Year :

Multiple point geostatistical approaches to spectrally enhance satellite imagery

Gravey Mathieu

UNIL | Université de Lausanne

Faculté des géosciences et de l'environnement
Institut des dynamiques de la surface terrestre

Multiple point geostatistical approaches to spectrally enhance satellite imagery

Thèse de doctorat

Présentée à la
Faculté des géosciences et de l'environnement
Institut des dynamiques de la surface terrestre
de l'Université de Lausanne

pour l'obtention du grade de
Doctorat en sciences de la Terre

par

Mathieu Gravey

Jury

Prof. Grégoire Mariéthoz, Directeur de thèse
Prof. Niklas Linde, Expert interne
Prof. Jocelyn Chanussot, Expert
Prof. Peter Atkinson, Expert
Dr. Li-Yi Wei, Expert

Sous la présidence de
Prof. Christian Kull, Président du jury

Lausanne, 2020

UNIL | Université de Lausanne
Faculté des géosciences et de l'environnement
bâtiment Géopolis bureau 4631

IMPRIMATUR

Vu le rapport présenté par le jury d'examen, composé de

Président de la séance publique :	M. le Professeur Christian Kull
Président du colloque :	M. le Professeur Christian Kull
Directeur de thèse :	M. le Professeur Grégoire Mariéthoz
Expert interne :	M. le Professeur Niklas Linde
Expert externe :	M. le Professeur Jocelyn Chanussot
Expert externe :	M. le Professeur Peter Atkinson
Expert externe :	M. le Docteur Li-Yi Wei

Le Doyen de la Faculté des géosciences et de l'environnement autorise l'impression de la thèse de

Monsieur Mathieu GRAVEY

Titulaire d'un
Master en informatique
De l'Ecole nationale supérieure des Mines (France)

intitulée

Multiple point geostatistical approaches to spectrally enhance satellite imagery

Lausanne, le 8 janvier 2020

Pour le Doyen de la Faculté des géosciences et de l'environnement

Professeur Christian Kull

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor, Grégoire Mariéthoz for the continuous support during my PhD, for his patience (with my daily new idea, and with the long process of writing), motivation, courage (to hire me knowing my English level), endless enthusiasm, but, first of all, the opportunities. First, the opportunity, to do a PhD. in geoscience – a field which, until then was almost unknown to me – and the opportunity to extend my knowledge in this interesting field. The opportunity to travel to dozens of international conferences to learn, share and debate about science. I could not have imagined having a better advisor and mentor for my PhD.

Besides my advisor, I would like to thank Philippe Montesinos who gave me a taste for research, and for his inspiring coding philosophy: “Let’s code it!”.

I thank my office mates, GAIAlab group members and chocolate group members for stimulating discussions, coffee/tea breaks and for all the fun we have had in the last four years.

My sincere thanks also go to Ehsan Baninajar for all the discussions, jokes, and mutual support, in particular during the paper publishing processes, proposal and thesis writing period.

I need to dedicate special thanks, first to Christophe Raymond, Fabian Guignard and Yann-Vivian Silvert for their time to help me to solve mathematical and algorithmic challenges. To Fabio Oriani, Luiz Gustavo Rasera, and Raphaël Nussbaumer, for all the endless debates about MPS and notations. To Thomas Mejer Hansen, Kirill Gerke, Jef Caers, and the people I met over the year and with whom I shared interesting conversations going from new geostatistical challenges, to the philosophy of science.

Thanks to the Intel teams who were always ready to share resources, answer challenging questions, fix bugs. To the Google Earth Engine team, for resources, collaboration and debates about remote sensing data management, and the endless reciprocal apologies when I crashed their servers (twice!).

A particular thanks to open source communities, Stack Overflow, Wikipedia, GitHub and other well-known website, that save a lot of my time, during coding, writing, reading and accessing papers.

Even if science put distance between us, I would like to thank my family: especially my parents, for giving birth to me in the first place and supporting (in all senses) me over the years!

Last but not the least, I want to thank everyone that helped me with English, Luiz Gustavo Rasera, Ehsan Baninajar, Leanne Phelps, Loni Beutler, Joanne Elkadi, Ruth Fullenwarth and Peter Atkinson.

This PhD was funded by the Swiss National Science Foundation (SNSF grant 200021_162882)

Abstract

The information content of Earth surface satellite images are getting richer and richer. In the long process from gray images on silver film to multispectral digital imagery, lots of different types of image were acquired. Currently images allow doing high quality classification, change detection, etc. The combined use of imagery of different generations is a challenge for long-term studies. The usual solution for a long-term study is to decimate spectral information of satellite imagery to a common level. Instead, in this thesis I propose to use geostatistics and in particular multiple-points statistics (MPS), tools originally developed to simulate subsurface processes using an analogue image (training image), to improve the usefulness of poor satellite imagery by artificially harmonizing their spectral resolution.

Due to the computational and parametrization challenges related to the use of existing MPS approaches for spectral enhancement, a new method was developed. Quantile Sampling (QS) is a robust and efficient solution to realize MPS simulations. Furthermore, QS was designed to be easily set, with few and independent parameters. QS was developed, with the primary constraint to handle continuous values in a particular efficient manner.

The spectral enhancement can be divided into two distinct tasks. First spectral disaggregation, such as converting a gray image into a color image, is addressed with traditional MPS algorithms. Second, the spectral extrapolation, such as determining near infrared from visible color image, is explored using a new framework: Narrow Distribution Selection (NDS) dedicated to this type of task. Built on top of the QS ideas, NDS provides high quality simulations by providing high probability simulations.

Finally, a new and simple calibration framework is presented. Even if QS reduces the sensitivity to algorithm parameterization and simplifies it significantly, calibration is still required. Here, a method to automatically determine an optimal verbatim-free calibration is presented. This method relies on the complete analysis of the training image. Furthermore, the method provides a calibration adapted to each step of the simulation.

Résumé

Les images satellites de la surface de la terrestre sont de plus en plus riches. Au cours de l'évolution qui s'étale de l'imagerie argentique monochromatique jusqu'aux dernières avancées en imagerie numérique multispectrale, de nombreux types d'images différentes ont été acquis. Les images actuelles permettent une classification de hautefidélité, la détection de changements, etc. L'utilisation d'images de différentes générations est un réel défi pour les études qui considère les évolutions sur le long terme. La solution habituelle utilisée consiste à réduire les informations spectrales des images satellitaires à un dénominateur commun. Dans cette thèse je propose d'utiliser la géostatistique et en particulier les statistiques multipoints (MPS), qui ont été développées à l'origine pour simuler des processus souterrains à l'aide d'images analogues (images d'entrainement), afin d'améliorer l'utilité des images satellites pauvres en harmonisant artificiellement leurs résolutions spectrales.

Dans le cadre de l'enrichissement spectral, une nouvelle méthode a été développée en raison des lourdeurs de calcul et de paramétrage liées à l'utilisation des approches MPS existantes. Quantile Sampling (QS) est une solution robuste et efficace pour réaliser des simulations MPS. De plus, QS a été conçu pour être configuré facilement, avec peu de paramètres et des paramètres indépendants. QS a été développé avec comme contrainte principale de gérer les variables continues de manière extrêmement efficace.

L'enrichissement spectral peut être divisé en deux tâches distinctes. La première tâche, la désagrégation spectrale, telle que la conversion d'une image grise en image couleur, peut être traitée à l'aide des algorithmes MPS traditionnels. La deuxième, l'extrapolation spectrale, telle que la reconstruction du proche infrarouge à partir d'une image en couleur, est explorée à l'aide d'une nouvelle approche NDS (Narrow Distribution Selection) spécialement développée pour ce type de tâche. S'appuyant sur les bases de QS, NDS fournit des simulations de haute qualité, grâce à des simulations de forte probabilité.

Enfin, une nouvelle méthode de calibration est présentée. Si QS simplifie le paramétrage et en réduit significativement la sensibilité, une calibration est toujours nécessaire. La méthode présentée permet de déterminer automatiquement une calibration optimale sans verbatim. Cette méthode repose sur l'analyse complète de l'image

d’entrainement. De plus, le procédé fournit une calibration évolutive adaptée à chaque étape de la simulation.

Résumé grand public

Les images satellites de la surface de la terrestre sont de plus en plus riches. À travers les avancées technologiques, depuis l'imagerie argentique en noir et blanc jusqu'aux dernières avancées en imagerie numérique multispectrale, de nombreux types d'images différentes ont été acquis. Les images actuelles permettent une classification de hautefidélité, la détection de changements, etc. L'utilisation d'images de différentes générations est un réel défi pour les études qui considèrent les évolutions sur le long terme. La solution habituelle consiste à réduire leur qualité à un dénominateur commun.

Dans cette thèse, je cherche à l'inverse à enrichir les images satellites pauvres pour rendre leur résolution spectrale similaire aux images plus récentes. Pour ce faire je propose d'utiliser la géostatistique et en particulier les statistiques multipoints (MPS), qui ont été développées à l'origine pour simuler des processus souterrains à l'aide d'images analogues (images d'entrainement)

Afin d'accélérer et simplifier l'utilisation des MPS, une nouvelle méthode a été développée. Quantile Sampling (QS) est une solution robuste, simple et efficace pour réaliser des simulations MPS. QS a été développé avec comme contrainte principale de gérer les variables continues de manière extrêmement efficace.

L'enrichissement spectral peut être effectué à l'aide d'algorithmes standards à condition que la transformation puisse être réversible (ex. gris = moyenne des couleurs ➔ couleurs). Toutefois, dans le contexte de l'extrapolation spectrale (couleur ➔ proche infrarouge) une nouvelle approche NDS (Narrow Distribution Selection) a dû être spécialement développée. S'appuyant sur les bases de QS, NDS fournit des simulations de haute qualité.

Enfin, une nouvelle méthode de calibration est présentée. Si QS simplifie le paramétrage et réduit significativement la sensibilité des paramètres, une calibration est toujours nécessaire. La méthode présentée, qui repose sur l'analyse complète de l'image d'entrainement, se veut être simple, automatique et évolutive.

Contents

1 Introduction

1.1 Satellite imagery

Satellite sensor imagery (formerly called Earth observation imagery) refers to any images of Earth collected from satellite sensors. From the first photograph during the first half of the 19th century humans were interested in taking pictures of the Earth's surface. This interest was revealed starting with the first aerial photograph taken by Nadar in 1858. Nadar (1900) directly sees the potential as he says himself *"Cette œuvre gigantesque du cadastre, avec son armée d'ingénieurs, d'arpenteurs, de chaîneurs, de dessinateurs, de calculateurs, a demandé plus d'un demi-siècle de travail, pour être mal faite. Cette année, je peux, moi tout seul, l'achever en trente jours, et de façon parfaite. Un bon aérostat captif (relié au sol), un bon appareil photographique, voilà mes seules armes."*. The potential is clear even for military purposes *"Les images, agrandies sous les yeux du général en chef lui présenteraient l'ensemble de son échiquier, constatant au fur et à mesure les moindres détails de l'action et lui assurant toute préexcellence pour conduire toute sa partie."* (Nadar, 1900), and that stays a main driver since then. From the time of the first aerial picture, the technique and support did not stop evolving, passing from balloons to pigeons (Hildebrandt, 1907), airplanes (1909) or UAVs (1959). The recording evolves too, from film (silver) to digital, and from gray to color and nowadays multi- and hyper-spectral. The first Earth picture taken from space was in 1946 using a converted V-2 rocket, and the first satellite true imagery was made in 1959. Since then the number of observation satellites has dramatically increases, for all types of application (Campbell and Wynne, 2011): intelligence (spy), meteorology, oceanography, landscape, geology, cartography and for agriculture and biodiversity.

Satellite imagery can be synthesized by the sensor type and an additional four major properties (resolutions) (Campbell and Wynne, 2011):

- Active/passive sensors: passive sensors are like cameras, when we accumulate the response of photons diffused by the environment. Active sensing is like a "flash" on a camera, we first send a signal, then we measure the response. For active sensing usually we emit polarized microwaves, then measure the polarized amplitude and phase of the backscatter signal.

- Spatial resolution: it represents the size covered in real space by a single measure (usually pixels), it can vary from a few km/pixels for weather observers to dm/pixels for the most precise.
- Spectral resolution: it represents the smallest variation in the spectrum that can be differentiated. By extension for satellite sensor imagery, it represents the wavelength on which the sensor is sensitive, usually expressed as a central wavelength and an associated bandwidth.
- Radiometric resolution: it represents the smallest amount of difference of energy (light intensity) that can be quantified. Usually this value is provided as a binary resolution. (e.g., 1 bit for two states, 12 bits for 4096 values)
- Temporal resolution: it represents the time interval between two revisits, of the satellite (by extension constellations), this property is not directly related to sensors, but by the orbit of the sensor.

Each application requires different needs (Campbell and Wynne, 2011) (e.g., meteorology needs fine temporal resolution, land surface cartography and biodiversity studies require fine spectral resolution), some of these requirements can be met satisfactorily but not all. For physical reasons, the amount of light that is received over a sensor from a given position on Earth is limited, and the sensitivity of sensors is limited by technological reasons.

1.1.1 Why the spectrum is important?

The spectral signature is the electromagnetic characteristic of a surface in function of the wavelength (Campbell and Wynne, 2011). Therefore, any surface has a unique spectral signature. Typically, minerals have very characteristic time-invariant signatures (Van der Meer et al., 2012) and vegetation has evolving signatures that are dependent on the type of plant, the age, environment (soil, humidity, sun exposure) (Gnyp et al., 2014; Li et al., 2014; Shendryk et al., 2016) or even diseases (Mahlein et al., 2010).

This unique spectral signature can be used to monitor plant growth and health or to classify vegetation and monitor landscape evolution (Hansen et al., 2008). The spectral signature can be used easily to realize geological surveys, and determine the type of minerals.

"Statistical thinking will one day be as necessary for efficient citizenship as the ability to read or write."

H. G. Wells

1.2 Geostatistics

1.2.1 From statistics to simulations

Statistics is becoming a standard tool not only for research but also for daily life. Inferential statistics are important for probing the unknown, because the data of interest are not available (forecasting, or study of past events), or is unreachable (technical or economic reasons). Statistics are needed to have a clear description and quantification of potential scenarios. The most probable case is not enough. If we imagine a flood risk of 10%, the most probable scenario is that there won't be a flood. However, 1 out of 10 will in fact have a flood. If the uncertainty around a prediction is important, it is even more important to propagate it through any forward step such as the destructive cost of a flood, which is highly nonlinear. (No flood → no cost, small flood → expensive, medium flood → maximum cost, biggest flood → maximum cost too).

Estimation refers to the probability that an event happens on a given location given known information. For the situation presented before we have a 10% chance of a flood. Simulation refers to scenarios, for example, if we have 10 villages close to each other, even if each village has a 10% chance to be flooded, the probability of a given village to be flooded, knowing that another village is flooded, is significantly higher. If we look at 10 events, two extreme cases are possible: each time a single village will be flooded (case 1), or 9 times no village will be flooded and one time all the villages will be submerged (case 2). Simulation will create scenarios that respect this relation, taking into account probability of one village be submerged knowing the status of each other. In short, estimation give statistics about the occurrence of an event, independently of all others, when simulation provides complete scenarios taking into account relations between elements.

Simulations are especially critical in cases of nonlinear transfer functions. Transfer functions are specific functions that takes a simulation entry and provides an output. We can

easily imagine the example of the flood water runoff model. But non-liner transfer function can be simple, like computing a perimeter or simply counting objects. If we considered a function cost for the evacuation like $cost = \sqrt{number\ of\ villages\ to\ evacuate}$, then in case 1 we have a cumulated cost of 10 and in case 2 only $\sqrt{10}$. It could be interesting to have a product in between estimation and simulation, such as the most probable simulation.

1.2.2 Traditional geostatistics

Traditional geostatistics was born at the end of the 1950s from the motivation to predict high ore concentration in mines. Formalized by Matheron (Matheron, 1973), kriging is the best linear unbiased estimator. Kriging is based on covariances between a pair of points, which means the method uses only the similarity between values separated by a given distance and direction. This mathematical approach is restricted to numerous hypotheses such as stationarity or a restricted number of "valid" models. Multivariate Gaussian stochastic simulations were developed based on kriging to get simulations. Many solutions have been proposed over the years to generate Gaussian simulations such as spectral and convolution approaches (Borgman et al., 1984; Le Ravalec et al., 2000), turning bands, or LU decomposition (Davis, 1987). The best known and the most straightforward solution is probably Sequential Gaussian Simulation (SGS) (Dimitrakopoulos and Luo, 2004), which at each uninformed location following a path, computes the distance between previously informed locations and the location of interest. Then it estimates the mean and variance, and samples a value according to these parameters.

1.2.3 Multiple point geostatistics

Twenty-five years ago, Extended Normal Equation Simulation (ENESIM) (Guardiano and Srivastava, 1993), the first MPS method, was developed, based on the sequential simulation approaches used by SGS. The breakthrough idea was to not describe relations between values using a mathematical model, but to use a so-called training image to infer relations. ENESIM is only targeting categorical variables, for each pixel to simulate along the path, the neighborhood is extracted then similar patterns are searched in the training images. After exhaustive recordings of all the matches, an exhaustive probability distribution is computed, then a value is sampled according to it. Even if the method is very slow due to explicit the exploration of the training image, for each pixel to simulate, it stays the inspiration

of all modern MPS algorithms. Single Normal Equation Simulation (SNESIM) (Strebelle, 2002) is the first viable algorithm. The main idea is to compute in advance and store in an adapted structure the conditional probability for all the possible patterns, like a tree structure that allows to significantly speedup the simulation. However, this approach increases the demandes of memory. To limit the memory consumption, a multigrid path was developed and only patterns at each given grid level are stored (Strebelle and Cavelius, 2013). Following the evolution of computing technology and parallelization in particular, IMPALA (Straubhaar et al., 2011) was developed, using a list rather than a tree, to facilitate parallel exploration and limit memory requirements. Finally, an hybrid solution was designed by combining both strategies, resulting in tree-list structures. Based on ENESIM that has a low memory footprint, Direct Sampling (DS) (Mariethoz et al., 2010) was developed using the idea that we do not need to explore the complete training image to get a sample of the conditional distribution. If the exploration is random, we can use the first sample found. This approach drastically reduces the computation time with respect to ENSESIM, but still keeps a major advantage of it: a limited memory need. Because DS does not explore the entire training image, the best matches stay unknown; therefore, it requires a new parameter, the “threshold”, to define what represents a match (Meerschman et al., 2013). This innovative approach allows DS to handle continuous variables, where usually no perfect match exists. Pixel-based methods manage very well conditioning data (values known at specific locations), by considering these values as early simulated pixels. In parallel to the development of DS and IMPALA, some patch-based solutions were developed over the last decades (Mariethoz and Caers, 2014). The major difference between the patch-based and the early pixel-based MPS is that a complete patch is simulated at each step of the algorithm. These approaches provide significant speed-up, but this gain comes with a major drawback: the junction between patches generate artifacts in the final results. These artifacts (usually straight lines) are a result of the patch design (usually a rectangle) and have nothing to do with the simulated variable. This can produce major issues in the transfer function. A whole subfield of MPS is dedicated to the best way to find the optimal patch and how to handle the merging in the best way (both steps are dependent). This research has resulted in different algorithms such as Simulation of Pattern (SIMPAT) (Arpat and Caers, 2007), Image quilting (IQ) (Mahmud et al., 2014), Graph-cut (X. Li et al., 2016), or Cross-Correlation based Simulation (CCSIM) (Tahmasebi et al., 2012). Patch based

methods tend predominantly to use a row-path strategy and struggle to incorporate conditioning data, especially for dense conditioning fields.

1.2.4 Similarities and differences between computer sciences and geostatistics

Many similarities can be observed between geostatistics and computer sciences (Mariethoz and Lefebvre, 2014). The first and maybe the most surprising is the Gaussian process that is similar to Kriging, both being developed relatively independently. MPS can be expressed as a texture synthesis problem, and in the case of simulations with conditioning data, as an inpainting or more precisely a sparse inpainting problem. ENESIM was developed based on an algorithm made to realize optical character recognition (ORC) of digits (Srivastava, 2018). The development of SNESIM followed a similar challenge from texture synthesis and patch-based approaches that were developed based on texture synthesis algorithms (Wei and Levoy, 2000). The proximity between these algorithms and random forest is remarkable, the only difference being that we keep the complete distribution, instead of an aggregate mean value. If similar research happens in both fields, the interest and major requirements are relatively different. In texture synthesis the focus is primarily on high quality and computational efficiency. These aspects are important in geosciences too, but the main focus is still on variability and handling of conditioning data, something that is at most of secondary importance in computer sciences. An interesting aspect of this duality can be seen in the fact that geostatistics focuses on the value and for historical reasons emphasize similarities (covariance and variogram), while computer science sometime does not use the value but only focuses on differences, gradients and changes, to be more robust to uninteresting variations (light, sensors).

1.3 Motivation: Why do spectral enhancement?

The main motivation for this work is to provide a tool to achieve automatic spectral enhancements. The main interest of this task is to align spectral signatures of two images. This is required if we have a standard processing pipeline adapted to a given sensor (e.g., classification process, change detection, evolution quantification). This processing is developed for a given sensor. If we change sensors, we need to recalibrate all the processing pipeline, or even redesign it because some bands are missing. Most modern processing is still done using only spectral signatures at each location, while the information in the spatial structure is ignored (Plaza et al., 2009; Tadjudin and Landgrebe, 1998; Varshney and Arora,

2013). Textural information can be very interesting in cases of poor spectral resolution. Usually, this task requests a complete redesign of the processing pipeline (Houborg and McCabe, 2018; Mandanici and Bitelli, 2016). Here, to capitalize on preexisting work, we propose to transform spatial information in spectral information using MPS (Figure 1.1).

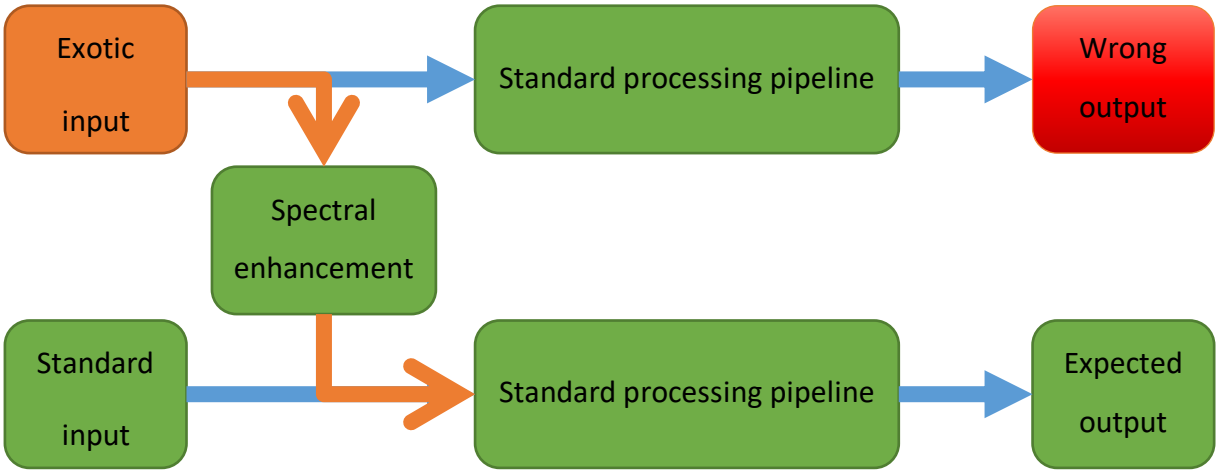

Figure 1.1 Using spectral enhancement to capitalize on standard processing pipeline. Exotic inputs refer any inputs for which the processing pipeline was not designed for.

