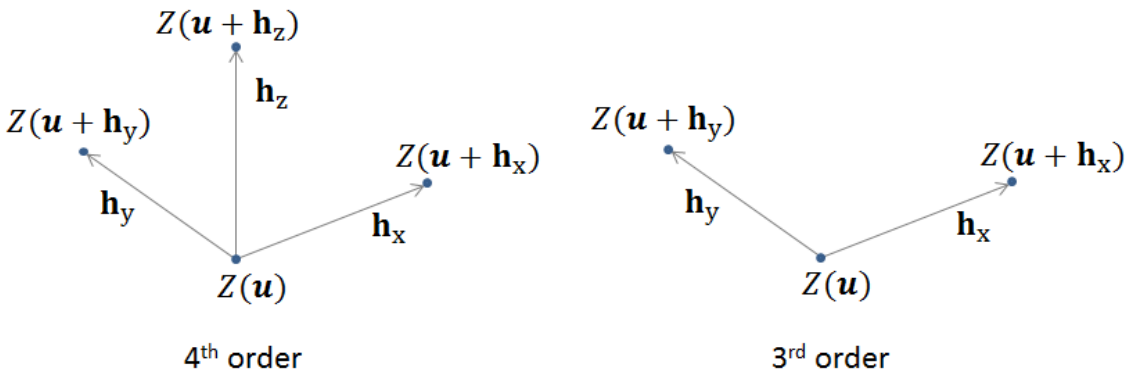

Figure 31. Fourth (left) and third order (right) orthogonal cumulant templates.

Figures 32 and 33 show the 4th order spatial cumulant maps for the benchmark models (A, B, C) presented in section 5.1.1, sparse entries, and their respective MSTC-4 reconstructions. The cumulants maps reflect the subtle differences in channels that exist between each benchmark model. It is expected that the cumulants map for a set of entries and that of its TI should be relatively close. This is confirmed by visual inspection. It is encouraging to see that the cumulants maps from the reconstructions are similar to their benchmark counterparts (i.e. the third row is similar to the first row). This reflects the reconstruction of the spatial architectures inherent to the original TIs, at least up to 4th order spatial statistics. Between reconstructions (i.e. along the third row), it is also possible to see the subtle differences that exist between models, more so than in the cumulants maps for sampled entries. This further suggests that the reconstruction

of entries occurs in agreement with the particular structure of each model, as defined by the original cumulants maps in row 1.

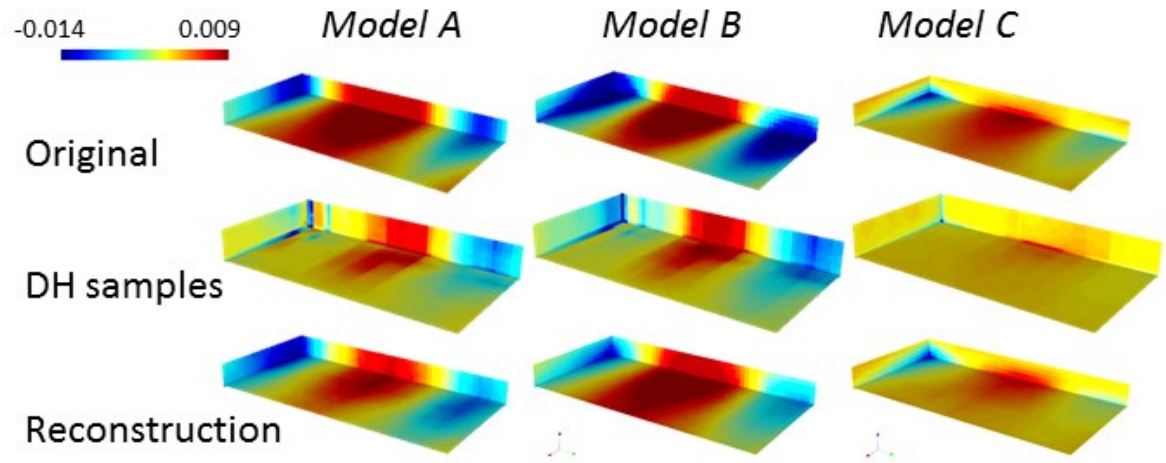

Figure 32. Fourth order cumulant map reproductions for benchmark set 1 models.

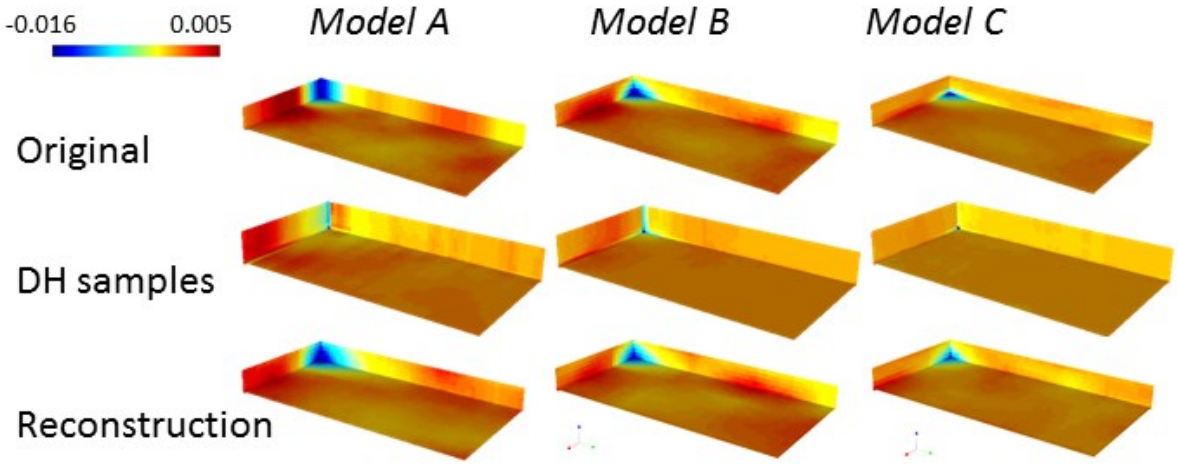

Figure 33. Fourth order cumulant map reproductions for benchmark set 2 models.

CHAPTER 6 CONCLUSIONS AND FUTURE RESEARCH

This thesis presents a method called MSTC, which is used to construct continuous, 3D TIs that are conditioned to sparse entries (<2%) and obtained from a real drilling scheme. TIs are pattern databases used by MPS and HOS simulation algorithms to build a simulated value by drawing a sample from a CCDF. A MPS or HOS algorithm that uses a TI built directly from data becomes a fully data-driven algorithm, with the ability to reproduce its own data statistics rather than those from an external TI.