Incrising the spectral (and radiometric) resolution requests a redesign of the sensor, when for temporal resolution we can just take measures more often (typically with a constellation of satellites) or for spatial resolution, we can move to a lower orbit, for example moving from an 800 km to a 400 km orbit increases spatial resolution by 4 (2 in each dimension) but increases speed by less than 3%. Therefore, even if algorithms developed in this thesis can be used to increase the spatial resolution, spectral resolution is the main challenge of this thesis.

The final output of this work is to develop and provide simple, efficient and automatic solutions that can be used by expert and non-expert practitioners.

1.4 Mathematical background /Annexes

This section introduces briefly some mathematical operator that would be use in the in the following chapters.

1.4.1 Convolution

The convolution $*$ is mathematically defined by:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(\tau) g(x - \tau) d\tau \qquad (1.1)$$

And in the discrete case by

$$(f * g)(x) = \sum_{\tau=-\infty}^{\infty} f(\tau)g(x-\tau) \tag{1.2}$$

With f and g two functions of x.We can notice that the convolution is commutative.

1.4.2 Cross-correlation

The cross-correlation $\star$ can be defined from the convolution as following:

$$(f \star g)(x) = \int_{-\infty}^{\infty} f(\tau)g(x+\tau)d\tau \tag{1.3}$$

and

$$(f \star g)(x) = \sum_{\tau=-\infty}^{\infty} f(\tau)g(x+\tau) \tag{1.4}$$

With f and g two functions of x.

The relation with the convolution is clear,

$$(f \star g) = (f * h) \tag{1.5}$$

with $\forall x, h(x) = g(-x)$

1.4.3 Cross-covariance

The cross-covariance is a cross-correlation with the mean subtracted

$$\mathrm{Cov}(f,g)(x) = \sum_{\tau=-\infty}^{\infty} \big(f(\tau) - \mathrm{mean}(f)\big)\big(g(x+\tau) - \mathrm{mean}(g)\big) \tag{1.6}$$

1.4.4 Fourier transform and Fast Fourier transform (FFT)

The Fourier transform $\mathcal{F}$ allow to decompose a signal as a series of frequencies with associated amplitudes.

$$\mathcal{F}\{f\}(\xi) = \int_{-\infty}^{\infty} f(x)e^{-2\pi i x\xi}d\xi \tag{1.7}$$

With ξ a real number.

It is possible to derivate the convolution theorem for convolutions:

$$\mathcal{F}\{f * g\} = \mathcal{F}\{f\}.\mathcal{F}\{g\} \tag{1.8}$$

And for cross-correlation:

$$\mathcal{F}\{f \star g\} = \overline{\mathcal{F}\{f\}}.\mathcal{F}\{g\} \tag{1.9}$$

With $\bar{x}$ the conjugate of x

1.5 Thesis structure

During my PhD, I addressed the question of spectral enhancement of remote sensed imagery using MPS. Early results using available MPS strategies shows great results, but the integration of available solutions in a more complex framework, resulted in a method so complex to parametrize that it is impossible to calibrate. Therefore, I need first to simplify available methods to reduce the number of tunable parameters. This was done in accord with the bigger framework required to enhance satellite imagery. The development of Quantile Sampling (QS) provides significant simplification and speedup with respect to existing techniques. The QS method is exposed in Chapter 2[1]. QS uses a mismatch map to open a new possibility to estimate conditional probability estimation. Chapter 3 benchmarks available MPS solutions to realize enhancements and provide a new solution (NDS) in-between estimation and simulation[2]. QS significantly simplifies the parametrization; however, a parametrization is still required. Chapter 4 presents a simple solution to efficiently predict optimal parameters without using an optimization that requires complex spatial functions. Chapter 5 conclude the thesis and open the discussion about future work.

[1] inspired form the paper DOI: 10.5194/gmd-2019-211, the paper submitted in Geoscientific Model Development
[2] This work is published in ISPRS Journal of Photogrammetry and Remote Sensing under the paper DOI: 10.1016/j.isprsjprs.2018.11.003

2 Quantile Sampling: a robust and simplified pixel-based multiple-point simulation approach

Mathieu Gravey[a], Grégoire Mariethoz[a]
[a] University of Lausanne, Faculty of Geosciences and Environment, Institute of Earth Surface Dynamics, Switzerland

Highlights

- A new approach is proposed for pixel-based multiple-point geostatistics simulation.
- The method is flexible and straightforward to parametrize.
- It natively handles continuous and multivariate simulations.
- High computational performance with predictable simulation times.
- A free and open-source implementation is provided.

Abstract

Multiple-point geostatistics enable the realistic simulation of complex spatial structures by inferring statistics from a training image. These methods are typically computationally expensive and require complex tuning of algorithmic variables. The approach that is presented in this paper is easier to use than existing algorithms, as it requires few independent algorithmic parameters. It is natively designed for handling continuous variables, and quickly implemented by capitalizing on standard libraries. The algorithm can handle incomplete training images of any dimensionality, with categorical or/and continuous variables, and stationarity is not explicitly required. It is possible to perform unconditional or conditional simulations, even with exhaustively informed covariates. The method provides new degrees of freedom by allowing kernel weighting for pattern matching. Computationally, it is adapted to modern architectures and runs in constant time. The approach is benchmarked against a state-of-the-art method. An efficient open-source implementation of the algorithm is released and can be found here (https://github.com/GAIA-UNIL/G2S), to promote reuse and further evolution.

Keywords

Multiple-point statistics, stochastic simulation, continuous variable, training image, cross-correlation, Fourier transform.

2.1 Introduction

Geostatistics is used widely to generate stochastic random fields for modeling and characterizing spatial phenomena such as Earth surface features and geological structures. Commonly used methods, such as the sequential Gaussian simulation (Gómez-Hernández and Journel, 1993) and turning bands algorithms (Matheron, 1973), are based on kriging (Graeler et al., 2016; J. Li and Heap, 2014; Tadić et al., 2017). This family of approaches implies spatial relations using exclusively pairs of points and expresses these relations using covariance functions. In the last two decades, multiple point statistics (MPS) emerged as a method for representing more complex structures using high-order nonparametric statistics (Guardiano and Srivastava, 1993). To do so, MPS algorithms rely on training images, which are images with similar characteristics to the modeled area. Over the last decade, MPS has been used for stochastic simulation of random fields in a variety of domains such as geological modeling (e.g. Barfod et al., 2018; Strebelle et al., 2002), remote sensing data processing (Gravey et al., 2019; e.g. Yin et al., 2017), stochastic weather generation (Oriani et al., 2017; e.g. Wojcik et al., 2009), geomorphological classification (e.g. Vannametee et al., 2014) and climate model downscaling (a domain that has typically been the realm of kriging-based methods (Bancheri et al., 2018; e.g. Jha et al., 2015; Latombe et al., 2018)).

In the world of MPS simulations, one can distinguish two types of approaches. The first category is the patch-based methods, where complete patches of the training image are imported into the simulation. This category includes methods such as SIMPAT (Arpat and Caers, 2007) and DISPAT (Honarkhah and Caers, 2010), which are based on building databases of patterns, and image quilting (Mahmud et al., 2014), which uses an overlap area to identify patch candidates, which are subsequently assembled using an optimal cut. CCSIM (Tahmasebi et al., 2012) uses cross-correlation to rapidly identify optimal candidates. More recently, Li et al. (2016) proposed a solution that uses graph-cuts to find an optimal cut between patches, which has the advantage of operating easily and efficiently independently of the dimensionality of the problem. Tahmasebi (2017) propose as a solution that is based on “warping” in which the new patch is distorted to match the previously simulated areas. For a multivariate simulation with an informed variable, Hoffimann (2017) presented an approach for selecting a good candidate based on the mismatch of the primary variable, and on the mismatch rank of the candidate patches for auxiliary variables. Although patch-based

approaches are recognized to be fast, they typically suffer from a lack of variability due to the pasting of large areas of the training image, which is a phenomenon that is called verbatim copy. Furthermore, patch-based approaches are typically difficult to use in the presence of dense conditioning data.

The second category of MPS simulation algorithms consists of pixel-based algorithms, which import a single pixel at the time instead of full patches. These methods are typically slower than patch-based methods. However, they do not require a procedure for the fusion of patches, such as an optimal cut, and they allow more flexibility in handling conditioning data. Furthermore, in contrast to patch-based methods, pixel-based approaches rarely produce artifacts when dealing with complex structures. The first pixel-based MPS simulation algorithm was ENESIM (Guardiano and Srivastava, 1993), where for a given categorical neighborhood – usually small – all possible matches in the training image are searched. The conditional distribution of the pixel to be simulated is estimated based on all matches, from which a value is sampled. This approach could originally handle only a few neighbors and a relatively small training image; otherwise, the computational cost would become prohibitive and the number of samples insufficient for estimating the conditional distribution. Inspired by research in computer graphics, where similar techniques are developed for texture synthesis (Mariethoz and Lefebvre, 2014), an important advance was the development of SNESIM (Strebelle, 2002), which proposes storing in advance all possible conditional distributions in a tree structure and using a multigrid simulation path to handle large structures. With IMPALA, Straubhaar (2011) proposed reducing the memory cost by storing information in lists rather than in trees. Another approach is direct sampling (DS) (Mariethoz et al., 2010), where the estimation and the sampling of the conditional probability distribution are bypassed by sampling directly in the training image, which incurs a very low memory cost. DS enabled the first use of pixel-based simulations with continuous variables. DS can use any distance formulation between two patterns; hence, it is well suited for handling various types of variables and multivariate simulations.

Despite, its advantages, DS has several shortcomings: DS requires a threshold – which is specified by the user – that enables the algorithm to differentiate good candidate pixels in the training image from bad ones based on a predefined distance function. DS can be highly sensitive to this threshold, that stay difficult to determine and often dramatically affects the computation time. This results in unpredictable computation times, as demonstrated by

Meerschman (2013). DS is based on the strategy of randomly searching the training image until a good candidate is identified (Shannon, 1948). This feature is an advantage of DS; however, it can also be seen as a weakness in the context of modern computer architectures. Indeed, random memory access and high conditionality can cause 1) suboptimal use of the instruction pipeline, 2) poor memory prefetch, 3) substantial reduction of the useful memory bandwidth and 4) impossibility of using vectorization (Shen, 2013). While the first two problems can be addressed with modern compilers and pseudorandom sequences, the last two are inherent to the current memory and CPU construction.

This paper presents a new and flexible pixel-based simulation approach named Quantile Sampling (QS), which makes efficient use of modern hardware. Our method takes advantage of the possibility of decomposing the standard distance metrics that are used in MPS (L^0, L^2) as sums of cross-correlations. As a result, we can use fast Fourier transforms (FFTs) to quickly compute mismatch maps. To rapidly select candidate patterns in the mismatch maps, we use an optimized partial sorting algorithm. A free, open-source and flexible implementation of QS is available, which is interfaced with most common programming languages (C/C++, MATLAB, R, and Python 3).

The remainder of this paper is structured as follows: Section 2.2 presents the proposed algorithm with an introduction to the general method of sequential simulation, the mismatch measurement using FFTs and the sampling approach of using partial sorting followed by methodological and implementation optimizations. Section 2.3 evaluates the approach in terms of quantitative and qualitative metrics via simulations and conducts benchmark tests against DS, which is the only other available approach that can handle continuous pixel-based simulations. Section 2.4 discusses the strengths and weaknesses of QS and provides guidelines. Finally, the conclusions of this work are presented in Section 2.5.

2.2 Methodology and Implementation

2.2.1 Pixel-based sequential simulation

We recall the main structure of pixel-based MPS simulation algorithms (Mariethoz and Caers, 2014, p. 156), which is summarized and adapted for QS in Pseudocode 1. The key difference between existing approaches is in lines 3 and 4 of , when candidate patterns are selected. This task is the most time-consuming in many MPS algorithms and we focus on computing it in a way that reduces its cost and minimizes the number of free parameters.

Pseudocode 1: QS Algorithm

Inputs:
T the training images
S the simulation grid, including the conditioning data
P the simulation path
The choice of pattern metric

1 **For** each unsimulated pixel x following the path P:
2 Find the neighborhood $N(x)$ in S that contains all previously simulated or conditioning nodes with in a specified radius
3 Compute the mismatch map between T and $N(x)$: Section 2.2.3
4 Select a good candidate using quantile sorting over the mismatch map: Section 2.2.4
5 Assign the value of the selected candidate to x in S
6 End

2.2.2 Decomposition of common mismatch metrics as sums of products

Distance-based MPS approaches are based on pattern matching (Mariethoz and Lefebvre, 2014). Here, we rely on the observation that many common matching metrics can be expressed as weighted sums of the pixelwise mismatch ε. This section explores the pixelwise errors for a single variable and for multiple variables. For a single variable, the mismatch metric ε between two pixels is the distance between two scalars or two classes. In the case of many variables, it is a distance between two vectors that are composed by scalars, by classes, or by a combination of the two. Here, we focus on distance metrics that can be expressed in the following form:

$$\varepsilon(a,b) \propto \sum_j f_j(a).g_j(b) \tag{2.1}$$

where a and b represent the values of two univariate pixels and f_j and g_j are functions that depend on the chosen metric. Here, we use the proportion symbol because we are interested in relative metrics rather than absolute metrics, namely, the objective is to rank the candidate patterns. We show below that many of the common metrics or distances that are used in MPS can be expressed as Equation (2.1).

For the simulation of continuous variables, the most commonly used mismatch metric is the L^2-norm, which can be expressed as follows:

$$\varepsilon_{L^2}(a,b) = (a-b)^2 = a^2 - 2ab + b^2 \tag{2.2}$$

Using Equation (2.1), this L^2-norm can be decomposed into the following series of functions f_j and g_j:

$$f_0: x \to x^2 \qquad g_0: x \to 1$$
$$f_1: x \to -2x \qquad g_1: x \to x$$
$$f_2: x \to 1 \qquad g_2: x \to x^2$$

A similar decomposition is possible for the L^0 -norm (also called Hamming distance), which is commonly used for the simulation of categorical variables. This measure of dissimilarity counts the number of nonzero values in a vector (Hamming, 1950)

$$\varepsilon_{L^0}(a,b) = (a-b)^0 = \sum_{j \in \mathcal{C}} 1 - \left(\delta_{a,j}.\delta_{b,j}\right) \propto \sum_{j \in \mathcal{C}} \delta_{a,j}.\delta_{b,j} \qquad (2.3)$$

where $\delta_{x,y}$ is the Kronecker delta between x and y, which is equal to 1 if x equals y and 0 otherwise, and $\mathcal{C}$ is the set of all possible categories of a specified variable.

Using Equation (2.1), this L^0 distance can be decomposed (Arpat and Caers, 2007) into the following series of functions f_j and g_j :

$$f_j: x \to -\delta_{xj}$$
$$g_j: x \to \delta_{xj}$$

with a new pair of f_j and g_j for each class j of $\mathcal{C}$.

For multivariate pixels, such as a combination of categorical and continuous values, the mismatch ε can be expressed as a sum of univariate pixelwise mismatches.

$$\varepsilon(\boldsymbol{a}, \boldsymbol{b}) \propto \sum_i \sum_j f_j(a_i).g_j(b_i) \qquad (2.4)$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ are the compared vectors and a_i and b_i are the individual components of $\boldsymbol{a}$ and $\boldsymbol{b}$.

2.2.3 Computation of a mismatch map for an entire pattern

The approach that is proposed in this work is based on computing a mismatch map in the TI for each simulated pixel. The mismatch map is a grid that represents the pattern-wise mismatch for each location of the training image and enables the fast identification of a good candidate, as shown by the red circle in Figure 2.1.

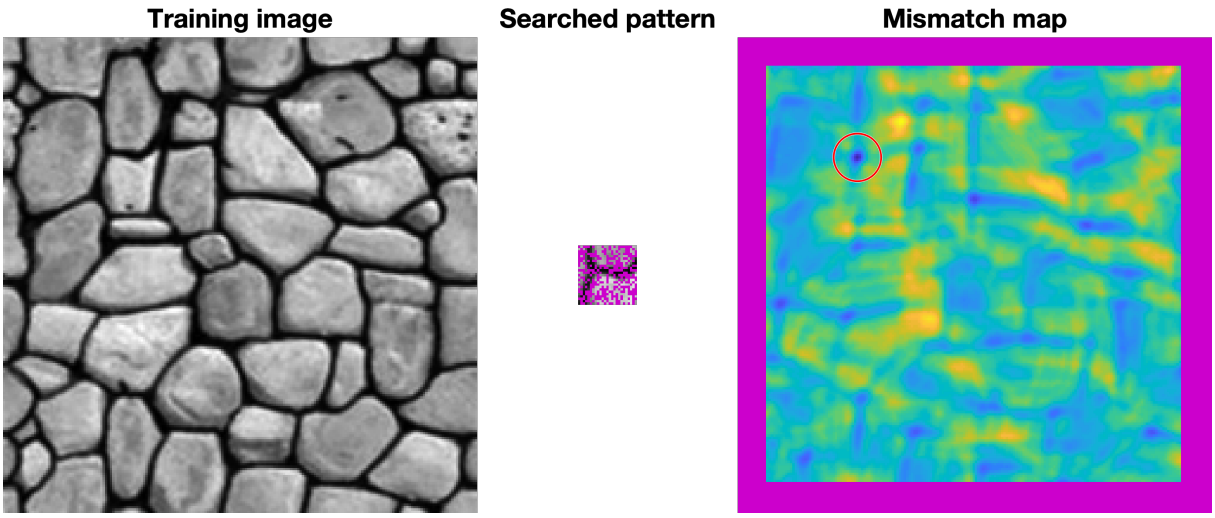

Figure 2.1 Example of a mismatch map for an incomplete pattern. Blue represents good matches and yellow bad matches. The red circle highlights the minimum of the mismatch map, which corresponds to the location of the best candidate.

If we consider the neighborhood $N(s)$ around the simulated position s, then we can express a weighted dissimilarity between $N(s)$ and a location in the TI $N(t)$:

$$\mathrm{E}\big(N(t), N(s)\big) = \sum_{\boldsymbol{l} \mid N_{\boldsymbol{l}}(t) \text{ and } N_{\boldsymbol{l}}(s) \text{ exist}} \omega_{\boldsymbol{l}} \varepsilon\big(N_{\boldsymbol{l}}(t), N_{\boldsymbol{l}}(s)\big) \tag{2.5}$$

where $N_{\boldsymbol{l}}(p)$ is the ensemble of neighbors of p (p can represent either s or t), $\boldsymbol{l}$ is the lag vector that defines the relative position of each value within N, and $\omega_{\boldsymbol{l}}$ is a weight for each pixelwise error according to the lag vector $\boldsymbol{l}$. By extension, ω is the matrix of all weights, which we call the weighting kernel or, simply, the kernel. E represents the mismatch between patterns that are centered on s and $t \in T$, where T is the training image.

Some lags may not correspond to a value, for example, due to edge effects in the considered images or because the patterns are incomplete. Missing patterns are inevitable during the course of a simulation using a sequential path. Furthermore, in many instances, there can be missing areas in the training image. This is addressed by creating an indicator variable to be used as a mask, which equals 1 at informed pixels and 0 everywhere else:

$$\mathbb{1}_{\boldsymbol{l}}(p) = \begin{cases} 1 \; if \; N_{\boldsymbol{l}}(p) \; is \; informed \\ 0 \; otherwise \end{cases} \tag{2.6}$$

Let us first consider the case in which for a specified position, either all or no variables are informed. Expressing the presence of data as a mask enables the gaps to be ignored because the corresponding errors are multiplied by zero.

Then, Equation (2.5 can be expressed as follows:

$$\mathrm{E}\big(N(t), N(s)\big) = \sum_{\boldsymbol{l}} \omega_{\boldsymbol{l}} . \mathbb{1}_{\boldsymbol{l}}(t) . \mathbb{1}_{\boldsymbol{l}}(s) . \varepsilon\big(N_{\boldsymbol{l}}(t), N_{\boldsymbol{l}}(s)\big). \tag{2.7}$$

Combining Equation (2.4) and Equation (2.7), we get:

$$\mathrm{E}\big(N(t),N(s)\big) \propto \sum_{\boldsymbol{l}} \omega_{\boldsymbol{l}}.\mathbb{1}_{\boldsymbol{l}}(t).\mathbb{1}_{\boldsymbol{l}}(s) \sum_{j}\sum_{i} f_j(N_{\boldsymbol{l}}(t)_i).g_j(N_{\boldsymbol{l}}(s)_i) \tag{2.8}$$

$$= \sum_{\boldsymbol{l}}\sum_{j}\sum_{i} \omega_{\boldsymbol{l}}.\mathbb{1}_{\boldsymbol{l}}(t).\mathbb{1}_{\boldsymbol{l}}(s).f_j(N_{\boldsymbol{l}}(t)_i).g_j(N_{\boldsymbol{l}}(s)_i)$$

$$= \sum_{i}\sum_{j}\sum_{\boldsymbol{l}} \omega_{\boldsymbol{l}}.\Big(\mathbb{1}_{\boldsymbol{l}}(t).f_j(N_{\boldsymbol{l}}(t)_i)\Big).\Big(\mathbb{1}_{\boldsymbol{l}}(s).g_j(N_{\boldsymbol{l}}(s)_i)\Big).$$

After rewriting and reordering, Equation (2.8) can be expressed as a sum of cross-correlations that encapsulate spatial dependencies:

$$\mathrm{E}\big(N(t),N(s)\big) \propto \sum_{i}\sum_{j} \Big(\mathbb{1}(t) \circ f_j(N(t)_i)\Big) \star \Big(\omega \circ \mathbb{1}(s) \circ g_j(N(s)_i)\Big), \tag{2.9}$$

where ω and $\mathbb{1}(.)$ represent the matrices that are formed by $\omega_{\boldsymbol{l}}$ and $\mathbb{1}_{\boldsymbol{l}}(.)$ for all possible vectors $\boldsymbol{l}$, $\star$ denotes the cross-correlation operator, and $\circ$ is the element-wise product (or Hadamard-product).

Finally, by applying cross-correlations for all positions $t \in T$, we obtain a mismatch map, which is expressed as:

$$\mathrm{E}\big(T,N(s)\big) \propto \sum_{i}\sum_{j} \Big(\mathbb{1}(T) \circ f_j(T_i)\Big) \star \Big(\omega \circ \mathbb{1}(s) \circ g_j(N(s)_i)\Big). \tag{2.10}$$

The term $\mathbb{1}(T)$ allows the consideration of the possibility of missing data in the training image T.

Let us consider the general case in which only some variables are informed and the weighting can vary for each variable. Equation (2.10 can be extended for this case by defining separate masks and weights ω_i for each variable:

$$\mathrm{E}\big(T,N(s)\big) \propto \sum_{i}\sum_{j} \Big(\mathbb{1}(T_i) \circ f_j(T_i)\Big) \star \Big(\omega_i \circ \mathbb{1}(s_i) \circ g_j(N(s)_i)\Big). \tag{2.11}$$

Equation (2.11) can be expressed using the convolution theorem:

$$\mathrm{E}\big(T,N(s)\big) \propto \sum_{i}\sum_{j} \mathcal{F}^{-1}\Big\{\overline{\mathcal{F}\big\{\mathbb{1}(T_i) \circ f_j(T_i)\big\}} \circ \mathcal{F}\big\{\omega_i \circ \mathbb{1}(s_i) \circ g_j(N(s)_i)\big\}\Big\}. \tag{2.12}$$

where $\mathcal{F}$ represents the Fourier transform, $\mathcal{F}^{-1}$ the inverse transform, and $\bar{x}$ the conjugate of x.

By linearity of the Fourier transform, the summation can be performed in Fourier space, thereby reducing the number of transformations:

$$\mathrm{E}\big(T, N(s)\big) \propto \mathcal{F}^{-1}\left\{\sum_i \sum_j \overline{\mathcal{F}\{\mathbb{1}(T_i) \circ f_j(T_i)\}} \circ \mathcal{F}\{\omega_i \circ \mathbb{1}(s_i) \circ g_j(N(s)_i)\}\right\}. \quad (2.13)$$

Equation (2.13) is appropriate for modern computers, which are well-suited for computing FFTs (Cooley et al., 1965; Gauss, 1799). Currently, FFTs are well implemented in highly optimized libraries (Rodríguez, 2002). Equation (2.13) is the expression that is used in our QS implementation because it reduces the number of Fourier transforms, which are the most computationally expensive operations of the algorithm. One issue with the use of FFTs is that the image T is typically assumed to be periodic. However, in most practical applications, it is not periodic. This can be simply addressed by cropping the edges of $E\big(T, N(s)\big)$ or by adding a padding around T.

The computation of the mismatch map (Equation (2.13)) is deterministic; as a result, it incurs a constant computational cost that is independent of the pixel values. Additionally, Equation (2.13) is expressed without any constraints on the dimensionality. Therefore, it is possible to use the *n*-dimensional FFTs that are provided in the above libraries to perform *n*-dimensional simulations without changing the implementation.

2.2.4 Selection of candidates based on a quantile

The second contribution of this work is the k-sampling strategy for selecting a simulated value among candidates. The main idea is to use the previously calculated mismatch map to select a set of potential candidates that are defined by the k smallest (i.e. a quantile) values of E. Once this set has been selected, we draw randomly a sample from this pool of candidates. This differs from strategies that rely on a fixed threshold, which can be cumbersome to determine. This strategy is highly similar to the ε-replicate strategy that is used in image quilting (Mahmud et al., 2014) in that we reuse and extend to satisfy the specific requirements of QS. It has the main advantage of rescaling the acceptance criterion according to the difficulty; i.e. the algorithm is more tolerant of rare patterns while requiring very close matches for common patterns.

In detail, the candidate selection procedure is as follows: All possible candidates are ranked according to their mismatch and one candidate is randomly sampled among the k best. This number k can be seen as a quantile over the training dataset. However, parameter k has the advantage of being an easy representation for users, who can associate $k = 1$ with the

best candidate, $k = 2$ with the two best candidates, etc. For fine-tuning parameter k, the sampling strategy can be extended to non-integer values of k by sampling the candidates with probabilities that are not uniform. For example, if the user sets $k = 1.5$, the best candidate has a probability of 2/3 of being sampled and the second best a probability of 1/3. For $k = 3.2$, (Figure 2.2) each of the 3 best candidates are sampled with an equal probability of 0.3125 and the 4th best with a probability of 0.0625. This feature is especially useful for tuning k between 1 and 2 and for avoiding a value of $k = 1$, which can result in the phenomenon of verbatim copy.

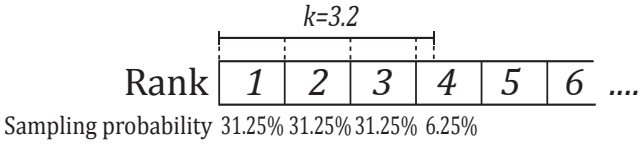

Figure 2.2 Illustration of the k-sampling strategy

An alternative sampling strategy for reducing the simulation time is presented in Appendix 2.8.3. However, this strategy can result in a reduction in the simulation quality.

2.2.5 Simplifications in the case of a fully informed training image

In many applications, spatially exhaustive TIs are available. In such cases, the equations above can be simplified by dropping constant terms from Equation (2.1), thereby resulting in a simplified form for Equation (2.13). Here, we take advantage of the ranking to know that a constant term will not affect the result.

As in Tahmasebi (2012), in the L^2-norm, we drop the squared value of the searched pattern, namely, b^2, from Equation (2.2). Hence, we can express Equation (2.4) as follows:

$$\varepsilon(\boldsymbol{a}, \boldsymbol{b}) = \sum_i a_i^2 - 2 \sum_i a_i . b_i \qquad (2.14)$$

The term a^2, which represents the squared value of the candidate pattern in the TI, differs among training image locations and, therefore, cannot be removed. Indeed, the assumption that $\sum a^2$ is constant is only valid under a strict stationarity hypothesis on the scale of the search pattern. While this hypothesis might be satisfied in some cases (as in Tahmasebi et al., 2012), we do not believe it is generally valid. Via the same approach, Equation 3 can be simplified by removing the constant terms; then, we obtain the following for the L^0-norm:

(2.15)

$$\varepsilon(\boldsymbol{a},\boldsymbol{b}) = -\sum_{j\in\mathcal{C}}\sum_{i}\delta_{a_i,j}.\delta_{b_i,j}.$$

2.2.6 Efficient Implementation

An efficient implementation of QS was achieved by 1) performing precomputations, 2) implementing an optimal partial sorting algorithm for selecting candidates and 3) optimal coding and compilation. These are described below.

According to Equation (2.13), $\overline{\mathcal{F}\{\mathbb{1}(T_i)\circ f_j(T_i)\}}$ is independent of the searched pattern $N(s)$. Therefore, it is possible to precompute it at the initialization stage for all i and j. This improvement typically reduces the computation time for an MPS simulation by a factor of at least 2.

In the QS algorithm, a substantial part of the computation cost is incurred in identifying the k best candidates in the mismatch map. In the case of noninteger k, the upper limit $\lceil k \rceil$ is used. Identifying the best candidates requires sorting the values of the mismatch map and retaining the candidates in the top k ranks. For this, an efficient sorting algorithm is needed. The operation of finding the k best candidates can be implemented with a partial sort, in which only the elements of interest are sorted, while the other elements remain unordered. This results in two sets: $\mathfrak{S}_s$ with the k smallest elements and $\mathfrak{S}_l$ with the largest elements. The partial sort guarantees that $x \leq y \mid (x,y) \in \mathfrak{S}_s \times \mathfrak{S}_l$. More information about our implementation of this algorithm is available in Appendix 2.8.1. Here, we use a modified vectorized online heap-based partial sort (Appendix 2.8.1). With a complexity of $O(n.\ln(k))$, it is especially suitable for small values of k. Using the cache effect, the current implementation yields results that are close to the search of the best value (the smallest value of the array). The main limitation of standard partial sort implementations is that in the case of equal values, either the first or the last element is sampled. Here, we develop an implementation that can uniformly sample a position among similar values with a single scan of the array. This is important because systematically selecting the same position for the same pattern will reduce the conditional probability density function to a unique sample, thereby biasing the simulation.

Due to the intensive memory access by repeatedly scanning large training images, interpreted programming languages, such as MATLAB and Python, are inefficient for a QS implementation and, in particular, for a parallelized implementation. We provide a NUMA-

aware and flexible C/C++/OpenMP implementation of QS that is highly optimized. Following the denomination of Mariethoz (2010), we use a path-level parallelization with a waiting strategy, which offers a good trade-off between performance and memory requirements. In addition, two node-level parallelization strategies are available: if many training images are used, a first parallelization is performed over the exploration of the training images; then, each FFT of the algorithm is parallelized using natively parallel FFT libraries.

The FFTw library (Frigo and Johnson, 2018) provides a flexible and performant architecture-independent framework for computing n-dimensional Fourier transformations. However, an additional speed gain of approximately 20% was measured by using the Intel MKL library (Intel Corporation, 2019) on compatible architectures. A GPU implementation that uses clFFT, for compatibility, is available too. Many Fourier transforms are sparse and, therefore, can easily be accelerated in n-dimensional cases with "partial FFT" since Fourier transforms of only zeros result in zeros.

2.3 Results

2.3.1 Simulation examples

This section presents illustrative examples for continuous and categorical case studies in 2D and in 3D. Additional tests are reported in Appendix 2.8.4. The parameters that are used for the simulations of Figure 2.3 are reported in Table 2.1.

The results show that simulation results are consistent with what is typically observed with state-of-the-art MPS algorithms. While simulations can accurately reproduce TI properties for relatively standard examples with repetitive structures (e.g., MV, Strebelle, and Folds), training images with long-range features (typically larger than the size of the TI) are more difficult to reproduce, such as in the Berea example. For multivariate simulations, the reproduction of the joint distribution is satisfactory, as observed in the scatterplots (Figure 2.3).

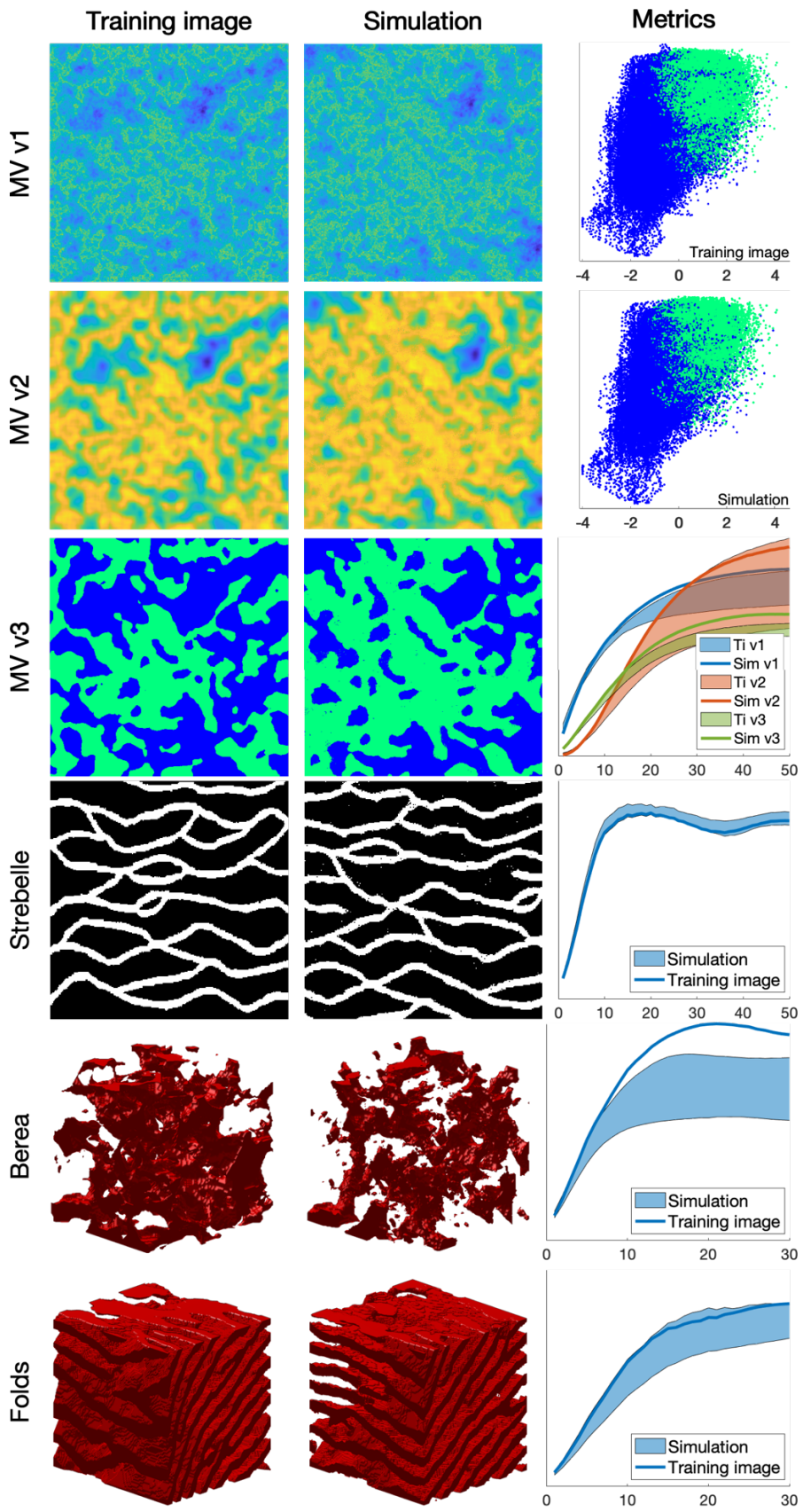

Figure 2.3 Examples of unconditional continuous and categorical simulations in 2D and 3D and their variograms. The first column shows the training images that were used, the second column one realization, and the third column quantitative quality metrics. MVs v1, v2 and v3 represent a multivariate training image (and the corresponding simulation) using 3 variables. The first two metrics are scatterplots of MV v1 vs. MV v2 of the training image and the simulation, respectively. The third metric represents the reproduction of the variogram for each of MVs v1, v2 and v3.

	MVs v1, v2, v3	Strebelle	Berea	Folds
Source	(Mariethoz and Caers, 2014)	(Strebelle, 2002)	Doi:10.6084/m9.figshare.1153794	(Mariethoz and Caers, 2014)
Size of the training image (px)	490×490	250×250	$100 \times 100 \times 100$	$180 \times 150 \times 120$
Size of the simulation (px)	490×490	250×250	$100 \times 100 \times 100$	$180 \times 150 \times 120$
Computation time (s)	1456	54	1665	76270
k	1.2			
N	80		125	

Table 2.1 Parameters that were used for the simulations in Figure 2.3. Times are specified for simulations without parallelization.

2.3.2 Comparison with direct sampling simulations

QS simulations are benchmarked against DS using the "Stone" training image (Figure 2.4). The settings that are used for DS are based on optimal parameters that were obtained via the approach of Baninajar et al. (submitted), which uses stochastic optimization to find optimal parameters. In DS, we use a fraction of scanned TI of $f = 1$ to explore the entire training image via the same approach as in QS and we use the L^2-norm as in QS. To avoid the occurrence of verbatim copy, we include 0.1% conditioning data, which are randomly sampled from a rotated version of the training image. The number of neighbors N is set to 20 for both DS and QS and the acceptance threshold of DS is set to 0.001.

The comparison is based on qualitative (Figure 2.5) and quantitative (Figure 2.6) metrics, which include directional and omnidirectional variograms, along with the connectivity function and the Euler characteristic (Renard and Allard, 2013). The results demonstrate that the simulations are of a quality that is comparable to DS. With extreme settings (highest pattern reproduction regardless of the computation time), both algorithms perform similarly, which is reasonable since both are based on sequential simulation and both directly import data from the training image.

With QS, kernel weighting enables the adaption of the parametrization to improve the results, as shown in Figure 2.5. In this paper, we use an exponential kernel:

$$\omega_{\boldsymbol{l}} = e^{-\alpha \|\boldsymbol{l}\|_2} \tag{2.16}$$

where α is a kernel parameter. The validation metrics of Figure 2.6 show that both QS and DS tend to slightly underestimate the variance and the connectivity. Figure 2.6 shows that an optimal kernel improves the results for all metrics, with all training image metrics in the 5-95% realization interval, except for the Euler characteristic.

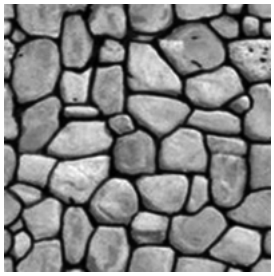

Figure 2.4 Training image that was used for benchmarking and sensitivity analysis.

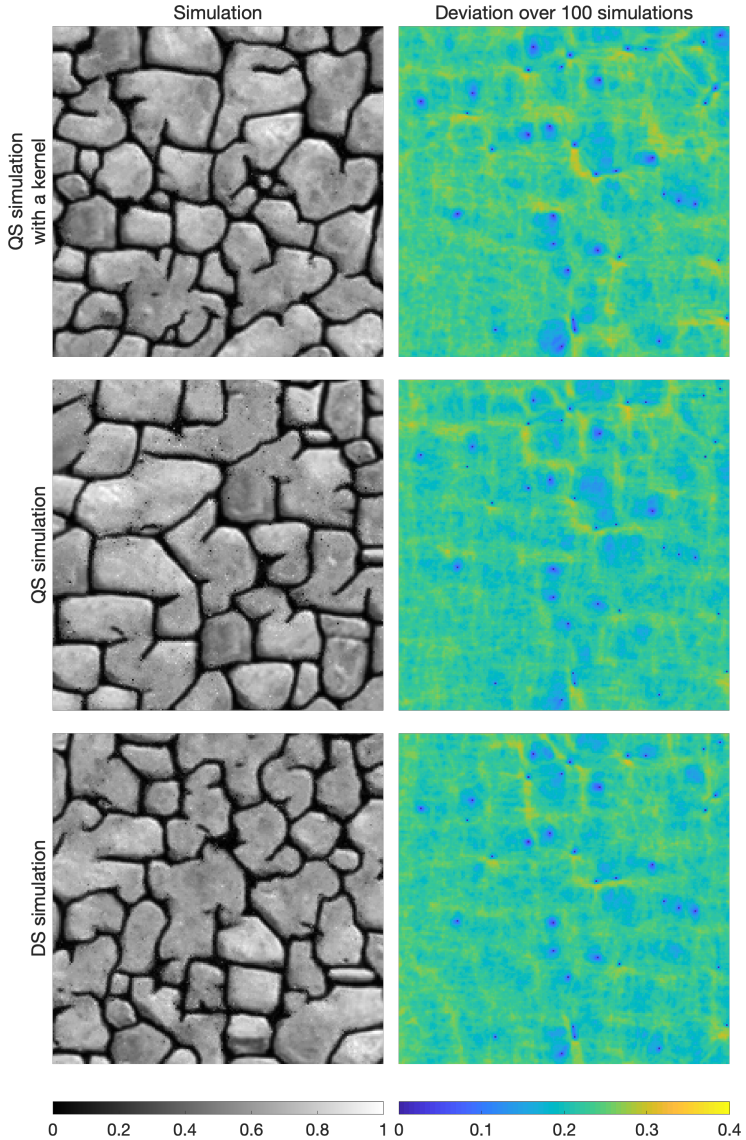

Figure 2.5 Examples of conditional simulations and their standard deviation over 100 realizations that are used in the benchmark between QS and DS.

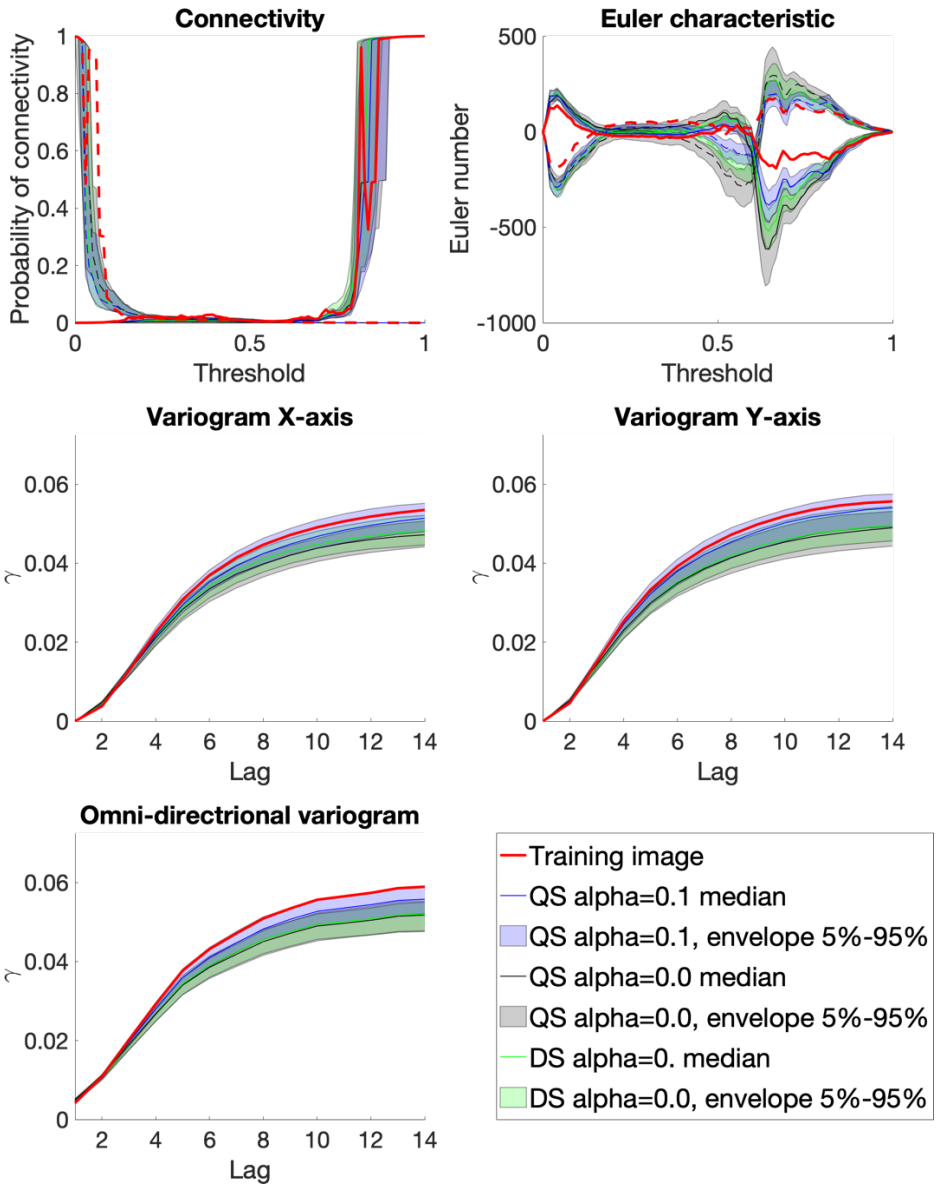

Figure 2.6 Benchmark between QS and DS using connectivity, Euler characteristic, and (omni-)directional variograms .

2.3.3 Parameter sensitivity analysis

In this section, we perform a sensitivity analysis on the parameters of QS using the training image in Figure 2.4. Only essential results are reported in this section (Figure 2.7 and Figure 2.8); more exhaustive test results are available in Appendix 2.8.4 (Figure 2.13 and Figure 2.14). The two main parameters of QS are the number of neighbors N and the number of used candidates k.

Figure 2.7 (and Appendix 2.8.4 Figure 2.13) shows that large N values and small k values improve the simulation performance; however, tend to induce verbatim copy in the simulation. Small values of N result in noise with good reproduction of the histogram.