The MSTC methodology extends on the earlier 2D reconstruction method called MSMC (Yahya, 2011; Yahya et al., 2012). Unlike MSMC, the MSTC reconstruction method is not limited to uniform random sampling schemes and may be applied to real-world drill-sampling schemes. MSTC uses the LRTC algorithm (Liu et al., 2009; Liu et al., 2013). Five different variants of the MSTC methodology were tested on various 3D benchmark porosity models. An error measure (RSE) was used to quantify the quality of the reconstructions. MSTC-4 provided the best reconstructions based on low RSE error measures, good visual representations - particularly for less complex datasets (e.g. benchmark set 1) and reasonable corroboration of spatial high-order statistics by 4^{th} order spatial cumulants. MSTC-4 is based on rebuilding a downscaled tensor (i.e. a low resolution version of the original sparse tensor) up to the original tensor resolution by upsizing the downscaled tensor through successive stages of LRTC reconstructions. At each stage, missing entries of the tensor are in-filled with available values from other tensors prior to the reconstruction. If the problem to be solved involves a not-too-complex reconstruction, MSTC may be a good TI generator to try. For more complex cases, it may be that increasing the number of multi-stages would help to capture higher resolution features (e.g. benchmark set 2). However, TIs with trends (e.g. Figure 7) may be too difficult to reconstruct given that they do not conform to repetitive, stationary models – this would need to be tested.

Future work could test other tensor completion algorithms that may prove to be more accurate. In this regard, a recent algorithm (Romera-Paredes & Pontil, 2013) showed that the nuclear norm, which is the tightest convex relaxation of the matrix rank function, does not equate to the tightest convex relaxation of the tensor rank function. In other

words, the nuclear norm metric is suboptimal given that it is not invariant under different mode-n matricizations of a tensor. Romera-Paredes & Pontil (2013) developed an algorithm based on the Euclidean ball as a convex approximation to the tensor rank. The Euclidean ball metric is advantageous since it is invariant under mode-n matricizations (the tensor nuclear norm is not) - thus avoiding the issue of trying to optimize multiple matrices simultaneously (refer to Section 4.2). They showed that this method improves on the reconstruction of tensors in terms of estimation error. The application of their algorithm, instead of LRTC, could further improve the quality of the reconstructions.

Graph theory as it relates to matrix completion may present further solutions to solve sparse matrix/tensor reconstruction problems. Recent research (Bhojanapalli & Jain, 2014) on this topic claims to provide recovery guarantees for matrix completion under adversarial sampling conditions.

An important point to consider with respect to the “blockiness” of the reconstructions in Chapter 5 is that TIs used in orebody models need not necessarily exist at point-support scale, given that mining extraction schedules are developed at block-support scales. If these side effects of the reconstruction are kept below the threshold of the orebody block dimensions, and are accurate enough to aid the simulation, such TIs may be useful for orebody modeling. Lastly, more testing on additional deposits, together with the application of different cumulants templates, would be useful to help refine the methodology. In this regard, the methodology from MSTC-4 was applied to a confidential mining dataset with generally positive results. While an exhaustive version of the mining dataset was not available to confirm the reproduction of HOS using 4th order spatial cumulants, visual inspection of the reconstructed TI indicated a geologically reasonable reconstruction of continuity along high and low grade values of the mineral deposit. In cases having very few samples within a specified range of the distribution, it is not possible to reconstruct a sufficient number of entries in the TI in order to satisfy the original data distribution. This point is important when trying to model orebodies with limited but very high grade values.

APPENDIX

A.1 MSTC Functions

The MSTC method runs on six functions not including the LRTC algorithm. Each of these is described next using brief examples.

The ***downgrid*** function: The $downgrid$ function $\boldsymbol{\mathcal{G}}_l = downgrid(\boldsymbol{\mathcal{T}})$ is used to map the original sample data from tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I \times J \times K}$ to lower resolution tensors $\boldsymbol{\mathcal{G}}_l, \dots, \boldsymbol{\mathcal{G}}_S \in \mathbb{R}^{I' \times J' \times K'}$. Consider the simple arbitrary example in Figure 34 where $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{4 \times 4 \times 4}$, $S = 1$, and $l = 0{,}1$ results in $\boldsymbol{\mathcal{G}}_0 \in \mathbb{R}^{2 \times 2 \times 2}$. A simplified methodology is achieved when the dimensions of the lower resolution tensors are chosen as a multiple of the original tensor dimensions. The mapping of values to $\boldsymbol{\mathcal{G}}_0$ is achieved by drawing uniformly at random from those elements that are represented in each sub-tensor of $\boldsymbol{\mathcal{T}}$ (outlined in red in Figure 34). For example

$$\boldsymbol{\mathcal{G}}_1(1,1,1) = \begin{cases} 2 \ with\ probability \dfrac{1}{2} \\ 3 \ with\ probability \dfrac{1}{2} \end{cases}. \qquad \boldsymbol{\mathcal{G}}_0(1,1,1) = \begin{cases} 1 \ with\ probability \dfrac{4}{14} \\ 2 \ with\ probability \dfrac{3}{14} \\ 3 \ with\ probability \dfrac{2}{14} \\ 5 \ with\ probability \dfrac{1}{14} \\ 6 \ with\ probability \dfrac{1}{14} \\ 8 \ with\ probability \dfrac{1}{14} \end{cases} \quad \textbf{(A2.1)}$$

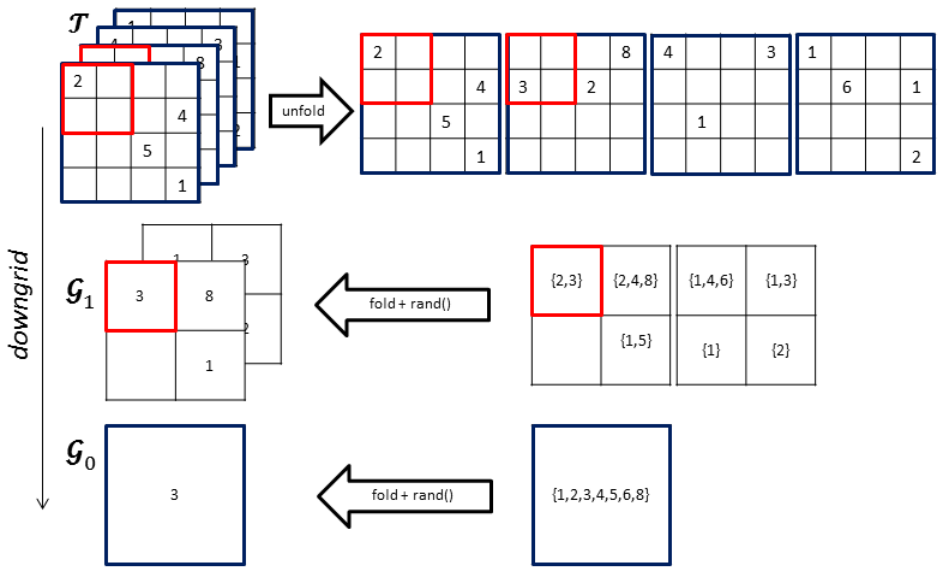

Figure 34. Example showing the usage of the $downgrid()$function.

The $\boldsymbol{regrid}$ function: This function $regrids$ the tensor $\mathcal{G}_l$ to the resolution scale of a given stage level, $\mathcal{G}_l^{regrid} = regrid(\mathcal{G}_l, uscaleVec)$. Refer to Figures 35 and 36 for illustrative examples. Mode-by-mode expansion is achieved by copying mode slices and inserting them next to the original slice. The parameter vector $uscaleVec$ is used to define how often this occurs along each mode. For example, tensor $\mathcal{G}_0 \in \mathbb{R}^{X \times Y \times Z}$ $regrids$ to $\mathcal{G}_l \in \mathbb{R}^{2^l X \times 2^l Y \times 2^l Z}$ where $uscaleVec = [2^l, 2^l, 2^l]$. This function allows later multi-stages to access values from $downgridded$ tensors. For example, in order for $\mathcal{R}_1^{up}$ to access values from $\mathcal{G}_0$, $\mathcal{G}_0$ must first be $regridded$ to the same resolution as $\mathcal{R}_1^{up}$. The simplest case implies a doubling of each mode-size as the multi-stage number increases – see Step 3a in Section 3.3.