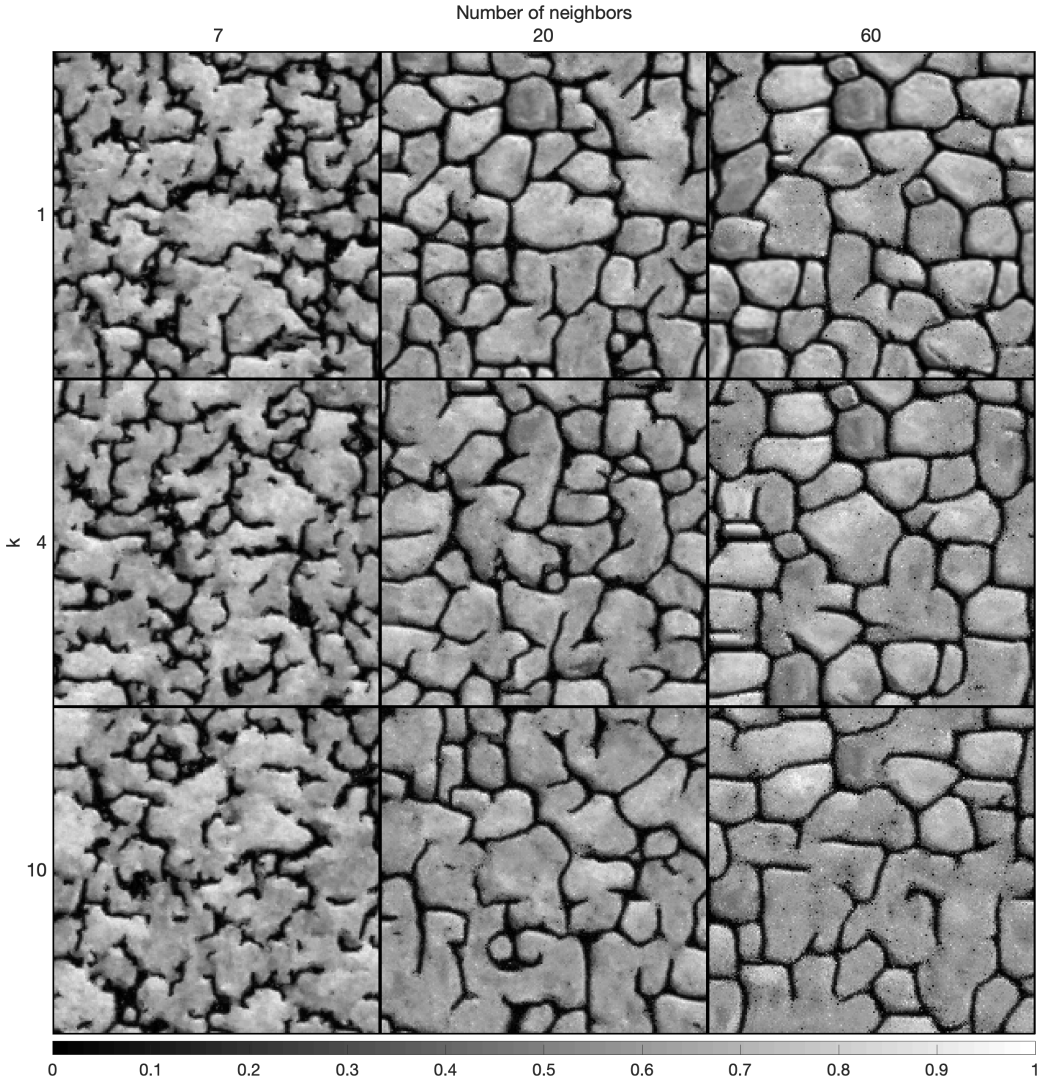

Figure 2.7 Sensitivity analysis on one simulation for the two main parameters of QS using a uniform kernel.

Weighting can be a very powerful tool, typically using the assumption that the closest pixels are more informative than remote pixels. The results of the study of the effect of the kernel value α are explored in Figure 2.8 and Figure 2.14, which shows that α provides a unique tool for improving the quality of the simulation.

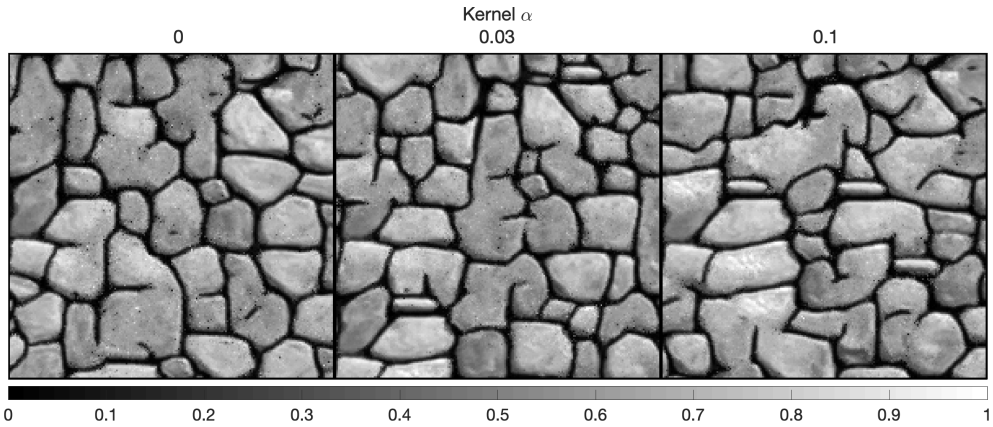

Figure 2.8 Sensitivity analysis on the kernel parameter α, with fixed parameters k=1.5 and N=40.

2.3.4 Computational efficiency and scalability

In this section, we investigate the scalability of QS with respect to the size of the simulation grid, the size of the training image grid, the number of variables, incomplete training images, and hardware. According to the test results, the code will continue to scale with new-generation hardware.

As explained in Section 2.2.3 and 2.2.4, the amounts of time that are consumed by the two main operations of QS (finding candidates and sorting them) are independent of the pixel values. Therefore, the training image that is used is not relevant (here, we use simulations that were performed with the TI of Figure 2.4 and its classified version for categorical cases). Furthermore, the computation time is independent of the parametrization (k and N). However, the performance is affected by the type of mismatch function that is used; here, we consider both continuous (Equation (2.2) and Equation (2.14)) and categorical cases (Equation (2.3) and Equation (2.15)).

We also test our implementation on different types of hardware, as summarized in Table 2.2. We expect Machine (2) to be faster than Machine (1) for medium-sized problems due to the high memory bandwidth requirement of QS. Machine (3) should also be faster than Machine (1) because it takes advantage of a longer vector computation (512-bit VS. 256-bit instruction set).

Name of the machine	Machine (1)	Machine (2)	Machine (3)
CPU	-2x Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80 GHz	-Xeon Phi, Intel(R) Xeon Phi (TM) CPU 7210 @ 1.30 GHz	-2x Intel(R) Xeon(R) Gold 6128 Processor @ 3.40 GHz
Memory type	- DDR3	- MCDRAM / DDR4	- DDR4
OS, compiler and compilation flags	Linux, Intel C/C++ compiler 2018 with -xhost		

Table 2.2 Hardware that was used in the experiments

Figure 2.9 plots the execution times on the three tested machines for continuous and categorical cases and with training images of various sizes. Since QS has a predictable execution time, the influence of the parameters on the computation time is predictable: linear with respect to the number of variables (Figure 2.9a, Figure 2.9b), linear with respect to the size (number of pixels) of the simulation grid and following a power function of the size of the

training image (Figure 2.9c). Therefore, via a few tests on a set of simulations, one can predict the computation time for any other setting.

Figure 2.9d shows the scalability of the algorithm when using the path-level parallelization. The algorithm scales well until all physical cores are being used. Machine (3) has a different scaling factor (slope). This suboptimal scaling is attributed to the limited memory bandwidth. Our implementation of QS scales well with an increasing number of threads (Figure 2.9d), with an efficiency above 80% using all possible threads. The path-level parallelization strategy that was used involves a bottleneck for large number of threads due to the need to wait for neighborhood conflicts to be resolved (Mariethoz 2010). This effect typically appears for large values of N or intense parallelization (>50 threads) on small grids. It is assumed that small grids do not require intense parallelization; hence, this problem is irrelevant in most applications.

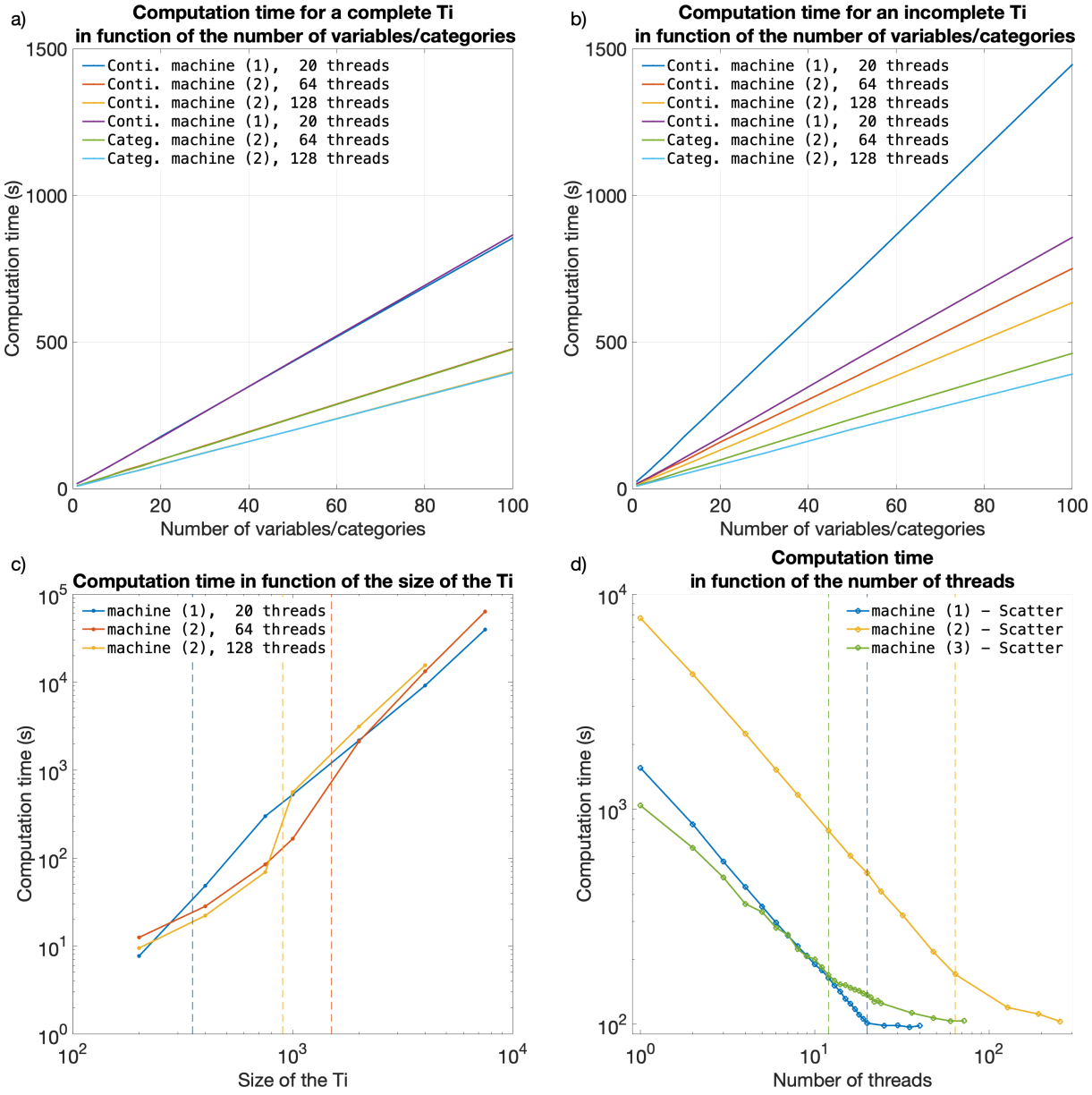

Figure 2.9 Efficiency of QS with respect to all key parameters. a) and b) are the evolutions of the computation time for complete and incomplete training images, respectively, with continuous and categorical variables. c) shows the evolution of the computation time as the size of the training image is varied; the dashed lines indicate that the training image no longer fits in the CPU cache. d) shows the evolution of the computation time as the number of threads is increased. The dashed lines indicate that all physical cores are used.

2.4 Discussion

The parameterization of the algorithm (and therefore simulation quality) has almost no impact on the computational cost, which is an advantage. Indeed, many MPS algorithms impose trade-offs between the computation time and the parameters that control the simulation quality, thereby imposing difficult choices for users. QS is comparatively simpler to set up in this regard. In practice, a satisfactory parameterization strategy is often to start with a small k value (say 1.2) and a large N value (> 50) and then gradually change these values to increase the variability if necessary (Figure 2.6 and Figure 2.13).

QS is adapted for simulating continuous variables using the L^2-norm. However, a limitation is that the L^1-norm does not have a decomposition that satisfies Equation (2.1) and, therefore, cannot be used with QS. Another limitation is that for categorical variables, each class requires a separate FFT, which incurs an additional computational cost. This renders QS less computationally efficient for categorical variables (if there are more than 2 categories) than for continuous variables. For accelerated simulation of categorical variables, a possible alternative to reduce the number of required operations is presented in Appendix 2.8.2. The strategy is to use encoded variables, which are decoded in the mismatch map. While this alternative yields significant computational gains, it does not allow the use of a kernel weighting and is prone to numerical precision issues.

The computational efficiency of QS is generally great compared to other pixel-based algorithms: for example, in our tests it performed faster than DS. QS requires more memory than DS, especially for applications with categorical variables with many classes and with a path-level parallelization. However, the memory requirement is much lower compared to MPS algorithms that are based on a pattern database, such as SNESIM.

There may be cases in which QS is slower than DS, in particular when using a large training image that is highly repetitive (e.g., chessboard). In such cases, using DS can be advantageous as it must scan only a very small part of the training image. For scenarios of this type, it is possible to adapt QS such that only a small subset of the training image is used; this approach is described in Appendix A3.

QS can be extended to handle the rotation and scaling of patterns by applying a constant rotation or affinity transformation to the searched patterns (Strebelle, 2002). However, the use of rotation-invariant distances and affinity-invariant distances (as in Mariethoz and Kelly, 2011), while possible in theory, would substantially increase the computation time. Mean-invariant distances can be implemented by simply adapting the distance formulation in QS. All these advanced features are outside the scope of this paper.

2.5 Conclusions

QS is an alternative approach for performing n-dimensional pixel-based simulations, which uses an L^2-distance for continuous cases and an L^0-distance for categorical data. The framework is highly flexible and allows other metrics to be used. The simple parameterization of QS renders it easy to use for nonexpert users. Compared to other pixel-based approaches,

QS has the advantage of generating realizations in constant and predictable time for a specified training image size. Using the quantile as a quality criterion naturally reduces the small-scale noise compared to DS. In terms of parallelization, the QS code scales well and can adapt to new architectures due to the use of external highly optimized libraries.

The QS framework provides a complete and explicit mismatch map, which can be used to formulate problem-specific rules for sampling or even solutions that take the complete conditional probability density function into account, for example, such as a narrowness criterion for the conditional pdf of the simulated value (Gravey et al., 2019), or to use the mismatch map to infer the optimal parameters of the algorithm.

2.6 Code availability

The source code and documentation of the QS simulation algorithm are available as part of the G2S package at: https://github.com/GAIA-UNIL/G2S

Platform: Linux / macOS / Windows 10 Language: C/C++

Interfacing functions in MATLAB, Python3, R

A package is available with our unbiased partial sort at: https://github.com/mgravey/randomKmin-max

2.7 Author contribution

MG proposed the idea, implemented and optimized the QS approach and wrote the manuscript. GM provided supervision, methodological insights and contributed to the writing of the manuscript.

2.8 Appendices

2.8.1 Partial sorting with random sampling

Standard partial sorting algorithms resolve tie ranks deterministically, which does not accord with the objective of stochastic simulation with QS, where variability is sought. Here, we propose an online heap-based partial sort. It is realized with a single scan of the array of data using a heap to store previously found values. This approach is especially suitable when we are interested in a small fraction of the entire array.

Random positions of the k best values are ensured by swapping similar values. If $k = 1$, the saved value is switched with a smaller value each time it is encountered. If an equal

value is scanned, a counter c is increased for this specific value and a probability of $1/c$ of switching to the new position is applied. If $k > 1$, the same strategy is extended by carrying over the counter c.

This partial sort outperforms random exploration of the mismatch map. However, it is difficult to implement efficiently on GPUs. A solution is still possible for shared-memory GPUs by performing the partial sort on the CPU. This is currently available in the proposed implementation.

```
k: the number of values of interest
D: the input data array
S: the array with the k smallest values (sorted)
Sp: the array with the positions that are associated with the values of S

for each value v of D
    if v is smaller than the smallest value of S
        search in S for the position p at which to insert v and insert it
        if p = k                        // last position of the array
            reinitialize the counter c to 0
            insert v at the last position
        else
            increment c by one
            swap the last position with another of the same value
            insert the value at the expected position p
        end
    else if v is equal to the smallest value of S
        increment c by one
        change the position of v to one of the n positions of equal value with a
    probability of n/(n + c)
    end
end
end
```

2.8.2 Encoded categorical variables

To handle categorical variables, a standard approach is to consider each category as an independent variable. This requires as many FFTs as classes. This solution renders it expensive to use QS in cases with multiple categories.

An alternative approach is to encode the categories and to decode the mismatch from the cross-correlation. It has the advantage of only requiring only a single cross-correlation for each simulated pattern.

Here, we propose encoding the categories as powers of the number of neighbors, such that their product is equal to one if the class matches. In all other cases, the value is smaller than one or larger than the number of neighbors.

$$\varepsilon_{L^0}(a,b) = \psi\big((a-b)^0 \propto -(N+1)^{-p(a)}.(N+1)^{-p(b)}\big)$$

where N is the largest number of neighbors that can be considered and $p(c)$ is an arbitrary function that maps index classes of $\mathcal{C}$, $c \in \mathcal{C}$.

In this scenario, in Equation (2.1) this encoded distance L_e^0 can be decomposed into the following series of functions f_j and g_j :

$f_0 : x \to -(N+1)^{p(x)}$

$g_0 : x \to (N+1)^{-p(x)}$

and the decoding function is

$$\psi(x) = \lfloor x \rceil \bmod N$$

Table 2.3 describes this process for 3 classes, namely, $a, b,$ and c, and a maximum of 9 neighbors. Then, the error can be easily decoded by removing decimals and dozens.

Products	$g_0(a) = 1$	$g_0(b) = 0.1$	$g_0(c) = 0.01$
$f_0(a) = 1$	1	0.1	0.01
$f_0(b) = 10$	10	1	0.1
$f_0(c) = 100$	100	10	1

Table 2.3 Example of encoding for 3 classes and 9 neighbors and their associated products

Consider the following combination:

$$f_0(a,\ b,\ a,\ c,\ c,\ b,\ a,\ a,\ b)$$
$$\times\, g_0(c,\ b,\ b,\ a,\ a,\ b,\ c,\ a,\ a)$$
$$-(0.01,\ 1,\ 0.1, 100, 100,\ 1, 0.01,\ 1,\ 10) = -213.12$$

The decoding $\lfloor -213.12 \rceil \bmod 10 = -213 \bmod 10 = -3$ yields 3 matches (in green).

This encoding strategy provides the possibility of drastically reducing the number of FFT computations. However, the decoding phase is not always implementable if a nonuniform

matrix ω is used. Finally, the test results show that the method suffers quickly from numerical precision issues, especially with many classes.

2.8.3 Sampling strategy using training image splitting

The principle of considering a fixed number of candidates can be extended by instead of taking the k^{th} best candidate, sampling the best candidate in only a portion $\frac{1}{k}$, of the TI. For instance, as an alternative to considering $k = 4$, this strategy searches for the best candidate in one fourth of the image. This is more computationally efficient. However, if all the considered candidates are contiguous (by splitting the TI in k chunks), this approximation is only valid if the TI is completely stationary and all k equal subdivisions of the TI are statistically identical. In practice, real-world continuous variables are often nonstationary. However, in categorical cases, especially in binary ones, the number of pattern replicates is higher and this sampling strategy could be interesting.

The results of applying this strategy are presented in Table 2.4 and Figure 2.10. The experimental results demonstrate that the partial exploration approach that is provided by splitting substantially accelerates the processing time. However, Figure 2.10 shows that the approach has clear limitations when dealing with training images with complex and nonrepetitive patterns. The absence of local verbatim copy can explain the poor-quality simulation results.

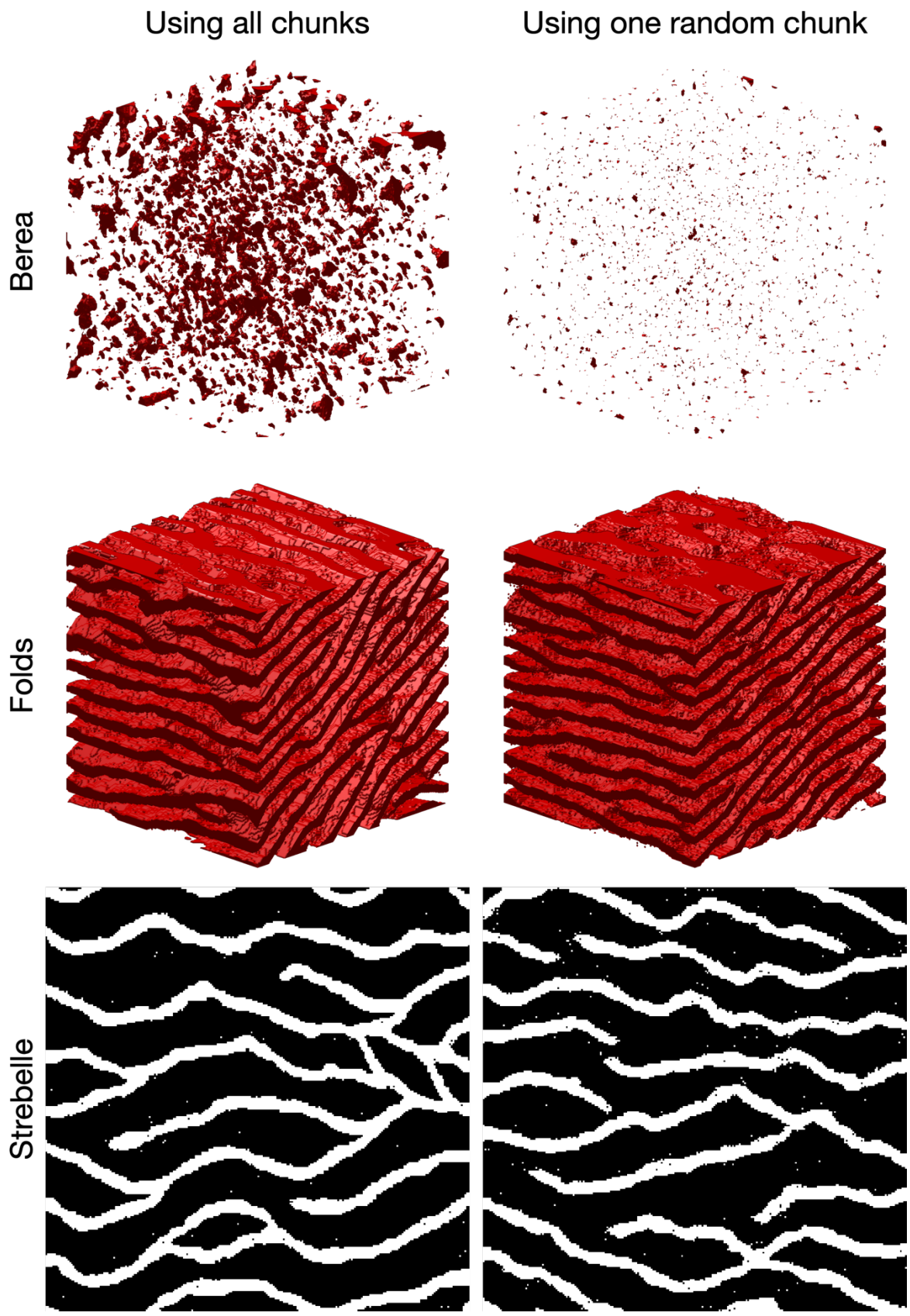

Figure 2.10 Comparison of QS using the entire training image and using training image splitting. In these examples, the training image is split into two images over each dimension. The original training images are presented in Figure 2.3.

Training image	Using all chunks	Using one random chunk	Speedup
Berea	11 052 s	1 452 s	7.61x
Folds	35 211 s	4 063 s	8.66x
Strebelle	7.95 s	3.16 s	2.51x

Table 2.4 Computation times and speedups for the full and partial exploration approaches. Times are specified for simulations with path level parallelization.

2.8.4 Additional results

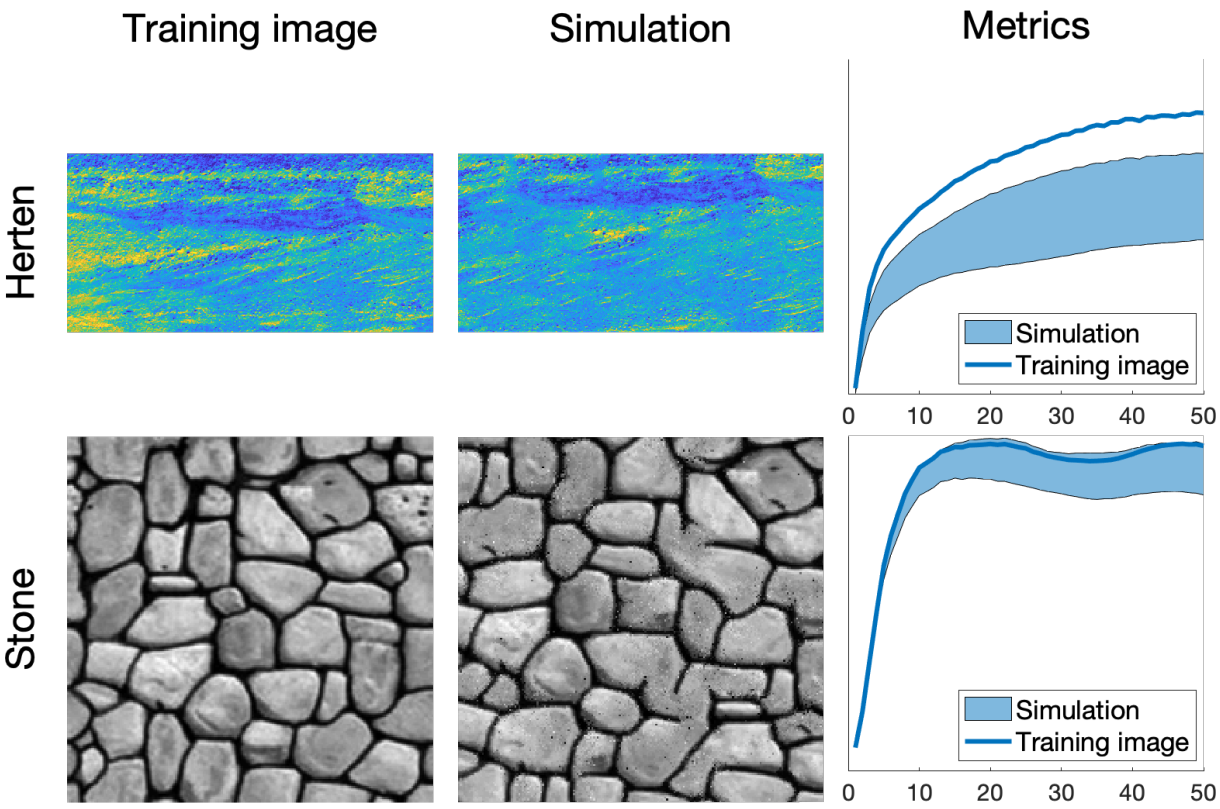

Figure 2.11 Examples of 2D simulations: the first 3 rows represent 3 variables of a single simulation.

	Herten	Stone
Source	(Mariethoz and Caers, 2014)	(Mariethoz and Caers, 2014)
Size of the training image (px)	716×350	200×200
Size of the Simulation (px)	716×350	200×200
Computation time (s)	1133	21
k	1.2	
N	80	

Table 2.5 Simulation parameters for Figure 2.11 Times are specified for simulations without parallelization.

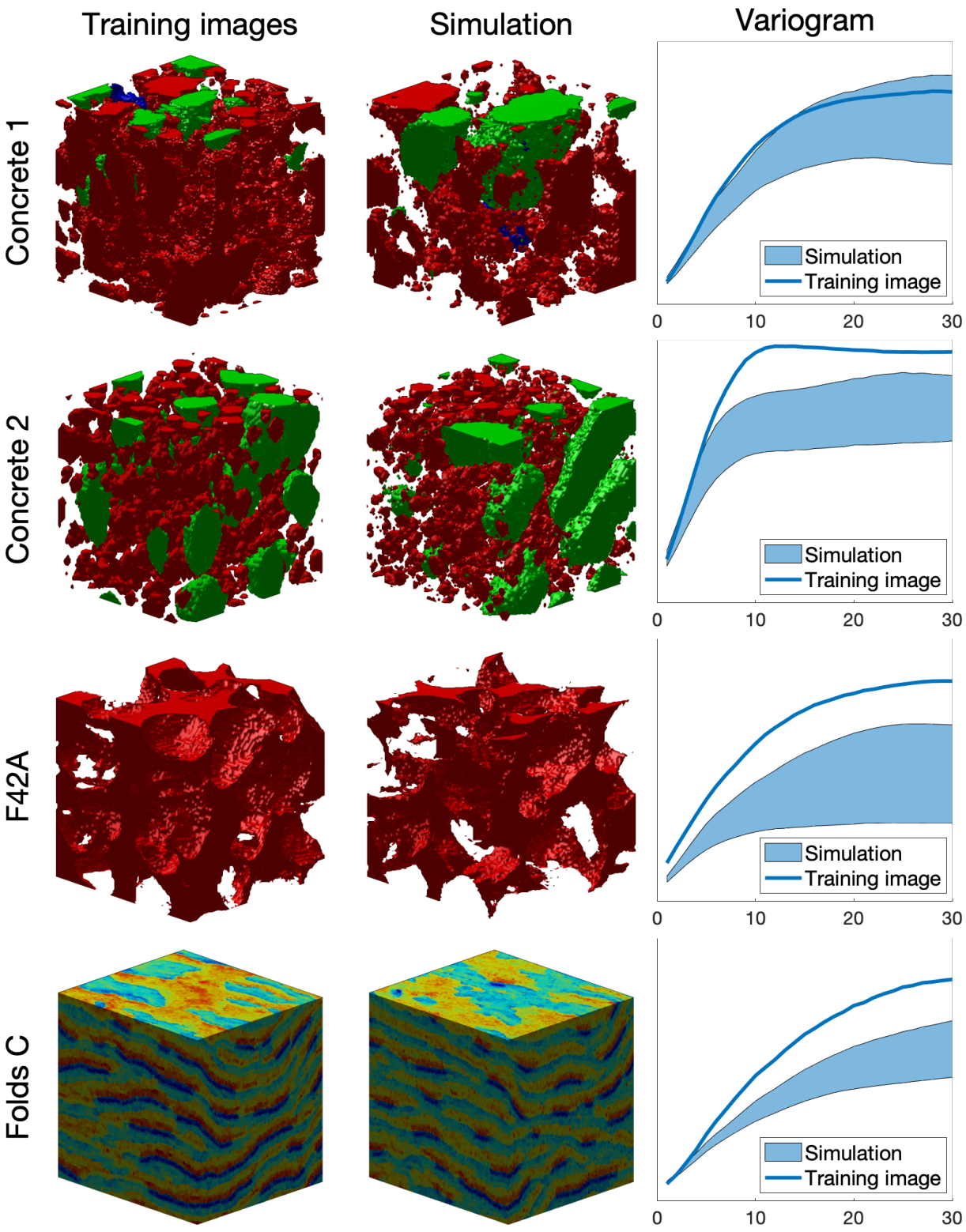

Figure 2.12 Examples of 3D simulation results.

	Concrete 1	Concrete 2	F42A	Folds continues
Source	(Meerschman et al., 2013)	(Meerschman et al., 2013)	Doi:10.6084/m9.figshare.1189259	(Mariethoz and Caers, 2014)
Size of the training image (px)	$150 \times 150 \times 150$	$100 \times 90 \times 80$	$100 \times 100 \times 100$	$180 \times 150 \times 120$
Size of the simulation (px)	$100 \times 100 \times 100$	$100 \times 100 \times 100$	$100 \times 100 \times 100$	$180 \times 150 \times 120$
Computation time (s)	11436	1416	1638	7637
k	1.2			
N	50		125	

Table 2.6 Simulation parameters for Figure 2.12. Times are specified for simulations without parallelization.

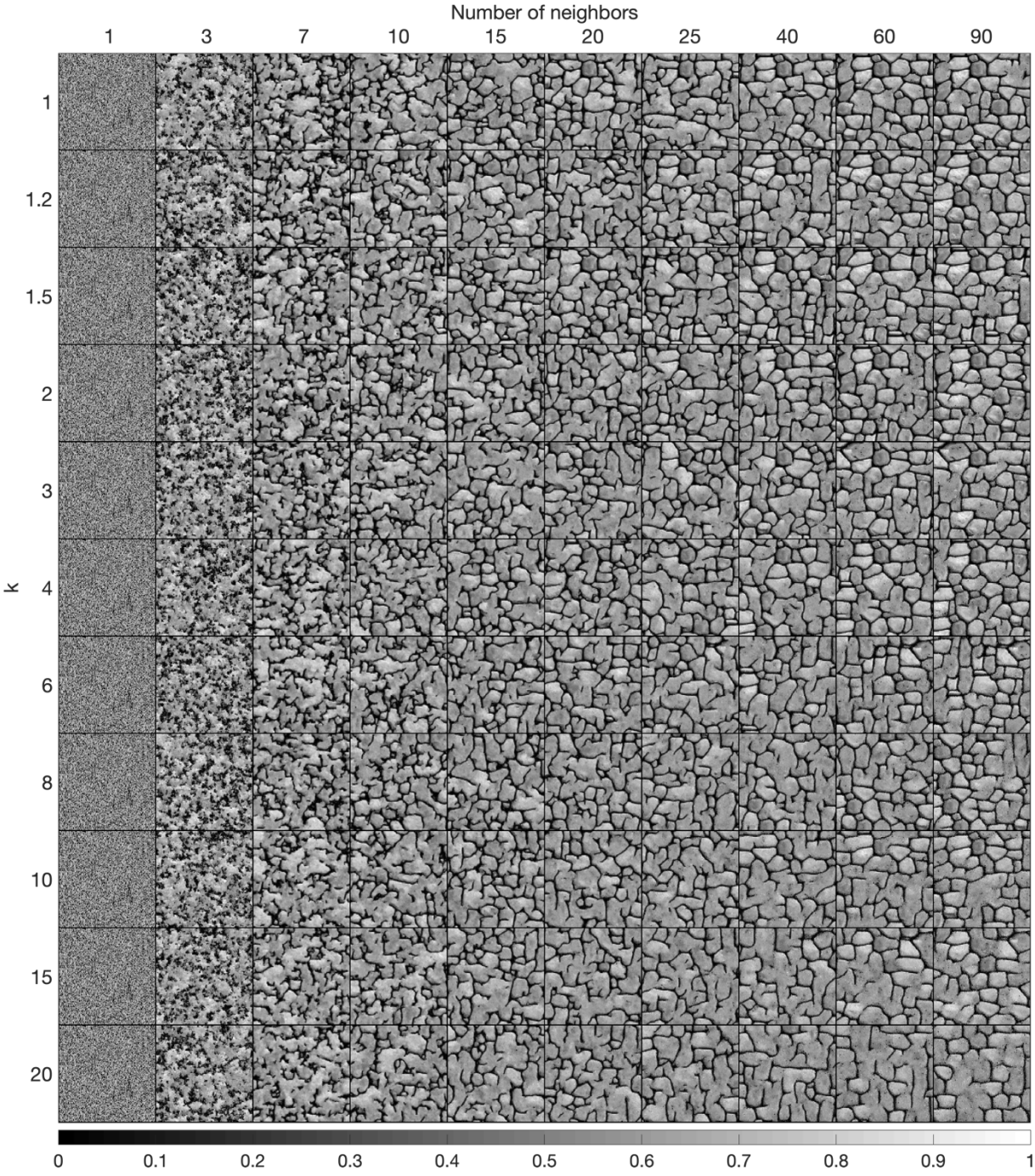

Figure 2.13 Complete sensitivity analysis, with one simulation for the two main parameters of QS.

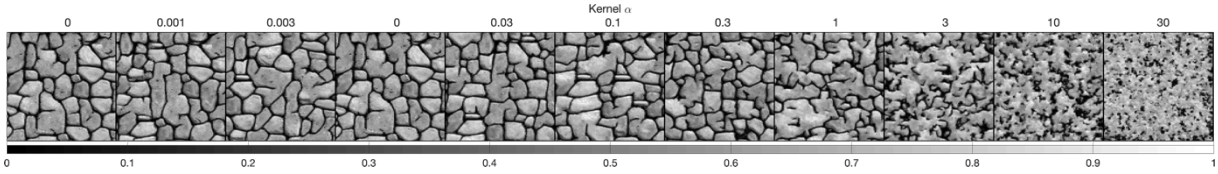

Figure 2.14 Complete sensitivity analysis, with one simulation for each kernel with k=1.5 and N=40.

2.9 Acknowledgments

This research was funded by the Swiss National Science Foundation, grant number 200021_162882. Thanks to Intel for allowing us to conduct numerical experiments on their

latest hardware using the AI DevCloud. Thanks to Luiz Gustavo Rasera for his comments, which greatly improved the manuscript; to Ehsan Baninajar for running his optimization method, which improved the reliability of the benchmarks; and to all the early users of QS for their useful feedback and their patience in waiting for this manuscript.

3 Analogue-based colorization of remote sensing images using textural information

Mathieu Gravey[a], Luiz Gustavo Rasera[a], Gregoire Mariethoz [a]
[a] Univerity of Lausanne, Faculty of Geosciences and Environment, Institute of Earth Surface Dynamics

Highlights

- A set of algorithms to spectrally enhance historical satellite images is proposed for the first time
- The spatial patterns are used to retrieve the spectral information
- The colorization is based on a given training image
- Comparison of algorithms shows that significant improvement of images is possible in most situations

Abstract

Satellite images are richer than ever before. For example, new Landsat-8 images with their 11 bands carry much more information than older generations of satellites. These differences in spectral representation imply a major difficulty for assessing long-term land surface changes. The easiest solution is to reduce the information of the most recent product, for example by only keeping a subset of the Landsat-8 bands that matches old imagery. To avoid such loss of information, we propose a new method based on multiband spatial pattern matching. We are focusing on increasing the spectral resolution of archive satellite images to the same level of spectral resolution and coverage as modern imagery. Our method uses analogous scenes taken from modern satellites, which have conceptually the same role as the training images used in multiple-point geostatistics simulation. The spectral characteristics of the training image are then transferred to a target archive image, where new synthetic spectral bands are generated. A spatial pattern matching procedure is used to control this transfer, resulting in preservation of spatial and spectral coherence in the results. We illustrate the methodology on Landsat 8 and Corona imagery. The proposed method was benchmarked against other state-of-the-art colorization techniques, and it shows globally better results.

3.1 Introduction

Multispectral images play a crucial role in a variety of scientific fields, ranging from geoscience to urban planning and ecology. Their spectral information enables using classification (Bruzzone and Demir, 2014) or object detection (Cheng and Han, 2016) to differentiate Earth surface properties such as land cover types (Vaduva et al., 2013) and, crucially, allows monitoring changes in those properties.

In the last decade, archive images have become available through major changes in data policies, such as the release of formerly classified imagery (USGS, 2008) and the waiving of restrictions for the civil use of very fine-resolution imagery. In addition, access to archive images has dramatically improved thanks to free online portals like those of the USGS, the ESA, and platforms such as Google and Amazon. As a result, multi-temporal information is now widely used for remote sensing change detection. For example, such data make it possible to measure urban growth (Weng, 2001), to determine the human impact on rainforests (Hansen et al., 2008), and the evolution of water bodies (Pekel et al., 2016). For such applications, historic remote sensing data are invaluable because they allow quantifying long-term changes induced by climate change or by direct human activity (Lu et al., 2004).

Despite the above advances in the possibilities offered by Earth observation, a significant remaining difficulty for long-term change detection is the inconsistency between different satellite sensors, which are due to the rapid evolution of sensors and the limited lifetime of each platform. The oldest continuously available satellite sensor data set dates back to only about 50 years, with Landsat 1 launched in 1972 that had only 4 spectral bands (Zanter, 2005). Today, Landsat 8 has 12 bands (Zanter, 2005), and other platforms propose up to 220 bands such as the Hyperion sensor on EO-1 (Beck, 2010). Pre-Landsat imagery is available, albeit with limited and infrequent coverage and often restricted to panchromatic information, starting from 1960 with the Corona spy imaging program (USGS, 2008). This great variability in the information content of multi-temporal images makes their analysis difficult (Yang et al., 2013). The changes in characteristics between the different generations of products particularly affect studies related to change detection, which cannot use the full potential of satellite archives. As a result, in most cases satellite-based long-term change detection studies use post-1984 data, because 1984 is the launch year of Landsat-4 and its modern Thematic Mapper. Among many examples, a few specific papers that face this

problem are for the study of long-term changes in Arctic vegetation (Fraser et al., 2011), for the detection of changes in Mediterranean urban and vegetation cover (Alphan and Yilmaz, 2005), for wetland monitoring in Zambia (Munyati, 2010) or for assessment of the tourism boom in Egypt (Dewidar, 2002). The temporal range of such studies would be greatly increased if they could include images dating from as early as 1972.

This paper focuses on the enrichment of old remote sensing images such that they become comparable with present-day acquisitions. While old and recent images may differ in terms of spatial, temporal and spectral resolutions, we only focus here on the spectral aspect. In order to compare time-series images that have different spectral bands, or different band definitions, one needs to find a common reference frame where both old and new images can be represented and compared. An usual and straightforward approach is to decimate the data to a common denominator by removing the specific spectral bands that are missing in the oldest part of the collection, or to compute indices that are easier to compare (Cohen et al., 2002). Nevertheless, this approach is suboptimal because the information from the most recent sensor is not used. Here, we investigate the possibility of enhancing the spectral characteristics of the old images such that they become comparable with more recent ones for studies of land surface changes.

The premise of the approach proposed in this paper is that spectral enhancement of remote sensing data can be treated as an image colorization problem. Given an older image with a limited number of spectral bands (the target image), we assume that we have another, more recent, and spectrally richer image of similar surface characteristics (the training image). Patterns are defined that characterize the spectral signature of pixels, as well as in their spatial neighborhood (we call those spatio-spectral patterns). Then, based on the training image, we identify correspondences by matching the spatio-spectral patterns, and these correspondences are used to import spectral signatures from the training image to the target image (Figure 3.1).

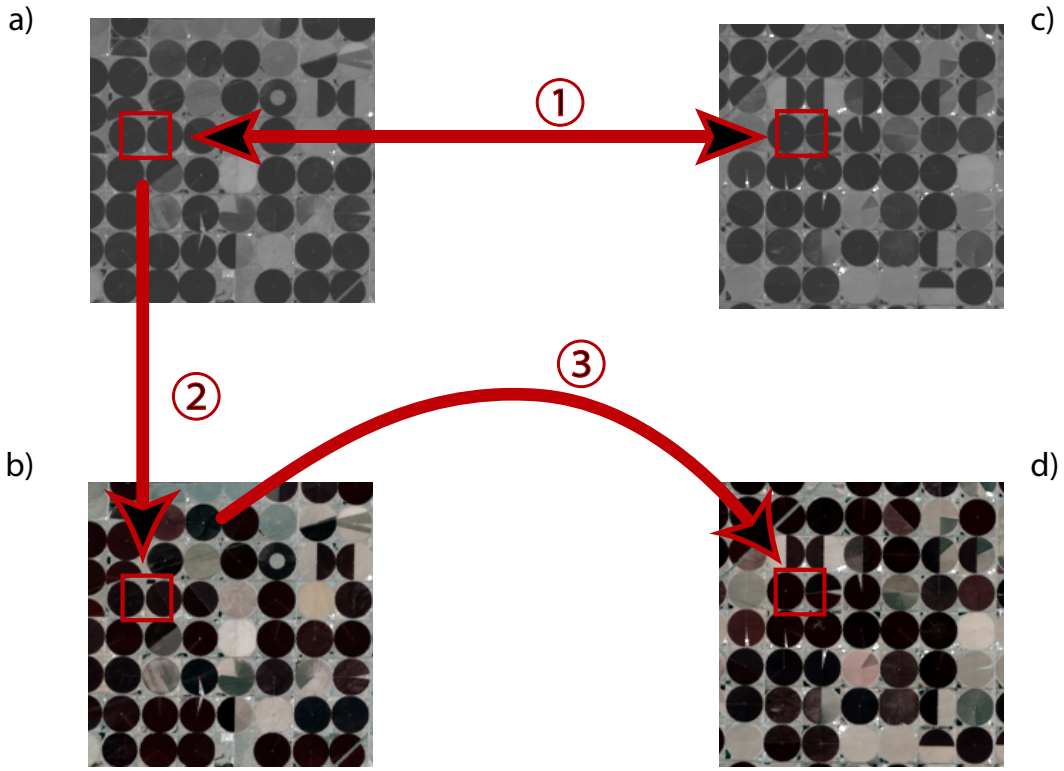

Figure 3.1. The principle of image colorization based on a training image. a) grayscale training image, b) color training image associated to a), c) target grayscale image to colorize, d) colorized output of c) using a) and b). ① spatial pattern matching, ② color identification and ③ color import.

The training image can be an analogue image with similar surface features to the target domain, or a recent image of the area of interest that contains all bands information.

When spectrally enhancing satellite sensor images, two possible situations can be encountered, as illustrated in Figure 3.2:

- Spectral disaggregation: when the information present in the old image integrates all bands of interest, but in a convoluted representation. This is, for example, the case when the panchromatic band is available in an old image and has to be disaggregated to the three RGB color bands. In this example, the panchromatic band is defined as the integral of all three-color wavebands.
- Spectral extrapolation: when the target band is not contained at all in the old image. A good example of this situation is when the panchromatic information is available, and the near-infrared (NIR) is the target band to recover. The NIR band corresponds to wavelengths that are not contained in the panchromatic band. However, extrapolating the NIR information could potentially have applications, such as computing the normalized difference vegetation index (NDVI). Logically, spectral extrapolation tends to be more challenging than spectral disaggregation.