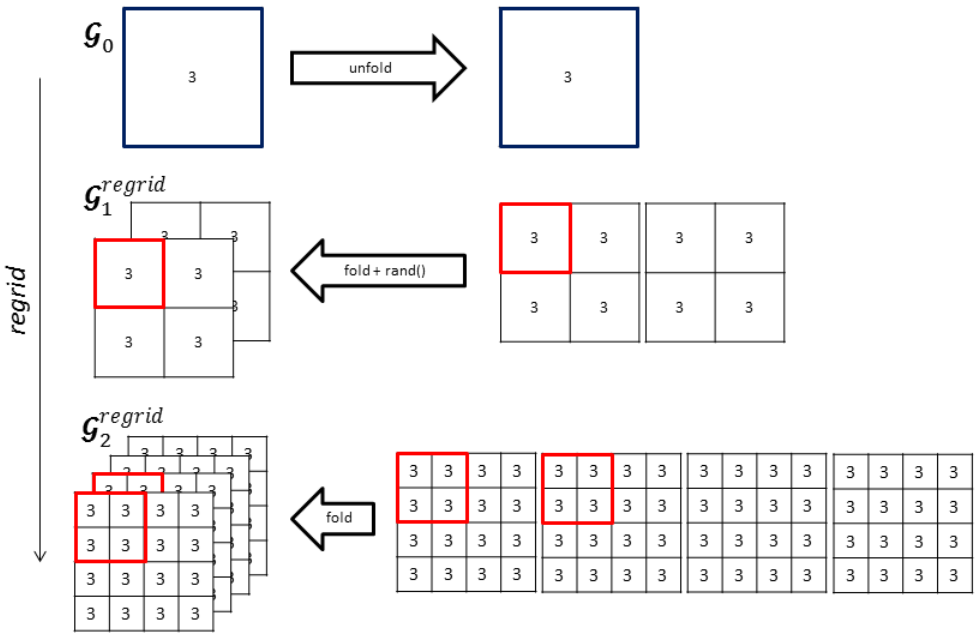

Figure 35. Example 1 showing the usage of the $regrid()$function.

The ***upsize*** function: The $upsize$ function $\boldsymbol{\mathcal{R}}_{l+1}^{up} = upsize(\boldsymbol{\mathcal{R}}_l, uscaleVec)$, where $\boldsymbol{\mathcal{R}}_l$ is the outcome of an LRTC reconstruction, increases the dimension of each mode of the tensor by inserting empty slices between each reconstructed slice. An example is shown in Figure 36 (see Step 2 in Section 4.3.2 and refer to Figure 22).

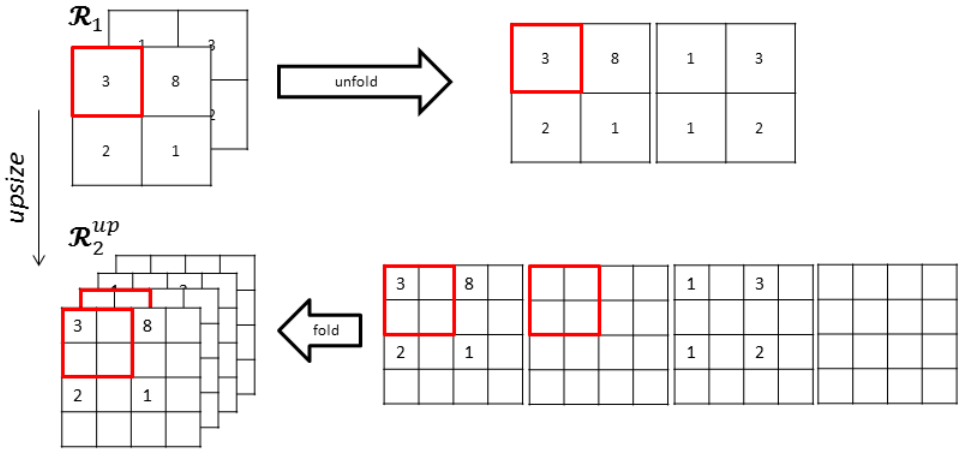

Figure 36. Example showing the usage of the $upsize()$ function.

The ***randup*** function: As per the method used in MSMC (Yahya, 2011) - see Step 4 in Figure 12, the $randup$ function $upsizes$ a tensor and then randomly interchanges entries that are adjacent to unrepresented slices – see Figure 37. This is done to avoid

having un-sampled slices prior to the reconstruction step (LRTC). Let $b_x, b_y,$ and b_z represent three independent realizations of a Bernoulli random variable R with a probability mass function defined by

$$f(\mathrm{R}, p) = \begin{cases} p, & if\ \mathrm{R} = 1, \\ 1 - p, & if\ \mathrm{R} = 0, \\ 0, & otherwise. \end{cases} \quad \textbf{(A2.2)}$$

The following randomization step enacts three random draws with probability $p = 0.5$ at each location within the tensor:

$$\mathcal{R}_1^{randup}(x + r_x, y + r_y, z + r_z) = \mathcal{R}_1(x, y, z) \quad \textbf{(A2.3)}$$

If $r_x = r_y = r_z = 0$ the value at (x, y, z) does not shift its position. There is no need to call the $upsize$ function first as it is incorporated into $randupsize$. The structure of the function call is $\mathcal{R}_1^{randupsize} = randupsize(\mathcal{R}_1, scaleVec)$.

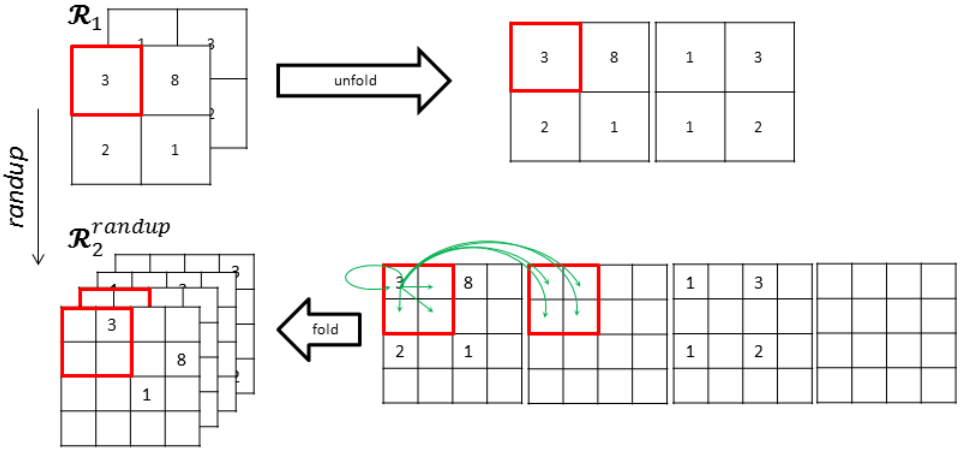

Figure 37. Example showing the usage of the $randup()$ function.

The ***correlup*** function: The $correlup$ function behaves similarly to the $randupsize$ function except that entries are shifted towards other similar entries (with reference to a different grid), rather than randomly. The intended action for this function is to preserve the continuity of structures rather than to break spatial continuity through a randomized process. $Correlup$ refers to a lower resolution (i.e. lower stage-level) tensor that has been $regridded$ to the current resolution. For each location of an $upsized$ tensor

(e.g. $\mathcal{R}_1^{upsize}$) the function scans the surrounding entries within the corresponding image (i.e. blue cube of $\mathcal{G}_2^{regrid}$ in Figure 38) and calculates an error difference given by

$$d = \left|\mathcal{R}_1^{up}(\boldsymbol{u}) - \mathcal{G}_2(\boldsymbol{u}+\boldsymbol{h})\right| \Big/ \mathcal{R}_1^{up}(\boldsymbol{u})' \qquad \textbf{(A2.4)}$$

where $\boldsymbol{u} = [x, y, z]$ and $\boldsymbol{h} = [h_x, h_y, h_z]$ for $h_x = h_y = h_z = 0,1$. A value at $\mathcal{R}_1^{upsize}(\boldsymbol{u})$ is shifted in the direction of minimum difference. A random shift only occurs if no values from $\mathcal{G}_2$ are available for comparison with a given $\mathcal{R}_1^{upsize}(\boldsymbol{u})$ a random shift.

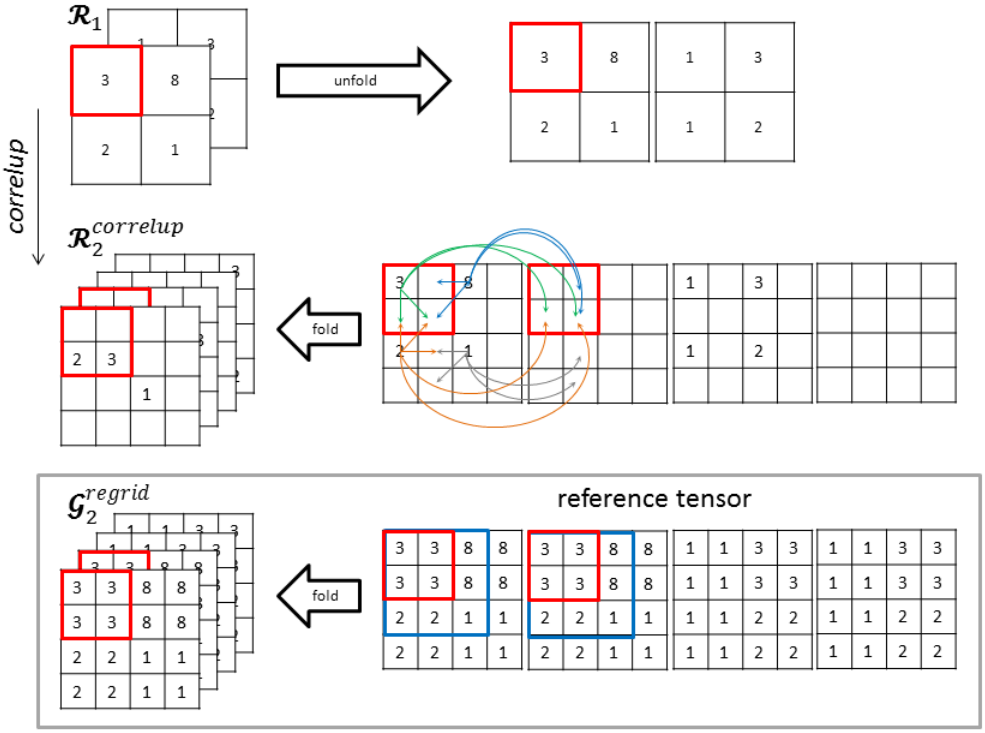

Figure 38. Example of the usage of the $correlup()$ function.

The ***upsample*** function: The $upsample$ function, illustrated in Figure 39, $\mathcal{R}_i^{upsample} = upsample(\mathcal{R}_i^{up}, \mathcal{G}_{i+1}, \mathcal{G}_{i-1}, \ldots)$ samples entries from a $downgridded$ tensor to infill missing values in the $upsized$ tensor. Only the first two function parameters are required and any number of lookup tensors $\mathcal{G}$ may be used. Sampling only occurs for empty values of $\mathcal{R}_i^{up}$ (i.e. values are never replaced). $Upsampling$ can be extended to include

$upsized$ lower level grids if more entries are required however this has a tendency to make the final output tensor more "blocky." Consider the function call

$$\mathcal{R}_1^{upsample} = upsample(\mathcal{R}_1^{up}, \mathcal{G}_2, \mathcal{G}_0), \tag{A2.5}$$

where

$$\mathcal{R}_1^{upsample}(\boldsymbol{u}) = \begin{cases} \mathcal{G}_0^{regrid}(\boldsymbol{u}), & if\ \mathcal{G}_0^{regrid}(\boldsymbol{u})\ exists \\ \mathcal{G}_2(\boldsymbol{u}), & if\ \mathcal{G}_2(\boldsymbol{u})\ exists \\ \mathcal{R}_1^{up}(\boldsymbol{u}), & if\ \mathcal{R}_1^{up}(\boldsymbol{u})\ exists \\ \emptyset, & if\ \mathcal{R}_1^{up}(\boldsymbol{u}) = \emptyset \end{cases}. \tag{A2.6}$$

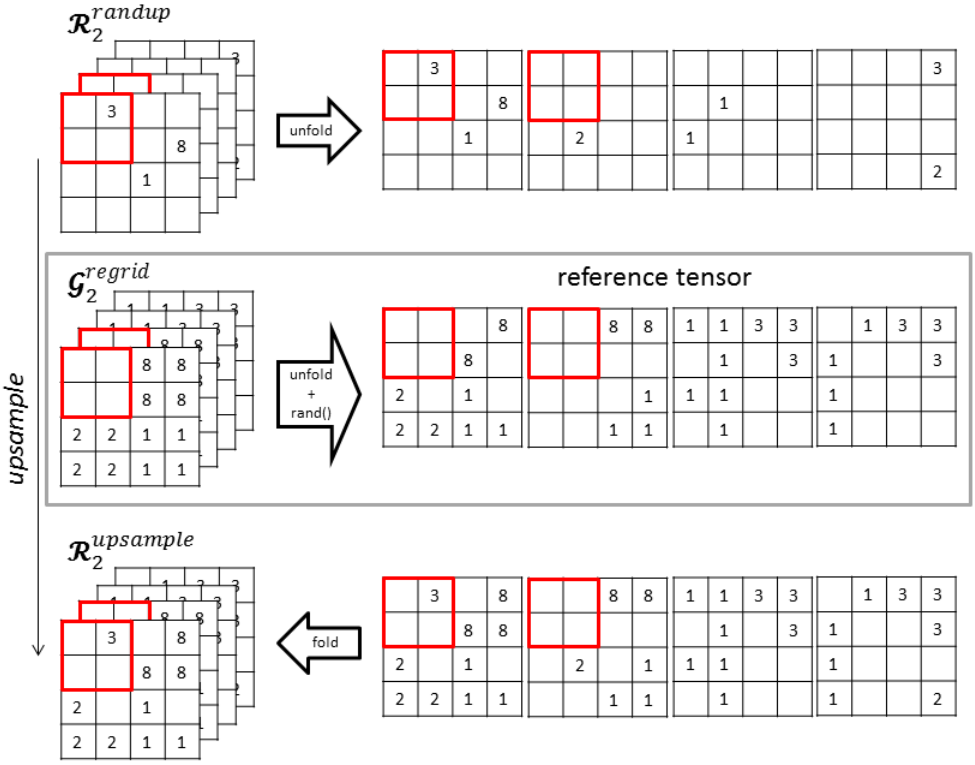

Figure 39. Example of the usage of the $upsample()$ function.