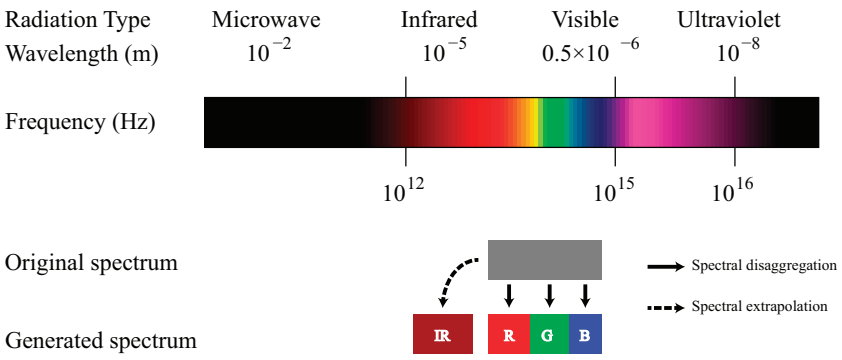

Figure 3.2: The difference between spectral disaggregation and spectral extrapolation.

In both cases, a naive approach would consist of estimating the spectral response in an unknown band based on the values of the known bands, using an assumption of continuity in the spectral signature of the observed surface. However, this is often not appropriate because the spectral response of a given object can be completely different in each spectral band. Therefore, one cannot simply interpolate band values across the electromagnetic spectrum. The approach we propose in this paper for spectral disaggregation does not make assumptions on the continuity of the spectral signatures, and instead relies on analogues to extend the spectral signatures of historical remote sensing images. This paper focuses on both issues of spectral disaggregation and extrapolation. It studies, for the first time, the possibility of spectral enhancement of remote sensing images based on training examples. Several algorithms coming from both computer graphics and multiple-point statistics are tested in the context of spectral enhancement, and a new approach that outperforms these classical algorithms is proposed.

This remaining of the paper is structured as follows. We start reviewing related problems in other fields and the different solutions that have been proposed. We then propose our algorithm and compare the results with other approaches. Finally, after discussing the results, we present guidelines to obtain optimal results with minimal computational cost.

3.2 Training-based colorization algorithms

In this section, we first explore the application of the work done by Welsh (2002), which proposes a training-based colorization approach of grayscale images by using pattern-matching applied independently to each pixel. As will be discussed later, this method has limitations due to the absence of spatial dependence between the simulated data. We then investigate the Direct Sampling multiple-point statistics method (Mariethoz et al., 2010),

which has the particularity of taking into account the spatial dependence between simulated values. Finally, we propose an improvement on the aforementioned methods with the use of a guided simulation path that allows reducing the prediction uncertainty and improving the overall results. The guided simulation path is a variant of the strategy originally proposed by Liu and Journel (2004) in the context of geostatistical simulation of categorical variables.

3.2.1 General pattern matching strategy

All algorithms investigated in this paper use a look-up approach in a training image as a basis to generate new data. Given a target pattern to colorize, they identify another pattern in the training image from which to import the missing spectral information. In all cases, this operation is done by using spatial pattern matching, possibly in a large database of multi-temporal training images. In consequence, the algorithms described in the coming sections will largely use the concept of pattern matching, which require a few preliminary definitions and notations:

$\boldsymbol{u}$ denotes a position in the training image (the source image with complete multispectral information), and $\boldsymbol{v}$ denotes a position in the target image (the old image with only a limited number of bands being informed). B is the set of all spectral bands considered. $\mathcal{P} \in \mathbb{R}^2$ represents the extent of a pattern, which is defined by a series of lag vectors $\boldsymbol{h} \in \mathcal{P}$ that describe the relative position between the central position of the pattern ($\boldsymbol{u}$ in the training image or $\boldsymbol{v}$ in the target image) and each position in the pattern. $w_i(\boldsymbol{h})$ is a kernel function that weights the importance of a matching error for a given lag $\boldsymbol{h} \in \mathcal{P}$ and band $i \in B$.

Each value in an image is defined by a spatial position and a spectral band. We denote $B_i(\boldsymbol{x})$ the value from the band $i \in B$ at the position $\boldsymbol{x}$. By extension, $\boldsymbol{B}(\boldsymbol{x})$ is the vector containing all spectral values at position $\boldsymbol{x}$. A neighborhood $N_i(\boldsymbol{x})$ around $\boldsymbol{x}$ is defined as the association of the lag vectors and the band values from $\mathcal{P}$ where a value exists. It is denoted:

$$N_i(\boldsymbol{x}) = \{\boldsymbol{h} \in \mathcal{P} | B_i(\boldsymbol{x} + \boldsymbol{h}) \; is \; known \}$$

By extension $N(\boldsymbol{x})$ is the set of $N_i(\boldsymbol{x})$ for all possible bands i. As $N_i(\boldsymbol{x})$ contains both spectral and spatial information, we call it a spatio-spectral pattern, and it is the entity used for defining matches between training and target images.

Typically, pattern matching involves computing a mismatch between a pattern found in the target image and another pattern coming from the training image. The mismatch

between two patterns can be measured in a variety of ways (e.g. structure of the pattern, position of the information). In this paper we use a combination of pixel-wise errors, which we denote $\varepsilon(\boldsymbol{u}, \boldsymbol{v})$. It is computed as follows:

$$\varepsilon(\boldsymbol{u}, \boldsymbol{v}) = \sqrt[\beta]{\sum_{i \in \mathrm{B}} \sum_{h \in N_i(\boldsymbol{u}) \cap N_i(\boldsymbol{v})} w_i(\boldsymbol{h}) |B_i(\boldsymbol{u} + \boldsymbol{h}) - B_i(\boldsymbol{v} + \boldsymbol{h})|^{\beta}} \quad (3.1)$$

Many possible functions can be used for the kernel *w* (Silverman, 1986). In this paper we use an exponential function $w_i(\boldsymbol{h}) = \mathrm{e}^{-\alpha_i \|h\|_2}$ where α_i controls the shape of the kernel and is specific to each band i. $\alpha \longrightarrow 0$ results in uniform weights for all neighbors. $\alpha \longrightarrow \infty$ only considers the central pixel, meaning that the spatial dependence between neighboring pixels is ignored. β is typically 1 or 2.

3.2.2 The computer graphics approach

For over a decade, the computer graphics community has faced the need to colorize grayscale images or movies, and several approaches and algorithms were developed to this end. One of the simplest methods consists in assigning a single color for each grayscale value using a mapping table (Pratt, 2007). Charpiat et al. (2009) determines the color probability at each location using previously learned gradients on a training image and then use a graph cut approach to generate a colorization having the highest global coherence. Another approach uses deep convolutional neural networks with a training phase from a large image bank, and then predicts the color channel by channel (Hwang and Zhou, 2016). Semary et al. (2007) use segmentation techniques to separate images in homogeneous areas, which are then colorized independently. Levin et al.(2004) propose to use seed points where the color is set manually by the user. The color is then diffused based on the grayscale gradient, which has the effect of propagating it along smooth areas. Noda et al. (2006) propose to formulate the problem as a Bayesian inference, and perform the colorization using maximum a posteriori estimation. So far, these methods have only been used for the colorization of grayscale images to visible color (RGB), and mostly applied to photography. Furthermore, many of these approaches work only for a given color space, and may fail when dealing with spectral bands other than RGB.

Of particular interest, Welsh (2002) proposed an algorithm that uses a training image where the color and the grayscale information are known. For each pixel $\boldsymbol{v}$ of the target, a neighborhood $N_i(\boldsymbol{v})$ is considered in a window of 5 by 5 pixels. The algorithm then searches for a similar location in the training image by pattern matching (based on computing $\varepsilon(\boldsymbol{u}, \boldsymbol{v})$

for 200 randomly sampled locations $\boldsymbol{u}$). The color at the location with lowest mismatch is then transferred to the target image.

This approach is interesting for processing old satellite products in at least two aspects:

1) It uses a training image deemed similar to the target image, which can be used to retrieve the missing multispectral information.
2) It exploits the spatial information contained in the grayscale spatial patterns to inform the missing spectral bands. The underlying assumption is that of a correspondence between a given spatial signature and a given color (or multispectral signature).

While Welsh (2002) did tests on satellite sensor data, these were very limited and restricted to the colorization from grayscale to color images (i.e. not considering multispectral images). In this paper, we use this approach as a benchmark against other algorithms. The pseudocode of the algorithm of Welsh (2002) is given in Algorithm 1.

Pseudocode of Algorithm 1 (modified implementation of Welsh (2002))

Inputs:
training image
target image
t the threshold
f the exploration ratio

1 **For** each pixel $B(\boldsymbol{v})$ of the target image, **do** independently
2 Extract the neighborhood $N(\boldsymbol{v})$ of the target image around $B(\boldsymbol{v})$
3 Find a candidate $B(\boldsymbol{u})$ in the training image that satisfies $\varepsilon(\boldsymbol{u}, \boldsymbol{v}) < t$ by scanning a fraction f of the training image
4 Import the color from the candidate $B(\boldsymbol{u})$ at location $B(\boldsymbol{v})$ in the target image.
5 End

Note that this algorithm is extremely easy to parallelize since each pixel is simulated independently.

For computational reasons, the original algorithm of Welsh (2002) searches for a matching pattern over a limited subset of the training image. Here, we modified it by using a strategy that keeps searching until a candidate with a mismatch under a given threshold is found. Therefore, if no candidate is found with an error below *t*, best candidate is accepted.

Once a candidate is found, the values of the additional bands are imported from the training image to the target image.

3.2.3 The geostatistical approach

In the field of geostatistics, the multiple point simulation approach (MPS) uses the concept of analogues, which is similar as the training image used by Welsh (2002). The goal of MPS is to generate random fields of unobserved variables, based on a training image, which represent a similar process as the one to model. In its most general application, MPS produces images by resampling patterns found in the training image and assembling them in a coherent way such as to obtain a new image that is either entirely random, or conditioned to a small number of known measurements (Guardiano and Srivastava, 1993).

There have been applications of MPS in different fields, including weather generation (Jha et al., 2015; Oriani et al., 2017), subsurface modeling (Kessler et al., 2012; Pirot et al., 2014; Strebelle, 2002), and notably in remote sensing with downscaling (Boucher, 2009; Tang et al., 2015), classification (Ge, 2013; Ge and Bai, 2011) or gap-filling of Landsat-7 images (Yin et al., 2017). Recent developments have extended the approach to condition the produced simulations to one or several exhaustively known covariates, thus extending its use for new applications such as geophysics, where indirect information is commonly available throughout the studied domain (Lochbühler et al., 2013). Over the last few years, several methods have been developed, aimed at simulating both categorical and continuous variables. Among the approaches suited for the simulation of continuous variables, Direct Sampling (DS) (Mariethoz et al., 2010) is interesting in the context of multispectral images because it allows multivariate simulation, that is, using training images consisting of multiple co-registered variables. This functionality can potentially be used for colorizing grayscale or multispectral images by considering each spectral band as a covariate.

The major difference between DS and the approach of Welsh is its sequential character. The DS algorithm first generates a path $\boldsymbol{P}$ defining the order in which pixels will be simulated sequentially. Then, the value of each pixel is simulated conditionally to the previous ones based on a neighborhood $N(x)$. This allows considering the previously simulated values when determining the value of an uninformed pixel. This link between simulated values increases their spatial coherence and improves the simulation output. Here, we use a random path $\boldsymbol{P}$, which is common in geostatistics.

Pseudocode of Algorithm 2 (Direct Sampling)

Inputs:
training image
target image
t the threshold
f the exploration ratio
$\boldsymbol{P}$ the path

1 **For** each pixel $B(\boldsymbol{v})$ of the target image **do** following sequentially the path $\boldsymbol{P}$
2 Extract the neighborhood $N(\boldsymbol{v})$ of the target image around $B(\boldsymbol{v})$ taking into account previously simulated values
3 Find a candidate $B(\boldsymbol{u})$ in the training image that satisfies $\varepsilon(\boldsymbol{u}, \boldsymbol{v}) < t$ by scanning a fraction f of the training image
4 Import the color from the candidate $B(\boldsymbol{u})$ at location $B(\boldsymbol{v})$ in the target image.
5 End
6 **Return** target image

Note that the threshold t is the same as in Algorithm 1.

3.2.4 Simulation by Narrow Distribution Selection (NDS)

We develop here the Narrow Distribution Selection (NDS) algorithm, which is partly related to an idea proposed by Liu and Journel (2004) for the simulation of categorical variables using multi-grids. While many geostatistical methods work by computing a conditional probability density function (CPDF) for a simulated value $B(\boldsymbol{v})$, Algorithms 1 and 2 do not require this CPDF. Instead, these algorithms sample only a single value for each colorized pixel. The absence of a CPDF is computationally advantageous; however, a CPDF can offer additional possibilities, which we exploit in the NDS approach. In NDS, the selection of a pattern is not only conditioned to a match with its neighborhood: an additional criterion is included, which is that the CPDF of the value for a target pixel must be narrow. Indeed, using only a best pattern match criterion (as in Algorithms 1 and 2) can yield a set of candidates with different pixel values, all with a low mismatch. Such poorly determined situations result in pixels being assigned with an irrelevant value, creating artifacts that are thereafter propagated throughout the sequential simulation. The addition of a narrowness criterion results in only simulating the pixels whose values can be determined unambiguously. Therefore, like DS, this algorithm is sequential, however the simulation path $\boldsymbol{P}$ is not random: it follows an order defined by the narrowness criterion.

Pseudocode of Algorithm 3 NDS

Inputs:
training image
target image
best pattern match criterion
narrowness criterion

Until all positions are simulated **do**
    For all non-simulated pixels $B(\boldsymbol{v})$ of the target image **do**
        Extract the neighborhood $N(\boldsymbol{v})$ of the target image around $B(\boldsymbol{v})$ taking into account previously simulated values
        Find all candidates in the training image satisfying the best pattern match criterion
        Compute the CPDF of $B(\boldsymbol{v})|N(\boldsymbol{v})$
        If the CPDF satisfies the narrowness criterion **do**
            Import the color from a random candidate $B(\boldsymbol{u})$ sampled from the CPDF, at location $B(\boldsymbol{v})$ in the target image.
        Endif
    End
End

Note that if many bands are simulated at the same time, there is one narrowness per band. In that case, all narrowness are aggregated into a single value by using a weighted average, the weights corresponding to the central value of the kernel for each band.

This approach avoids the propagation of prediction errors caused by the simulation of uncertain values. Instead, it only propagates the information contained in values with a narrow distribution (hence highly certain). A benefit of the narrowness criterion is that it is based on multiple possible candidates, which offers more information than Algorithms 1 and 2 that only retain a single candidate.

Algorithm 3 is computationally expensive. We propose an efficient implementation of it, described in Algorithm 4. It produces similar results while limiting redundant computations. Algorithm 4 is also geared towards implementation on modern hardware such as MIC (Many Integrated Core) or GPUs (graphics processing units) by separating the parts of the code that can be run in parallel. To approximate the CPDF and to determine a narrowness criterion, we use pattern matching to obtain a set of candidate pixels from the training image. For each simulated pixel, once a sufficient number of candidates is available, the CPDF of the target value can be determined empirically. In practice, we define the set of candidates by taking the

k candidates that have the smallest matching error. Then, we formulate the narrowness criterion for the CPDF as a constraint on the interquartile range (IQR).

Pseudocode of Algorithm 4: an efficient implementation of Algorithm 3

Inputs:
training image
target image
k number of candidates to estimate the narrowness

1 **//Initialization loop**
2 **For** each pixel $B(\boldsymbol{v})$ of the target image do independently
3 Extract the neighborhood $N(\boldsymbol{v})$ of the target image around $B(\boldsymbol{v})$
4 Find the set of k candidates in the training image with lowest $\varepsilon(\boldsymbol{u}, \boldsymbol{v})$
5 Compute and store the narrowness based on the candidates in $E(\boldsymbol{v})$; select and store one candidate $B(\boldsymbol{u})$ randomly chosen among the set in $C(\boldsymbol{v})$.
6 **End**
7 **//Simulation loop**
8 **Until** all positions are simulated **do**
9 Find the position $\boldsymbol{v}$ with the lowest narrowness $E(\boldsymbol{v})$ among all non-simulated pixels in E.
10 Import the color from $B\big(C(\boldsymbol{v})\big)$, at location $B(\boldsymbol{v})$ in the target image.
11 **For** all non-simulated pixels $B(\boldsymbol{v})$ of the target image that are directly adjacent to the simulated pixel (including diagonals) do independently
12 Extract the neighborhood $N(\boldsymbol{v})$ of the target image around $B(\boldsymbol{v})$ taking into account previously simulated values
13 Find the set of k candidates in the training image with lowest $\varepsilon(\boldsymbol{u}, \boldsymbol{v})$
14 Compute and store the narrowness based on the candidates in $E(\boldsymbol{v})$; select and store one candidate $B(\boldsymbol{u})$ randomly chosen among the set in $C(\boldsymbol{v})$.
15 **End**
16 **End**

The two main algorithmic parameters are the kernel and the number of candidates k.

Note that in this algorithm most computations are done in the initialization part [Line 2-6 Algorithm 4] and updating part [Line 11-15 Algorithm 4], where operations on each pixel are independent and can be parallelized.

3.3 Numerical tests

In this section, we test the colorization algorithms presented in Section 2 in different scenarios. In the first example, the target multispectral image is known, so that it provides a reference against which the results can be validated. This case is used to study both the

spectral disaggregation (PAN→RGB) and spectral extrapolation (PAN→NIR) problems. In the second case study, we test the colorization (PAN→RGB) of some of the oldest satellite acquisitions, the Corona images, for which no reference color is available. For all the following tests, we generate 10 independent realizations for each technique, using an exponential kernel with $\alpha = 4.5$. For NDS $k = 10$, and for DS, $f = 0.3$ and $t = 0$.

The validation of the results is performed based on well-established metrics: the root mean square error (RMSE), correlation coefficient (CC) and Structural Similarity (SSIM) (Wang et al., 2004).

3.3.1 Landsat-8 test case

To test and compare the methods presented in Section 2 in the context of colorization, we apply them for the colorization of a raw OLI (Operational Land Imager) Landsat-8 scene of the Swiss plateau, acquired on August 30th, 2015. The scene is separated in two subareas of 500 x 500 pixels, which are shown in Figure 3.3: the first one is used as training image and the second one as target, where we use the approaches presented to restore missing spectral bands. Both images present similar features, including mostly small urban areas, forests, and crops.

Usually, satellite sensor images are acquired under varieties of conditions, such as Sun position. Many of these variations can be modelled with a linear transformation. For practical reasons, each band is standardized (mean=0 and standard deviation=1) separately, which results in a homogeneous representation between bands. If required, the back-transformation is trivial to apply.

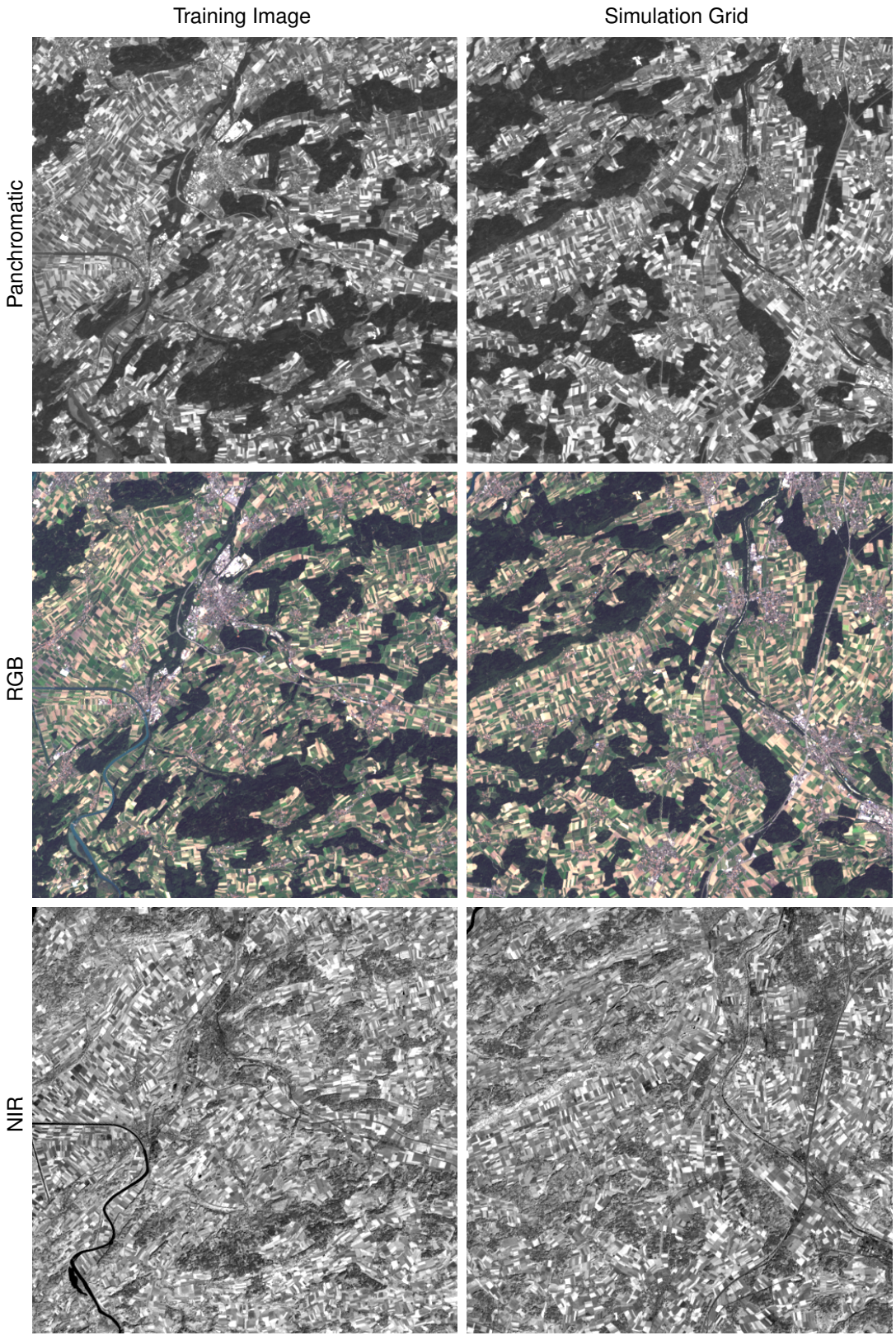

Figure 3.3: Training Image and reference scenes in Panchromatic (PAN), RGB and NIR. Landsat id: LC81950272015242LGN00.

3.3.2 Spectral disaggregation

The spectral disaggregation is applied with each of the three algorithms presented in Section 2. The training image is composed of the panchromatic (PAN), red (R), green (G) and blue (B) channels. The target image contains only the PAN channel, which corresponds to the band 8 on Landsat 8. This band originally has a 15 m spatial resolution, but in this example, it was interpolated to 30 m using bicubic interpolation to match the spatial resolution of the color bands.

The output of the simulations, shown in Figure 3.4, are visually satisfying for all three algorithms. However, when zooming in on the colorized images it is possible to observe that algorithm 4 produces less artifacts or noise than algorithms 1 and 2. In detail, it is clear that NDS produces more structured textures. For example, the NDS results depict a more accurate reconstruction of the forest colors (the bottom right of the zoom in Figure 3.4) and a better preservation of features' edges; the roads are more clearly visible, for example (Zoom Figure 3.4). These visual interpretations are confirmed by the error metrics. This is confirmed by the quantitative comparison results given in Table 3.1.