A.2 Example calculation of a 3rd order cumulant

The step-by-step calculation of the third order cumulant using the expression in (**A3.1**) and the third order template in Figure 40 is explicitly presented next.

$$c_k^Z(\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{k-1}) = \sum (-1)^{p-1}(p-1)!\, E\{\prod_{s\in p} E(\prod_{i\in s} Z_i)\} \qquad \textbf{(A3.1)}$$

Let $c_3^Z(\mathbf{h}_x, \mathbf{h}_y)$ represent the third order cumulant with random variables $Z(\boldsymbol{u}), Z(\boldsymbol{u}+\mathbf{h}_x)$, and $Z(\boldsymbol{u}+\mathbf{h}_y)$ with lag separations $\mathbf{h}_x$ and $\mathbf{h}_y$. For simplicity, restate the RVs as

$$X = Z(\boldsymbol{u}), \quad Y = Z(\boldsymbol{u}+\mathbf{h}_x), \quad Z = Z(\boldsymbol{u}+\mathbf{h}_y). \qquad \textbf{(A3.2)}$$

Table 5 lists the RV partitions p and blocks s corresponding to lags $\mathbf{h}_x$, and $\mathbf{h}_y$:

Table 5. Partitions of 3rd order spatial cumulant.

Partitions of the RVs defined by the template in Figure 40	Number of blocks per partition
$\{\{X\},\{Y\},\{Z\}\}$,	3
$\{\{XY\},\{Z\}\}$,	2
$\{\{XZ\},\{Y\}\}$,	2
$\{\{X\},\{YZ\}\}$,	2
$\{\{X,Y,Z\}\}$	1

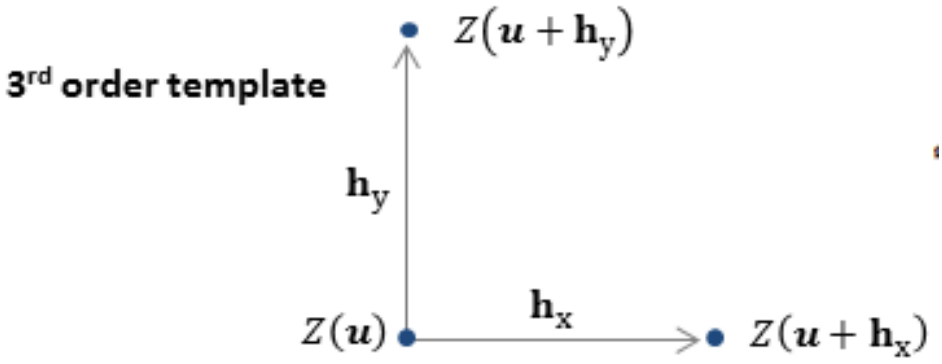

Figure 35. Example of a third order spatial cumulant template.

The third order cumulant, using (**A3.1**), is given by

$$c_3^Z(\mathbf{h}_x,\mathbf{h}_y) = (3-1)!\,(-1)^{3-1}E\{X\}E\{Y\}E\{Z\} + (2-1)!\,(-1)^{2-1}E\{XY\}E\{Z\}$$
$$+ (2-1)!\,(-1)^{2-1}E\{XZ\}E\{Y\} + (2-1)!\,(-1)^{2-1}E\{YZ\}E\{X\}$$
$$+(1-1)!\,(-1)^{1-1}E\{XYZ\}$$
$$= 2E\{X\}\{Y\}\{Z\} - E\{XY\}E\{Z\} - E\{XZ\}E\{Y\} - E\{YZ\}E\{X\} + E\{XYZ\}.$$

Substituting back into (**A3.2**) gives

$$c_3^Z(\mathbf{h}_x,\mathbf{h}_y) = 2E\{Z(\boldsymbol{u})\}\{Z(\boldsymbol{u}+\boldsymbol{h_x})\}\{Z(\boldsymbol{u}+\boldsymbol{h_y})\}$$
$$-E\{Z(\boldsymbol{u})Z(\boldsymbol{u}+\boldsymbol{h_x})\}E\{Z(\boldsymbol{u}+\boldsymbol{h_y})\}$$
$$-E\{Z(\boldsymbol{u})Z(\boldsymbol{u}+\boldsymbol{h_y})\}E\{Z(\boldsymbol{u}+\boldsymbol{h_x})\}$$
$$-E\{Z(\boldsymbol{u}+\boldsymbol{h_x})Z(\boldsymbol{u}+\boldsymbol{h_y})\}E\{Z(\boldsymbol{u})\}$$
$$+E\{Z(\boldsymbol{u})Z(\boldsymbol{u}+\boldsymbol{h_x})Z(\boldsymbol{u}+\boldsymbol{h_y})\}.$$

For zero-mean RF $Z(\boldsymbol{u})$ the above expression simplifies to a third order moment. That is,

$$c_3^Z(\mathbf{h}_x,\mathbf{h}_y) = E\{Z(\boldsymbol{u})Z(\boldsymbol{u}+\boldsymbol{h_x})Z(\boldsymbol{u}+\boldsymbol{h_y})\}.$$

REFERENCES

Allard, D., 1993. On the connectivity of two random set models: the truncated Gaussian and the Boolean, Geostatistics Tróia'92. Springer, pp. 467-478.

Allard, D., Froidevaux, R. and Biver, P., 2005. Accounting for non-stationarity and interactions in object simulation for reservoir heterogeneity characterization, Geostatistics Banff 2004. Springer, pp. 155-164.

Arpat, G.B., 2005. Sequential simulation with patterns. PhD thesis, Dept. of Petrol. Eng., Stanford University, CA

Balzano, L., Nowak, R. and Recht, B., 2010. Online identification and tracking of subspaces from highly incomplete information, Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on. IEEE, pp. 704-711.

Benndorf, J. and Dimitrakopoulos, R., 2013. Stochastic long-term production scheduling of iron ore deposits: Integrating joint multi-element geological uncertainty. Journal of Mining Science, 49(1): 68-81.

Bhojanapalli, S. and Jain, P., 2014. Universal Matrix Completion. arXiv preprint arXiv:1402.2324. available online: http://arxiv.org/abs/1402.2324.