		RMSE	CC	SSIM
Algorithm 1 (Welsh)	Blue	0.1808 ± 0.0049	0.9251 ± 0.0036	0.7292 ± 0.0005
	Green	0.1719 ± 0.0020	0.9581 ± 0.0010	0.7462 ± 0.0008
	Red	0.2063 ± 0.0019	0.9602 ± 0.0007	0.7542 ± 0.0006
Algorithm 2 (DS)	Blue	0.1635 ± 0.0076	0.9378 ± 0.0050	0.7577 ± 0.0011
	Green	0.1621 ± 0.0032	0.9629 ± 0.0013	0.7681 ± 0.0007
	Red	0.1925 ± 0.0032	0.9655 ± 0.0011	0.7765 ± 0.0006
Algorithm 4 (NDS)	Blue	0.1385 ± 0.0053	0.9546 ± 0.0033	0.8259 ± 0.0010
	Green	0.1338 ± 0.0018	0.9744 ± 0.0007	0.8430 ± 0.0004
	Red	0.1542 ± 0.0017	0.9778 ± 0.0005	0.8447 ± 0.0007

Table 3.1: Disaggregation error for each algorithm, shown separately for the Blue, Green and Red bands. For RMSE, smaller is better, and for CC and NSME closer to 1 is better. The best result for each metric is shown in bold.

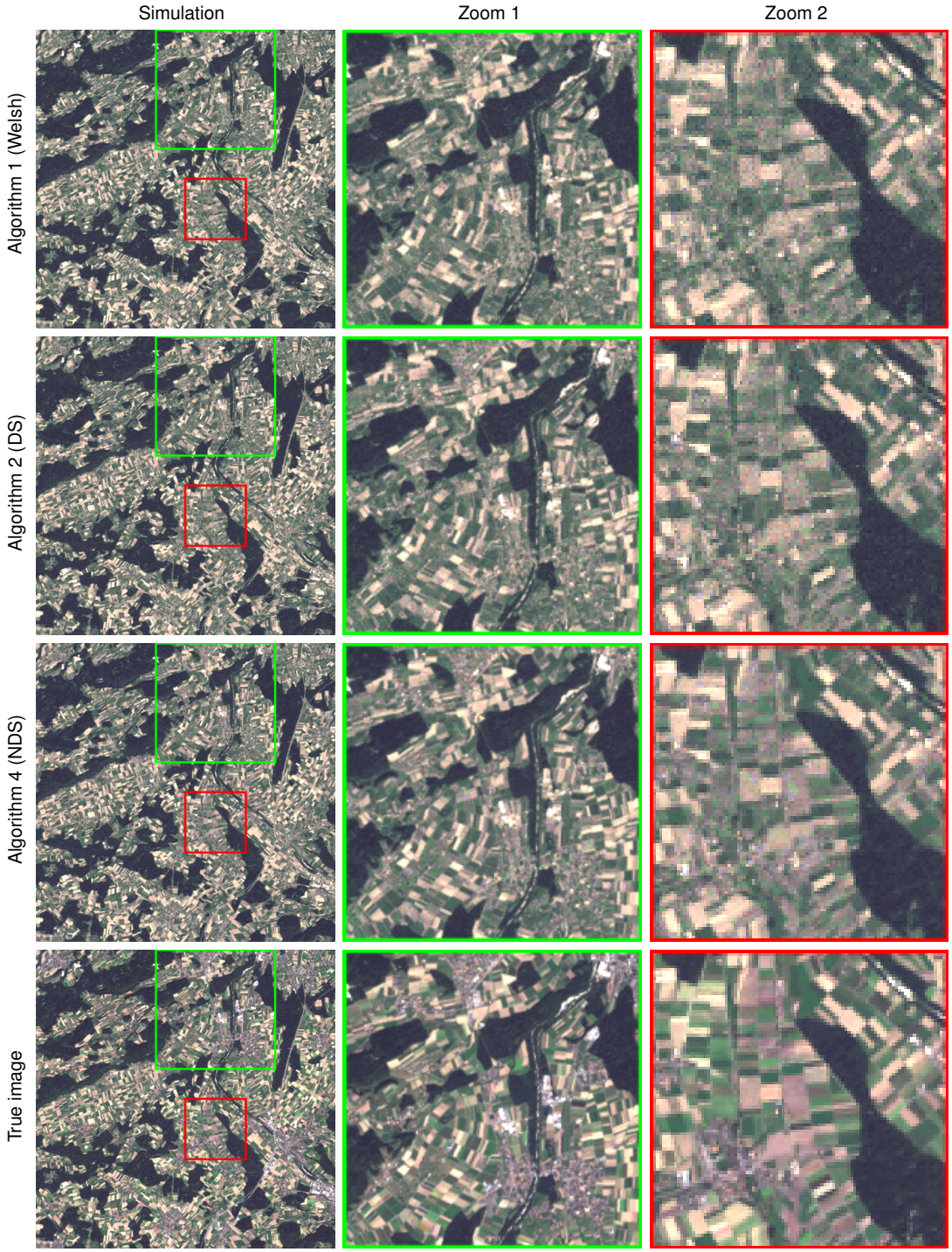

Figure 3.4: Spectral disaggregation from PAN to RGB for all three algorithms.

3.3.3 Spectral extrapolation

For spectral extrapolation, we use the same training and target images. Two series of tests are shown. In the first series, the unknown NIR band is simulated based on the known PAN band, with all three algorithms (PAN→NIR). In the second series, we simulate again the NIR, but this time, based on the three RGB color bands (RGB→NIR). The results are reported

in Table 3.2 and Figure 3.5 for PAN→NIR (black in Table 3.2) and Figure 3.6 for RGB→NIR (red in Table 3.2).

		RMSE	CC	SSIM
Algorithm 1 (Welsh)	PAN → IR	0.8712 ± 0.0019	0.3555 ± 0.0017	0.1097 ± 0.0009
	RGB → IR	0.4886 ± 0.0006	0.7763 ± 0.0005	0.4072 ± 0.0008
Algorithm 2 (DS)	PAN → IR	0.7747 ± 0.0021	0.4009 ± 0.0028	0.1202 ± 0.0015
	RGB → IR	0.4485 ± 0.0007	0.8048 ± 0.0007	0.4238 ± 0.0014
Algorithm 4 (NDS)	PAN → IR	0.7364 ± 0.0018	0.4609 ± 0.0048	0.1589 ± 0.0020
	RGB → IR	0.4110 ± 0.0008	0.8376 ± 0.0006	0.4733 ± 0.0014

Table 3.2: Extrapolation error for all three algorithms, shown separately for the PAN → NIR and for RGB → NIR. For RMSE, smaller is better, and for CC and SSIM closer to 1 is better. The best result for each metric is shown in bold.

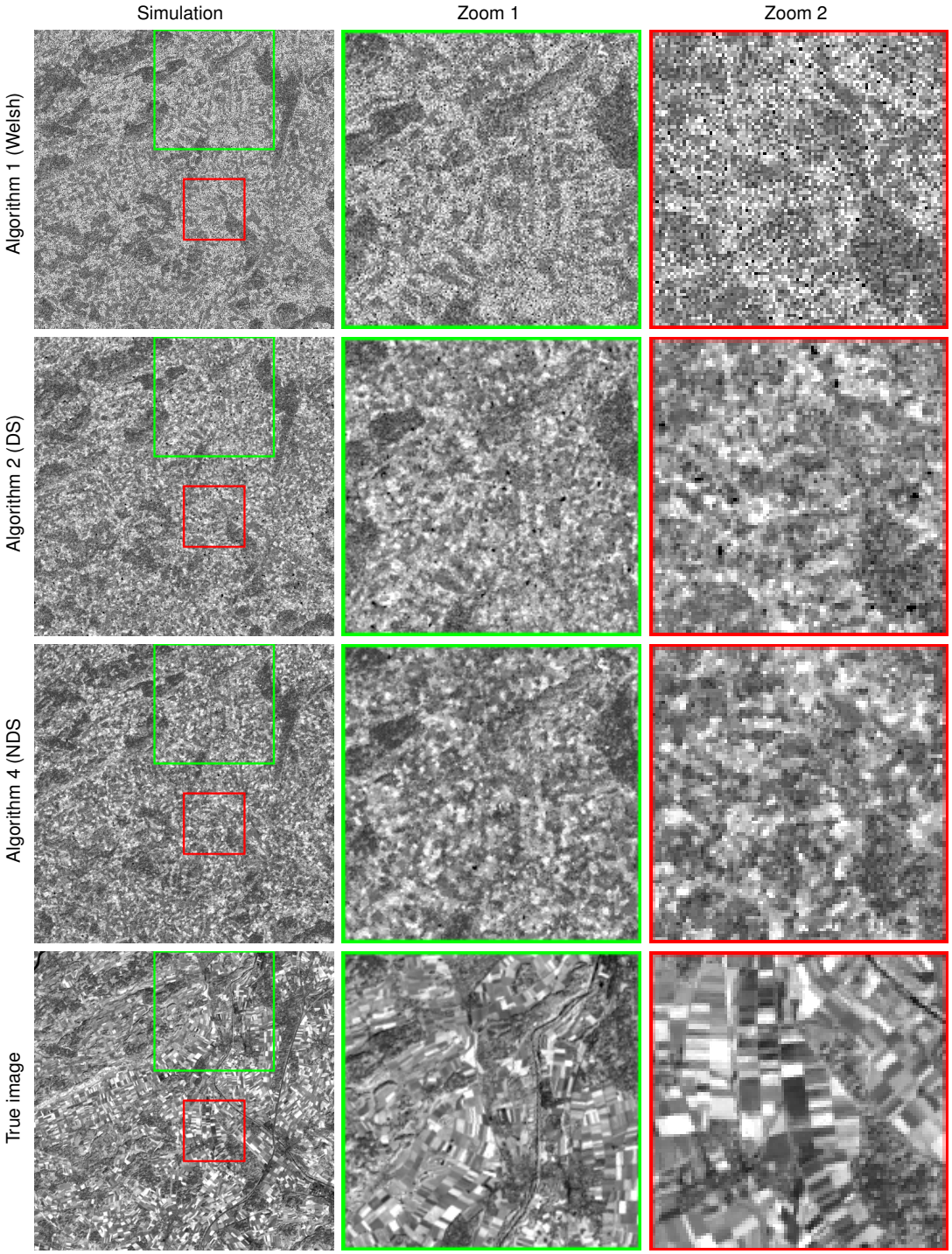

Figure 3.5: Spectral extrapolation PAN→NIR for all three algorithms.

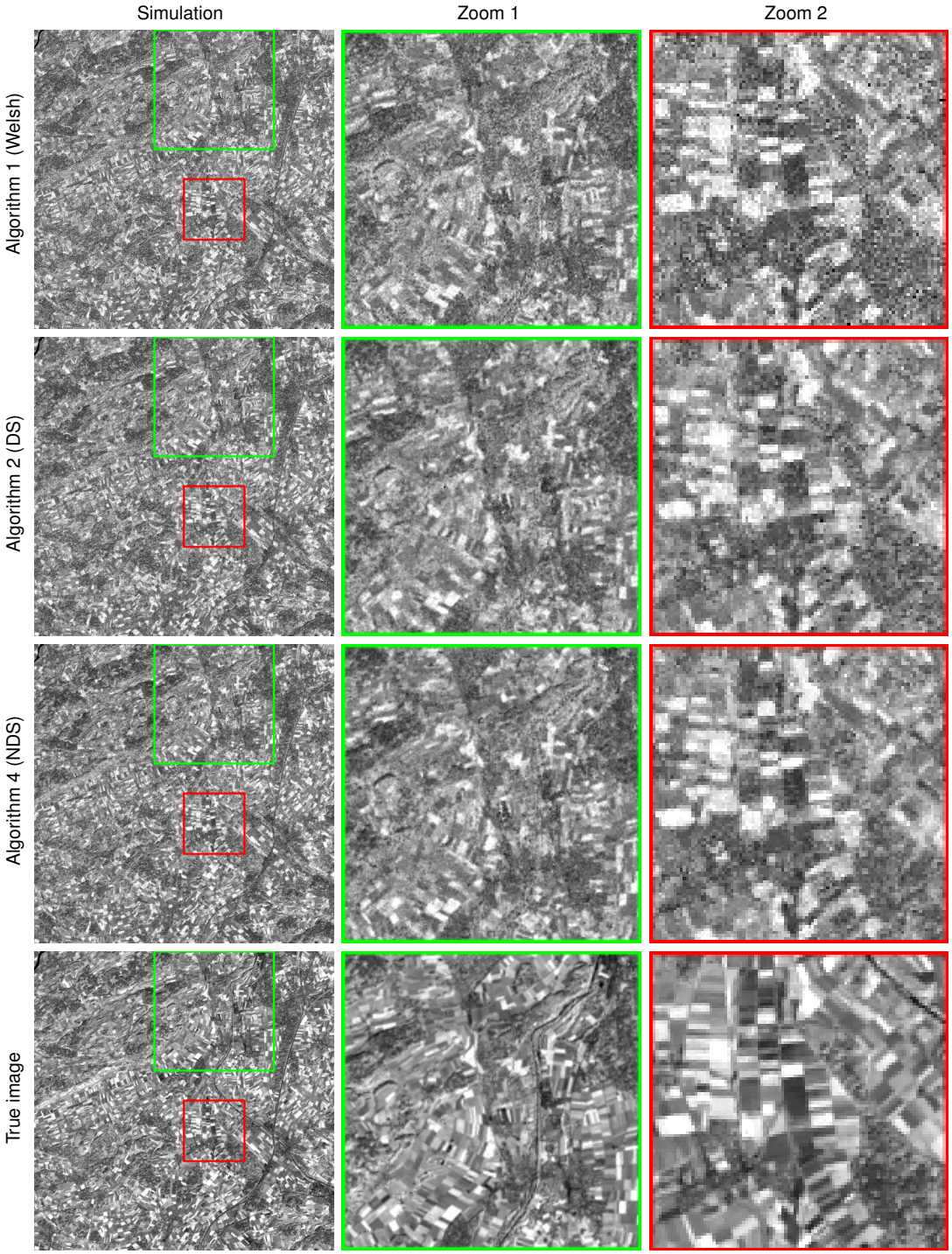

Figure 3.6: Spectral extrapolation RGB→NIR for all three algorithms.

3.3.4 Sensitivity of the NDS parameters

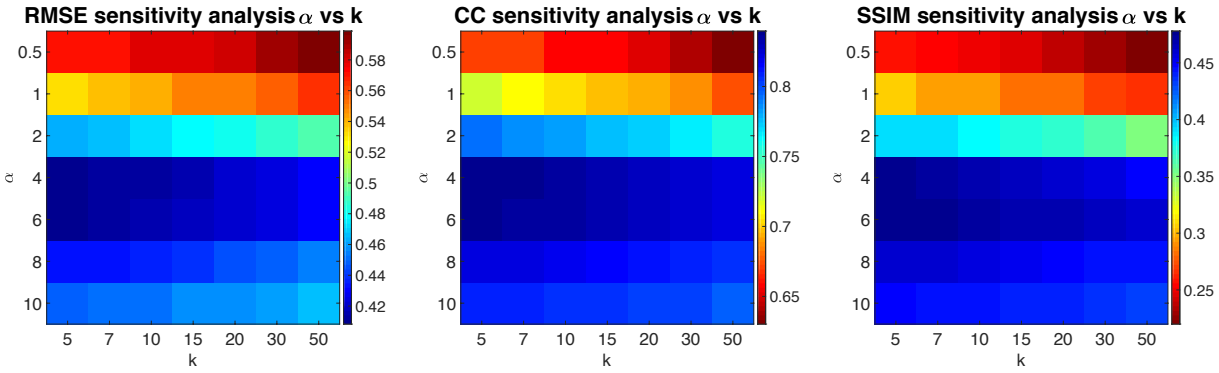

Figure 3.7 Sensitivity of the NDS parameters using three different error metrics

To better understand the behavior of the proposed NDS algorithm, we perform a sensitivity analysis to determine the influence of each parameter on the simulation performance. The sensitivity analysis in based on the case of the RGB→NIR simulation. The most influential parameters in the NDS algorithm are *k* and α, hence the analysis is based on these only. A series of values for k and α were initially predefined, and we ran a single simulation for each parameter combination because the variability between realizations is negligible when compared to the magnitude of the errors as showed Table 3.2. Each set of parameters is then assessed by computing the RMSE, CC and SSIM metrics.

Figure 3.7 shows that regardless of the metric, the sensitivity to parameter α is dominant compared to parameter *k*. While the error does not change significantly between $k = 5$ and $k = 20$, the error space seems convex along α. It is also clear that, for this case study, the calibration of α is more important than that of k. Both variables are globally independent, thus we advise for practical applications to perform in priority a sensitivity analysis over α.

3.3.5 Corona test case

As a real-life test of our approach for enriching old satellite sensor data, we applied it to the colorization of declassified Corona spy imagery (USGS, 2008). Such images were originally acquired on an argentic film, which was sent back to Earth in capsules, and digitalized many years later. Only the panchromatic band is available, with a variable spatial resolution between missions. The images themselves present a number of artifacts due to their age, or to the recuperation process, or artifact (lines) that are introduced by the digitalization process. Our goal here is to reconstruct a full RGB and NIR image with

comparable spectral and spatial resolutions as Landsat 8 (Figure 3.9 and Figure 3.10). We use a Corona image acquired on the 13th of July 1975 over the same type of Swiss plateau landscape as in the case study of Section 3.1. As a consequence, the same training image as in Section 3.1 is used (Figure 3.3). We compared the performance of algorithms 1, 2 and 4 in this exercise. It is important to note that it is impossible to validate such simulations quantitatively because there is no reference data to compare with, however a qualitative evaluation is possible.

Due to the specific nature of Corona images (based on film photography, with problems related to over and under-exposition), some preprocessing steps had to be applied before the colorization process:

- The Corona image that has originally a fine spatial resolution was upscaled using a bicubic interpolation to a spatial resolution of 30 m in order to correspond to the resolution of the Landsat-8 image.
- The histogram of the values in the grayscale Corona image did not correspond to the histogram of the panchromatic Landsat-8 image, due to a saturation of the high and low values. Since all colorization approaches rely on corresponding training /target images pairs, we performed a histogram adjustment of the Landsat-8 panchromatic band based on a quantile-quantile mapping, as illustrated in Figure 3.8. This allows having corresponding patterns in the Corona and in the Landsat-8 panchromatic band.

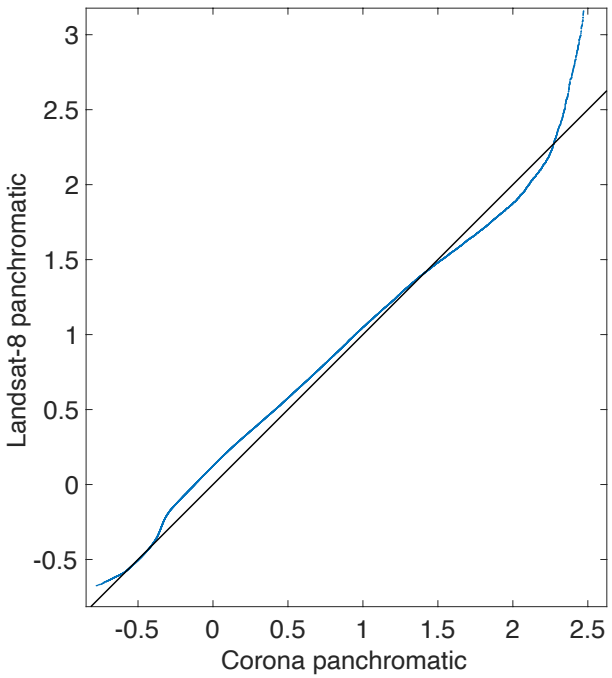

Figure 3.8 q-q plot between corona and Landsat 8 panchromatic bands

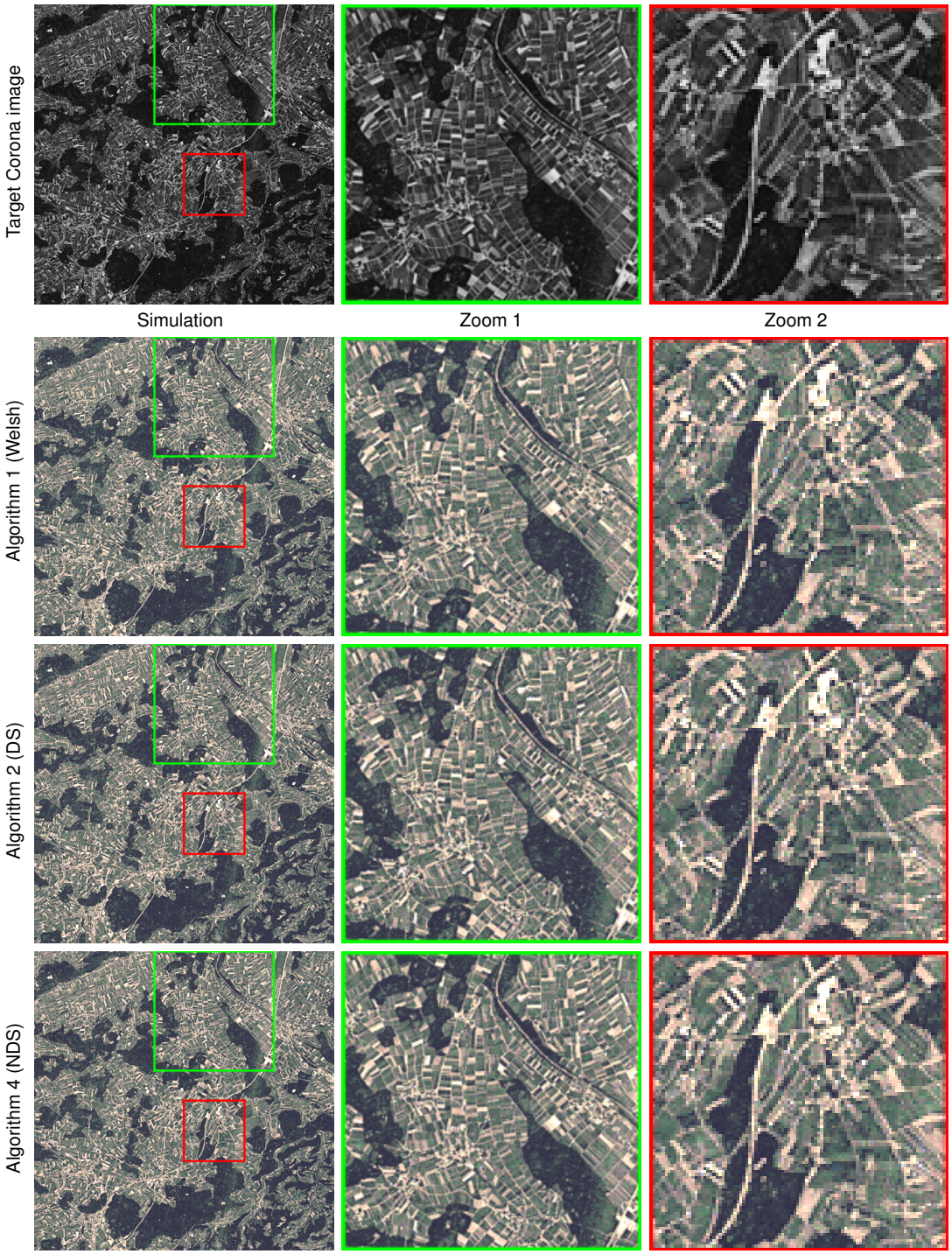

Figure 3.9: Spectral disaggregation from PAN to RGB for all three algorithms on Corona images