Boisvert, J.B., Leuangthong, O., Ortiz, J.M., Deutsch, C.V., 2008. A methodology to construct training images for vein-type deposits. Computers & Geosciences, 34(5), 491-502.

Boisvert, J.B., Pyrcz, M.J. and Deutsch, C.V., 2007. Multiple-point statistics for training image selection. Natural Resources Research, 16(4): 313-321.

Boucher, A., Gupta, R., Caers, J. and Satija, A., 2010. Tetris: a training image generator for SGeMS. Stanford Center for Reservoir Forecasting.

Caers, J. and Zhang, T., 2004. Multiple-point geostatistics: A quantitative vehicle for integrating geologic analogs into multiple reservoir models, AAPG Mem 80:383–394

Cai, J.-F., Candès, E.J. and Shen, Z., 2010. A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization, 20(4): 1956-1982.

Candès, E.J. and Recht, B., 2009. Exact matrix completion via convex optimization. Foundations of Computational Mathematics, 9(6): 717-772.

Candès, E.J., Romberg, J.K. and Tao, T., 2006a. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. Information Theory, IEEE Transactions on, 52(2): 489-509.

Candès, E.J., Romberg, J.K. and Tao, T., 2006b. Stable signal recovery from incomplete and inaccurate measurements. Communications on pure and applied mathematics, 59(8): 1207-1223.

Candès, E.J. and Tao, T., 2006. Near-optimal signal recovery from random projections: Universal encoding strategies? Information Theory, IEEE Transactions on, 52(12): 5406-5425.

Candès, E.J. and Tao, T., 2010. The power of convex relaxation: Near-optimal matrix completion. Information Theory, IEEE Transactions on, 56(5): 2053-2080.

Candès, E.J. and Wakin, M.B., 2008. An introduction to compressive sampling. Signal Processing Magazine, IEEE, 25(2): 21-30.

Chatterjee, S., Dimitrakopoulos, R. and Mustapha, H., 2012. Dimensional reduction of pattern-based simulation using wavelet analysis. Mathematical Geosciences: 1-32.

Chilès, J.-P. and Delfiner, P., 1999. Modeling spatial uncertainty. Geostatistics, Wiley series in probability and statistics.

Chugunova, T.L. and Hu, L.Y., 2008. Multiple-point simulations constrained by continuous auxiliary data. Mathematical geosciences, 40(2): 133-146.

Cojan, I., Fouché, O., Lopéz, S. and Rivoirard, J., 2005. Process-based reservoir modelling in the example of meandering channel. Geostatistics Banff 2004: 611-619.

Comunian, A., Jha, S.K., Giambastiani, B.M.S, Mariethoz, G., Kelly, B.F.J., 2014. Training Images from Process-Imitating Methods. Mathematical Geosciences, 46(2), 241-260.

Consuegra, F. and Dimitrakopoulos, R., 2010. Algorithmic approach to pushback design based on stochastic programming: Method, application and comparisons. Mining Technology, 119(2): 88-101.

Cressie, N. (1993). Statistics for Spatial Data: Wiley Series in Probability and Statistics: New York: John Wiley & Sons.

Dai, W., Milenkovic, O. and Kerman, E., 2011. Subspace evolution and transfer (SET) for low-rank matrix completion. Signal Processing, IEEE Transactions on, 59(7): 3120-3132.

David, M., 1988. Handbook of applied advanced geostatistical ore reserve estimation. Developments in Geomathematics 6. Elsevier, Amsterdam.

De Lathauwer, L., De Moor, B. and Vandewalle, J., 2000. A multilinear singular value decomposition. SIAM journal on Matrix Analysis and Applications, 21(4): 1253-1278.

dell'Arciprete, D., Bersezio, R., Felletti, F., Giudici, M., Comunian, A. and Renard, P., 2012. Comparison of three geostatistical methods for hydrofacies simulation: a test on alluvial sediments. Hydrogeology journal, 20(2): 299-311.

Deutsch, C.V., 2002. Geostatistical reservoir modeling. Oxford: Oxford University Press.

Deutsch, C.V. and Journel, A.G., 1998. Geostatistical software library and user's guide (GSLIB). Oxford University Press Oxford.

Deutsch, C.V. and Tran, T., 2002. FLUVSIM: a program for object-based stochastic modeling of fluvial depositional systems. Computers & Geosciences, 28(4): 525-535.

Deutsch, C.V. and Journel, A.G., 1992. Geostatistical software library and user&s guide, 1996. Oxford university press New York.

Deutsch, C.V. and Wang, L., 1996. Hierarchical object-based stochastic modeling of fluvial reservoirs. Mathematical Geology, 28(7): 857-880.

Dimitrakopoulos, R., 2011. Stochastic optimization for strategic mine planning: A decade of developments. Journal of Mining Science, 47(2): 138-150.

Dimitrakopoulos, R., Farrelly, C. and Godoy, M., 2002. Moving forward from traditional optimization: grade uncertainty and risk effects in open-pit design. Mining Technology, 111(1): 82-88.

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

Ding, C. and He, X., 2004. K-means clustering via principal component analysis, Proceedings of the twenty-first international conference on Machine learning. ACM, pp. 29.

Donoho, D.L., 2006. Compressed sensing. Information Theory, IEEE Transactions on, 52(4): 1289-1306.

Dunham, S. and Vann, J., 2007. Geometallurgy, geostatistics and project value—does your block model tell you what you need to know, Project evaluation conference, Melbourne, Victoria, pp. 19-20.

Emery, X., Lantuéjoul, C., 2014. Can a training image be a substitute for a random field model? Mathematical Geosciences, 46(2), 133-147.

Fazel, M., Hindi, H. and Boyd, S.P., 2001. A rank minimization heuristic with application to minimum order system approximation, American Control Conference, 2001. Proceedings of the 2001. IEEE, pp. 4734-4739.

Gandy, S., Recht, B. and Yamada, I., 2011. Tensor completion and low-n-rank tensor recovery via convex optimization. Inverse Problems, 27(2).

Gloaguen, E. and Dimitrakopoulos, R., 2009. Two-dimensional conditional simulations based on the wavelet decomposition of training images. Mathematical Geosciences, 41(6): 679-701.

Godoy, M., 2003. The effective management of geological risk in long-term production scheduling of open pit mines, PhD Thesis, McGill University, Montreal

Godoy, M. and Dimitrakopoulos, R., 2011. A risk analysis based framework for strategic mine planning and design method and application. Journal of Mining Science, 47.

Golub, G.H. and Van Loan, C.F., 1996. Matrix computations, 3. Johns Hopkins University Press.

Gómez-Hernández, J.J. and Wen, X.-H., 1998. To be or not to be multi-Gaussian? A reflection on stochastic hydrogeology. Advances in Water Resources, 21(1): 47-61.

Goodfellow, R., Consuegra, F.A., Dimitrakopoulos, R. and Lloyd, T., 2012. Quantifying multi-element and volumetric uncertainty, Coleman McCreedy deposit, Ontario, Canada. Computers & Geosciences, 42: 71-78.

Goodfellow, R. and Dimitrakopoulos, R., 2013. Algorithmic integration of geological uncertainty in pushback designs for complex multiprocess open pit mines. Mining Technology, 122(2): 67-77.

Goovaerts, P., 1997. Geostatistics for natural resources evaluation. Oxford University Press, USA.

Grant, M. and Boyd, S., 2011. CVX users' guide for CVX version 1.21 (build 808). Stanford University, CA

Guardiano, F. and Srivastava, R.M., 1993. Multivariate geostatistics: beyond bivariate moments. Geostatistics-Troia, 1: 133-144.

Gull, S. and Skilling, J., 1984. Maximum entropy method in image processing. Communications, Radar and Signal Processing, IEE Proceedings F, 131(6): 646-659.

Haldorsen, H.H. and Chang, D.M., 1986. Notes on stochastic shales: from outcrop to simulation model. Reservoir characterization, Academic Press, 445-485

Haldorsen, H.H. and Lake, L.W., 1984. A new approach to shale management in field-scale models. Society of Petroleum Engineers Journal, 24(4): 447-457.

Hartigan, J.A. and Wong, M.A., 1979. Algorithm AS 136: A k-means clustering algorithm. Applied statistics: 100-108.

Håstad, J., 1990. Tensor rank is NP-complete. Journal of Algorithms, 11(4): 644-654.

Hastings, W.K., 1970. Monte Carlo sampling methods using Markov chains and their applications. Biometrika, 57(1): 97-109.

Hildreth, C., 2006. A quadratic programming procedure. Naval research logistics quarterly, 4(1): 79-85.

Honarkhah, M. and Caers, J., 2010. Stochastic simulation of patterns using distance-based pattern modeling. Mathematical Geosciences, 42(5): 487-517.

Honarkhah, M. and Caers, J., 2012. Direct Pattern-Based Simulation of Non-stationary Geostatistical Models. Mathematical Geosciences: 1-22.

Hu, L. and Chugunova, T., 2008. Multiple-point geostatistics for modeling subsurface heterogeneity: A comprehensive review. Water Resources Research, 44(11)..

Jaynes, E.T., 1989. ET Jaynes: Papers on probability, statistics, and statistical physics, 50. Springer.

Jones, P., Douglas, I., & Jewbali, A., 2013. Modeling Combined Geological and Grade Uncertainty: Application of Multiple-Point Simulation at the Apensu Gold Deposit, Ghana. Mathematical Geosciences, 45(8), 949-965. doi: 10.1007/s11004-013-9500-3 Journel, Andre G. (1974). Geostatistics for conditional simulation of ore bodies. Economic Geology, 69(5), 673-687.

Journel, A.G.,1974. Geostatistics for conditional simulation of ore bodies. Economic Geology, 69(5), 673-687.

Journel, A.G., 1989. Fundamentals of geostatistics in five lessons, Vol. 8. Washington, DC: American Geophysical Union.

Journel, A.G., 1993. Geostatistics: Roadblocks and challenges, Geostatistics Troia'92. Springer, pp. 213-224.

Journel, A.G., 2005. Beyond covariance: The advent of multiple-point geostatistics. Geostatistics Banff 2004, Vols 1 and 2, 14, 225-233 pp.

Journel, A.G. and Alabert, F., 1988. Focusing on spatial connectivity of extreme-valued attributes: Stochastic indicator models of reservoir heterogeneities. SPE paper, 18324.

Journel, A.G. & Alabert, F., 1989. Non-Gaussian data expansion in the earth sciences. Terra Nova, 1(2): 123-134.

Journel, A.G. and Deutsch, C.V., 1993. Entropy and spatial disorder. Mathematical Geology, 25(3): 329-355.

Journel, A.G. and Huijbregts, C.J., 1978. Mining geostatistics, 600. Academic press London.

Keshavan, R.H., Montanari, A. and Oh, S., 2010. Matrix completion from a few entries. Information Theory, IEEE Transactions on, 56(6): 2980-2998.

Keshavan, R.H. and Oh, S., 2009. A gradient descent algorithm on the grassman manifold for matrix completion, technical report, Dept. of Electrical Engineering, Stanford University.

Kiers, H.A., 2000. Towards a standardized notation and terminology in multiway analysis. Journal of chemometrics, 14(3): 105-122.
Kolda, T.G. and Bader, B.W., 2009. Tensor decompositions and applications. SIAM review, 51(3): 455-500.
Koltermann, C.E. and Gorelick, S.M., 1996. Heterogeneity in sedimentary deposits: A review of structure-imitating, process-imitating, and descriptive approaches. Water Resources Research, 32(9): 2617-2658.
Kruskal, J.B., 1989. Rank, decomposition, and uniqueness for 3-way and N-way arrays. Multiway data analysis, Coppi and Bolasco, eds., North-Holland, Amsterdam, 1989,1 7-18.
Lantuéjoul, C., 2002. Geostatistical simulation: Models and algorithms. Springer Verlag.
Lin, Z., Chen, M. and Ma, Y., 2010. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Technical report, UIUC Technical Report UILU-ENG-09-2215, 2009.
Liu, J., Musialski, P., Wonka, P. and Ye, J., 2009. Tensor completion for estimating missing values in visual data, Computer Vision, 2009 IEEE 12th International Conference on. IEEE, pp. 2114-2121.
Liu, J., Musialski, P., Wonka, P. and Ye, J., 2013. Tensor completion for estimating missing values in visual data. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35(1): 208-220.
Machuca-Mory, D.F., Ortiz, J.M., & Deutsch, CV. 2008. On the challenge of using sequential indicator simulation for the estimation of recoverable reserves. International Journal of Mining, Reclamation and Environment, 22(4), 285-299.
Machuca-Mory, D.F., Ortiz J.M., & Dimitrakopoulos, R., 2011. High-order Spatial Simulation of the Structurally Controlled Apensu Gold Deposit, Ghana. In: Deschênes (Editor), Proceedings World Gold, Montreal, Quebec, Canada, pp. 771-783.
MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations, Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. California, USA, pp. 14.
Maharaja, A., 2008. TiGenerator: Object-based training image generator. Computers & Geosciences, 34(12): 1753-1761.
Mao, S. and Journel, A., 1999. Generation of a reference petrophysical/seismic data set. the Stanford V reservoir, Dept. of Petrol. Eng., Stanford University, CA.
Mariethoz, G. and Kelly, B.F., 2011. Modeling complex geological structures with elementary training images and transform-invariant distances. Water Resources Research, 47(7).
Mariethoz, G., Renard, P. and Straubhaar, J., 2010. The direct sampling method to perform multiple-point geostatistical simulations. Water Resources Research, 46(11).
Mariethoz, G., & Renard, P., 2010. Reconstruction of incomplete data sets or images using direct sampling. Mathematical Geosciences, 42(3), 245-268.
Menabde, M., Froyland, G., Stone, P. and Yeates, G., 2004. Mining schedule optimisation for conditionally simulated orebodies, Proceedings of the international symposium on orebody modelling and strategic mine planning: Uncertainty and risk management, pp. 347-52.

Michael, H., Li, H., Boucher, A., Sun, T., Caers, J. and Gorelick, S., 2010. Combining geologic-process models and geostatistics for conditional simulation of 3-D subsurface heterogeneity. Water Resources Research, 46(5).

Michenková, M., 2011. Numerical algorithms for low-rank matrix completion problems. Tech. report, ETH, Switzerland.

Miller, J., Sun, T., Li, H., Stewart, J., Genty, C., Li, D. and Lyttle, C., 2008. Direct Modeling of Reservoirs Through Forward Process-Based Models: Can We Get There?, International Petroleum Technology Conference.

Montiel, L. and Dimitrakopoulos, R., 2013. Stochastic mine production scheduling with multiple processes: Application at Escondida Norte, Chile. Journal of Mining Science, 49(4): 583-597.

Mustapha, H. and Dimitrakopoulos, R., 2010. High-order stochastic simulation of complex spatially distributed natural phenomena. Mathematical Geosciences, 42(5): 457-485.

Mustapha, H. and Dimitrakopoulos, R., 2011. HOSIM: A high-order stochastic simulation algorithm for generating three-dimensional complex geological patterns. Computers & Geosciences, 37(9): 1242-1253.

Mustapha, H., Dimitrakopoulos, R. and Chatterjee, S., 2011. Geologic heterogeneity representation using high-order spatial cumulants for subsurface flow and transport simulations. Water Resources Research, 47(8).

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

Osterholt, V. and Dimitrakopoulos, R., 2007. Resource model uncertainty at the Yandi channel iron deposits, Western Australia - An application of multiple-point simulation for orebody geology. Australasian Inst, M. and Metallurgy, Iron Ore Conference 2007, Proceedings, 147-151 pp.

Paola, C., Mullin, J., Ellis, C., Mohrig, D.C., Swenson, J.B., Parker, G., Hickson, T., Heller, P.L., Pratson, L. and Syvitski, J., 2001. Experimental stratigraphy. GSA TODAY, 11(7): 4-9.

Polak, E., 1971. Computational methods in optimization: a unified approach, 77. Academic Press, New York.

Pyrcz, M. and Strebelle, S., 2006. Event-based geostatistical modeling of deepwater systems, In: Proceedings of the Reservoir Characterization: Integrating Technology and Business Practices: Gulf Coast Section-The Society for Sedimentary Research Twenty-Sixth Annual Research Conference, Houston, USA, pp. 893-922.

Pyrcz, M.J., Boisvert, J.B. and Deutsch, C.V., 2008. A library of training images for fluvial and deepwater reservoirs and associated code. Computers & Geosciences, 34(5): 542-560.

Pyrcz, M.J., Boisvert, J.B. and Deutsch, C.V., 2009. ALLUVSIM: A program for event-based stochastic modeling of fluvial depositional systems. Computers & Geosciences, 35(8): 1671-1685.

Pyrcz, M.J. and Deutsch, C.V., 2004. Stochastic modeling of inclined heterolithic stratification with the bank retreat model, Proceedings of the 2004 Canadian Society of Petroleum Geologists. Canadian Well Logging Society and Canadian Heavy Oil Association Joint Convention (ICE2004), Calgary, Canada.

Ramazan, S. and Dimitrakopoulos, R., 2013. Production scheduling with uncertain supply: a new solution to the open pit mining problem. Optimization and Engineering, 14(2): 361-380.
Remy, N., 2004. S-GeMS: A geostatistical earth modeling library and software, Doctoral dissertation, Ph. D. thesis, Stanford University.
Remy, N., Boucher A. and Wu, 2009. Applied Geostatistics with SGEMS: A User's Guide, Cambridge University Press, 288p.
Renard, P. and Allard, D., 2013. Connectivity metrics for subsurface flow and transport. Advances in Water Resources, 51: 168-196.
Ripley, B.D., 2009. Stochastic simulation, John Wiley & Sons.
Romera-Paredes, B. and Pontil, M., 2013. A New Convex Relaxation for Tensor Completion, Advances in Neural Information Processing Systems, pp. 2967-2975.
Rosenblatt, M., 1985. Stationary sequences and random fields, Birkhauser. Boston, MA, 285p.
Shmaryan, L. and Deutsch, C., 1999. Object-based modeling of fluvial/deepwater reservoirs with fast data conditioning: methodology and case studies, SPE Annual Technical Conference and Exhibition.
Signoretto, M., Dinh, Q.T., De Lathauwer, L. and Suykens, J.A., 2013. Learning with tensors: A framework based on convex optimization and spectral regularization. Machine Learning: 1-49.
Signoretto, M., Van de Plas, R., De Moor, B. and Suykens, J.A., 2011. Tensor versus matrix completion: A comparison with application to spectral data. Signal Processing Letters, IEEE, 18(7): 403-406.
Stoyan, D., 1987. Statistical Analysis of Spatial Point Processes: A Soft-Core Model and Cross-Correlations of Marks. Biometrical Journal, 29(8): 971-980.
Straubhaar, J., Renard, P., Mariethoz, G., Froidevaux, R. and Besson, O., 2011. An improved parallel multiple-point algorithm using a list approach. Mathematical Geosciences, 43(3): 305-328.
Strebelle, S., 2000. Sequential simulation drawing structures from training images, PhD Thesis, Stanford University, 200 p.
Strebelle, S., 2002. Conditional simulation of complex geological structures using multiple-point statistics. Mathematical Geology, 34(1): 1-21.
Strebelle, S., 2005. Multiple-point statistics simulation of continuous variables. GIS and Spatial Analysis, Vol 1and 2, 732-736 pp.
Tan, X., Tahmasebi, P., Caers, J., 2014. Comparing Training-Image Based Algorithms Using an Analysis of Distance. Mathematical Geosciences, 46(2), 149-169.
Tetzlaff, D.M., 1989. Limits to the predictive ability of dynamic models that simulate clastic sedimentation. Quantitative Dynamic Stratigraphy: 55-65.
Toh, K.-C. and Yun, S., 2010. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. Pacific Journal of Optimization, 6(615-640): 15.
Tomioka, R., Suzuki, T., Hayashi, K. and Kashima, H., 2011. Statistical performance of convex tensor decomposition. Advances in Neural Information Processing Systems, pp. 972-980.

Vallee, M., 2000. Mineral resource+ engineering, economic and legal feasibility. CIM bulletin, 93(1038): 53-61.

Warga, J., 1963. Minimizing certain convex functions. Journal of the Society for Industrial & Applied Mathematics, 11(3): 588-593.

Wu, J., 2007. 4D seismic and multiple-point pattern data integration using geostatistics, PhD Thesis, Stanford University.

Wu, J., Zhang, T. and Journel, A., 2008. Fast FILTERSIM simulation with score-based distance. Mathematical Geosciences, 40(7): 773-788.

Xiaojin, T., Tahmasebi, P., & Caers, J. (2014). Comparing Training-Image Based Algorithms Using an Analysis of Distance. Mathematical Geosciences, 46(2), 149-169.

Yahya, W., 2011. Image reconstruction from a limited number of samples: A matrix completion-based approach, M. Eng. Thesis, McGill University, Montreal.

Yahya, W., Dimitrakopoulos, R. and Psaromiligkos, I., 2012. Building training images from hard data: A matrix-completion based approach. COSMO – Stochastic Mine Planning Laboratory, Department of Mining and Materials Engineering. McGill University.

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

Zhang, T., Bombarde, S., Strebelle, S. and Oatney, E., 2006. 3D porosity modeling of a carbonate reservoir using continuous multiple-point statistics simulation. SPE Journal, 11(3): 375-379.