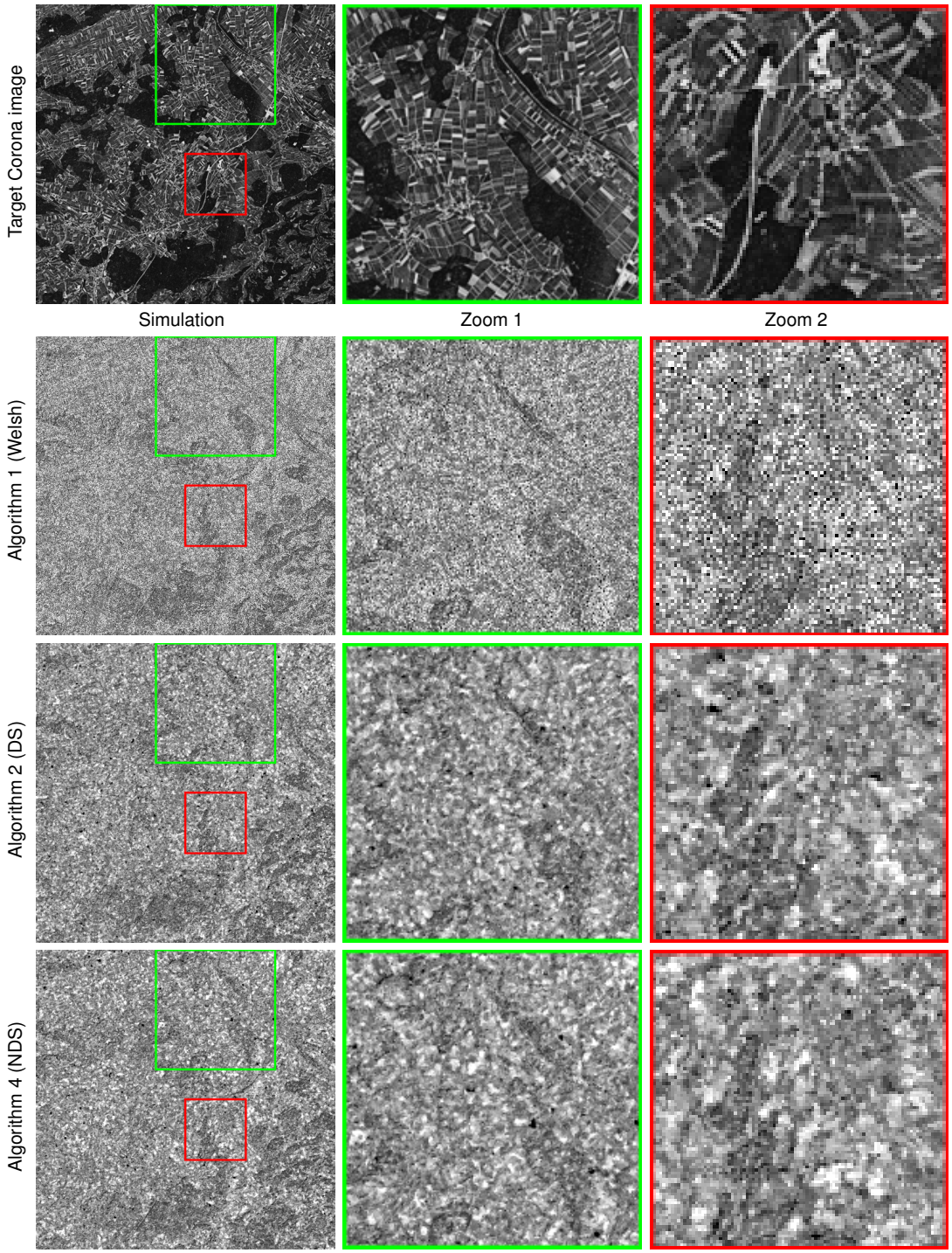

Figure 3.10 Spectral extrapolation from PAN to NIR for all three algorithms on Corona images

3.4 Discussion

All three algorithms show acceptable results for the case of disaggregation, which can be explained by considering that the color information is present in a convoluted way:$PAN = \int RGB$. This results in a strong dependence between the known and unknown variables. While all methods produce color images that are reasonably similar, NDS results are systematically more accurate than DS, and Algorithm 1 shows systematically the lowest

performance. However, the accuracy of the results is inversely proportional to the algorithm computational cost: NDS simulations are usually one order of magnitude slower (about 30 minutes on a bi-Xeon machine) than Algorithm 1 (about 3 minutes). The cost of algorithm 2 (DS) lies in-between, and varies depending on the parameterization. Accordingly, we recommend to always start by applying Algorithm 1 for disaggregation as this method often provides acceptable results at a relatively low cost. The NDS approach is recommend only if very high-quality spectral enhancement is required.

The recommendation is different for applications dealing with spectral extrapolation, where Algorithm 1 results in highly noisy images. This can be explained by the fact that spectral extrapolation is challenging, as can be seen in Figure 3.11. It is clear that the dependence between the PAN and NIR bands is non-linear, whereas the dependence between the PAN and Green bands is close to linear.

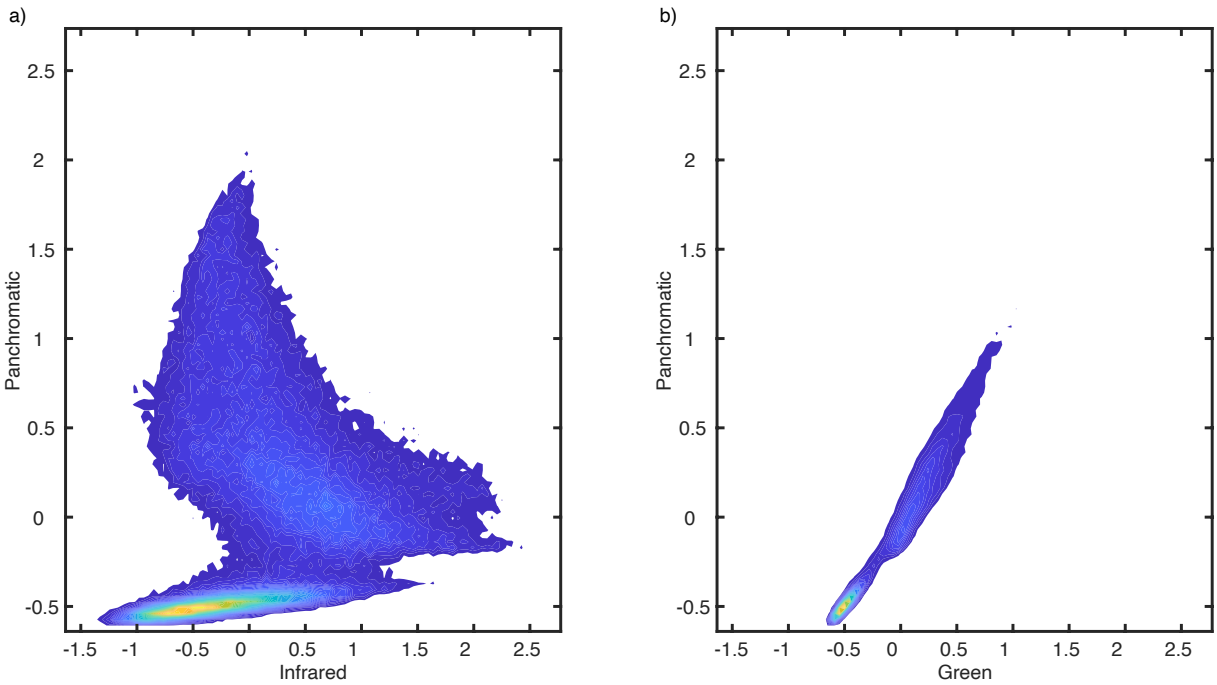

Figure 3.11: Joint probability distributions of Landsat 8 data, a) NIR and Panchromatic bands b) Green and Panchromatic bands

For spectral extrapolation, NDS results are clearly more accurate than Algorithm 1. This is explained by the fact that NDS takes into account the dependence between neighborhoods, which allows constructing coherent textures, and thus reducing noise. This link between neighbors results in the successive simulation of nearby pixels (see video in supplementary material). Compared to DS, the narrowness constraint of NDS reduces the occurrence of incoherent neighborhoods, which results in increased colorization accuracy of features edges

and textures. The result of the NIR colorization of Corona imagery clearly show that NDS can capture more complex structures, as visible in the zooms of Figure 3.10 (especially in zoom 2).

The most important parameter, which is common to all three algorithms considered, is α which controls the shape of the kernel, as seen at Figure 3.7. Indeed, it defines the relative importance between the spectral constraint on the pixel and the spatial constraint. Therefore, α is a very important parameter that needs to be set carefully. Currently, cross validation on the training image appears to be the best approach to this end. Our tests show that $\alpha = 2$ is usually a good starting point, which attributes approximately half of the sum of weights to the central pixel. However, this value is not universal as for the example in Section 3.1, $\alpha = 4.5$ provided the best setup.

NDS has significantly more accurate results, however it requires two additional parameters. The first one is k, the number of candidates to consider for computing the narrowness. The minimum value for k is around 5 to allow a proper measurement of the narrowness of the CPDF. If the training image is large and presents enough repetition of patterns, k can be increased to 30 to allow for a more reliable narrowness measurement. The second parameter refers to the statistical dispersion metric used to compute the narrowness. In this paper, we use the interquartile range, which is a robust non-parametric dispersion measure, and not very sensitive to extreme values.

A common challenge of the proposed methods is to find a compatible training image. Our tests show that a good starting point is to use a present-day acquisition of the same or a neighboring area. In the particular case of important changes in the landscape configuration (such as for example urban growth, deforestation or disasters such as forest fires or tsunamis), expert knowledge is required to find a suitable analogue area to use as training image.

The generality of the proposed framework allows envisioning new possibilities related to the spectral readjustment of sensor, like to convert the blue-green and green-red bands of Landsat 1 to the blue, green and red bands of Landsat 8. More generally it can be extended to new data sources such as archive aerial photography, or for the disaggregation of near infrared on filterless standard cameras, which could be used to measure NDVI at low-cost.

3.5 Conclusion

Remotely sensed data are available for more than 45 years; however, only data from the last 30 years is being widely used. Analogue-based colorization algorithms provide an

opportunity to use data from very early sensors, allowing for the integration of archive data that were previously unused, such that they can be exploited and compared to recent data sources. In this paper, propose a novel and general method with high computational cost (Algorithm 4), and the use of another much faster approach (Algorithm 1), but only reliable in the case of spectral disaggregation.

These methods can potentially be generalized to produce data with much finer spectral resolution, if compatible training images are available. For example, the methods presented in this paper could be applied to data acquired from hyperspectral airborne or satellite-based imagery such as Hyperion EO-1 data. As such, training-based colorization could offer an interesting alternative to spectral band reconstruction approaches based on regression, which do not account for spatial patterns (Hoang and Koike, 2018; 2017). In addition, the proposed algorithms can be useful to improve image classification techniques, by synthesizing the missing spectral bands of the target image. Finally, of the methodology can be potentially extended for downscaling of both spatial and temporal information simultaneously.

3.6 Code availability

Source code and documentation of the NDS simulation algorithm is available at: https://github.com/GAIA-UNIL/G2S

Platform: Linux / macOS / Windows 10

Language: C/C++

Interfacing functions in MATLAB and Python3

3.7 Author contribution

GM proposed the idea of using MPS for spectral enhancement. MG implemented all the algorithms and realize all the tests. MG propose the algorithmic solution at the root of NDS. GM provided supervision, methodological insights. MG and GM wrote the manuscript. LGR provide key methodological observation, and a careful proofreading of the paper.

3.8 Acknowledgments

We are thankful to Thomas Hansen, Peter Atkinson, Julien Straubhaar and Raphaël Nussbaumer for their insightful comments that helped improve this paper. We also thank two anonymous reviewers for their constructive comments.

4 Automatic parameterization of Multiple point statistics algorithms by pattern-based analysis of the training image

Mathieu Gravey[a] (mathieu.gravey@unil.ch), Grégoire Mariethoz[a]
[a] University of Lausanne, Faculty of Geosciences and Environment, Institute of Earth Surface Dynamics, Switzerland

Highlights

- An objective function-free MPS calibration
- Adaptative calibration as a function of the simulation progression
- Calibration only depending on the training image

Abstract

Multiple-point geostatistics are vastly used to mimic simulations of complex spatial structures based on a training image. The use of these methods relies on the possibility of finding optimal training images and parametrization. The parametrization in particular can be a cumbersome task. Here we propose to find an optimal set of parameters using only the training image as input. The main advantage over previous work using parametrization optimization is that it requires no complex objective function. Indeed, the proposed approach is solely based on the analysis of the training image to find an optimal set of parameters. Its principle is to analyze the errors that occur when filling artificially constructed patterns borrowed from the training image. The main advantage of our approach is to remove the risk of overfitting an objective function, which may result in underestimating the variance or verbatim copy of the training image. Since it is not based on an optimization, our approach finds a set of acceptable parameters in a predictable time, by using knowledge and understanding of how the algorithms work. This technique was developed for the recently developed Quantile Sampling algorithm, but it can be easily adapted to other pixel-based MPS algorithms using pattern matching, such as Direct Sampling.

4.1 Introduction

Geostatistics are extensively used to map spatial variables such as surface and subsurface features. Originally developed to estimate ore grades in mines, geostatistics were then extended to many fields from subsurface modeling to meteorology forecasting, and have evolved from estimation to simulation of phenomena.

Traditional two-point statistics preserve the histogram and variogram inferred from point data (Matheron, 1973). However, inherent limitations make realization of complex structures impossible. Multiple point statistics (MPS), taking into account more complex relations, enables the reproduction of complex structures (Guardiano and Srivastava, 1993). MPS come with their own limitations (Mariethoz and Caers, 2014). To perform good simulations, MPS algorithms require analog images (called training images) and appropriate parametrization. Training images can often be provided by expert knowledge. Indeed, the training image is related to the property to simulate and, therefore, it is common to all the MPS algorithms. However, the parametrization of an MPS algorithm depends not only on the chosen training image, but also on the specifics of the algorithm. This makes the task of finding a good parametrization cumbersome, and a trial and error approach is often used (Meerschman et al., 2013).

Over the last few years, many studies have addressed the challenge to find good MPS parameters. These can be categorized in two different philosophies. The first approach is “simulation grid” focused, assuming that a parametrization is related to the simulation grid, the training image and MPS algorithm. Dagasan et al. (2018) proposed using a few hard data known from the simulation grid as a reference for statistics and metrics, then trying to improve the parametrization through a simulated annealing optimization process until matching the metrics as well as possible. The second approach is “training image” focused and assumes that the parametrization is only related to the training image and MPS algorithm. Along these lines, Baninajar et al. (2019) proposes the MPS-APO method. This approach is based on the cross-validation of the TI to quantify simulation quality and CPU cost. In this approach, artificially generated gaps in high gradient areas of the training image are created, and MPS algorithms are used to simulate the gaps. The performance of a particular parameterization is quantified by assessing the correspondence between filled and original training data. By design, this approach is extremely interesting for gap-filling problems. The authors state that it can be

used for parametrization of unconditional simulations; however, the use of limited gaps cannot guarantee reproduction of long-range dependencies. Furthermore, due to the design of the framework for generating gaps, each MPS algorithm needs to be able to handle gap-filling problems for the error to be estimated properly.

If both approaches show good results, they are both related to optimization methods, and therefore we have no control over the duration of the optimization process. Furthermore, an objective function is required. Finding this objective function is a challenge in itself, because it can change depending on the training image used. Using an optimization approach, many metrics can be taken into account in the objective function, such as for example: histogram, variogram, pattern histogram, connectivity function, Euler characteristic and other multiple-point pattern reproduction characteristics (Boisvert et al., 2010; Renard and Allard, 2013; Tan et al., 2013), or a weighted combination of them. Similarly, one has to define the meta-parameter linked to the optimization algorithm itself, such as cooling rate or maximum number of iterations. As a result, MPS parameter optimization approaches tend to be complex and difficult to use.

In this contribution, we propose to skip the complexity of an optimization algorithm, and instead simplify the optimization procedure to a key element: the simulation of a single pixel. The underlying principle of our approach is that a sequence of well-simulated pixels converges to a good simulation overall. Therefore, the goal is to find the optimal parameters to simplify the simulation of a single pixel using the training image as our only reference. Baninajar et al. (2019) shows that computing the prediction error (i.e., the error between the simulation and the reference), is an adapted metric to find optimal parameters. Following this approach, we propose to exhaustively explore the parameter space by performing pixel predictions over patterns artificially generated from the training image, and use these predictions to compute the associated prediction error. This results in a prediction error map for each combination of parameters.

The remainder of this chapter is structured as follows: Section 2 presents the proposed method. Section 3 evaluates the approach in terms of quantitative and qualitative metrics. Section 4 discusses the strengths and weaknesses of the proposed approach and presents the conclusions of this work.

4.2 Method

The objective of the presented approach is to find an optimal set of parameters using only the training image and the simulation algorithm as information. The main target application of the presented approach is for pattern matching algorithms like Direct Sampling (DS) (Mariethoz et al., 2010) or Quantile Sampling (QS), where the values are sampled directly from the training image, using continuous variables. However, the method is suitable for categorical variable simulations as well.

4.2.1 Background on pixel-based simulation

Simulation algorithms like QS or DS can be summarized by Algorithm 1. The key operation is at line 3, when the algorithm searches for an optimal match based on the conditioning.

Pseudo code algorithm 1 (inspired from (Mariethoz and Caers, 2014))

Inputs:
T the training images
S the simulation grid, including the conditioning data
P the simulation path
θ the parametrization (including n: number of neighbors)

1 **For** each unsimulated pixel x following the path P **do**:
2 Find the neighborhood $N(x)$ in S composed of the n closest neighbors
3 Find a candidate in T that matches $N(x)$ using θ
4 Assign the value of the selected candidate to x in S
5 End

4.2.2 Proposed calibration approach

Here, we propose to apply a divide and conquer approach, by dividing any pixel-based sequential simulation to its atomic operation: the simulation of a single pixel. We assume that all pixels are perfectly simulated. By perfectly simulated we mean the conditional probability distribution function such as k the quantile for QS, th the threshold, f the scanning faction for DS, and n the number of neighbors or the α parameter of a kernel.

If we consider the simulation of a pixel, many values can potentially be valid.

$$|\{A|P(A|N(x)) > 0\}| \geq 1 \quad (4.1)$$

Where | . | represents the cardinality of a set. $P(A|N(x))$ represents the probability of A (a given value) knowing $N(x)$, the neighborhood. Each of these possibilities will still respect the probability distribution.

The proposed approach consists of reducing the pool of candidates, meaning that we are looking for a set of parameters that results in the most unique solution of each pattern. This extreme situation results in a simulation of values resulting from a sampling of perfect matches (the neighborhood is available in the training image). This results in a simulation identical to the training image. This last phenomenon is called verbatim copy.

The search for the optimal parameterization is carried out by exhaustive exploration (Algorithm 2). The prediction error, the error between the original value of the pattern and the value of the selected, is computed (Figure 4.1).

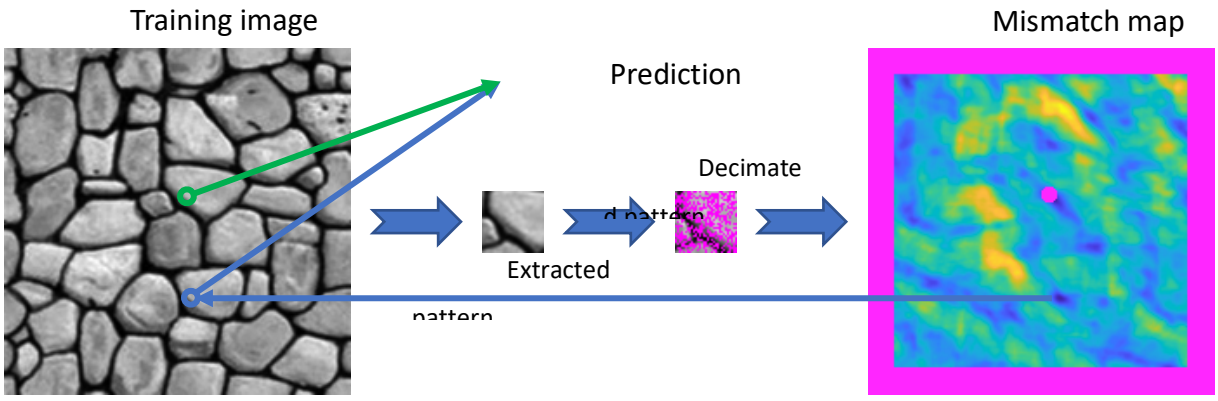

Figure 4.1 All steps for a single pattern, summarizing Algorithm 2, lines 2-4.

This proposed algorithm explores a discretized parameter space θ (line 1). For some parameters this discretization is natural, such as n that is an integer. Some other parameters such as the threshold th requires an explicit discretization. We explore the parameter for a number of representative mimicked progressions D of the simulation (line 1). In a case of random paths, the progression of a simulation is directly related to the density of the neighborhood. For each combination of D and θ multiple measures over a large ($500 < |V| < 10000$) set of random locations V are averaged to improve the robustness of the metric. This is summarized in Algorithm 2 lines 1-5 and mathematically in Equation (2.1).

Pseudo code algorithm 2

Inputs:
List of D, and list of θ
T the training image

1 **For** each possible combinations of D and θ, **do** for all $v \in V$ (V is randomly generated):
2 Generate a neighborhood $N(v)$ from T respecting D
3 Using θ, find a candidate in T that matches $N(v)$, except for v itself
4 Compare the value of the selected candidate to the value of v
5 End
6 Compute the errors ε and determine the best θ for each D.

$$\varepsilon(\theta, D, T) = \sqrt{\frac{1}{|\mathcal{V}|}\sum_{v \in \mathcal{V}}\left(\mathrm{Z}(v) - \mathrm{Z}\left(\underset{T\setminus\{v\}}{\mathrm{Cand}}\big(\theta, N(v, D)\big)\right)\right)^2}, \tag{4.2}$$

where $Cand(\theta, N)$ returns a single candidate position for a given neighborhood N and following the parametrization θ. $N(v, D)$ represent a decimated neighborhood around v that respects the condition D. V represents a random set of positions in the training image. $Z(v)$ refers to the actual values at a position v in the training image.

Finally, for each progression considered, the set of parameters with the minimum associated error ε are considered as the optimal parameters for this progression (Equation (4.3) and Algorithm 2, line 6)

$$\varepsilon\big(\theta_{optimal}, D, T\big) = \min_{\theta} \varepsilon(\theta, D, T). \tag{4.3}$$

To improve the quality of the metric during the search for patterns, the position v and its direct neighborhoods are removed from the set of potential candidates to avoid over-constrained situations that generate verbatim copies of the training images. Furthermore, in case of equality between several optimal options, we propose the simple rules of taking the cheapest parameter set in terms of computational cost.

4.2.3 An efficient implementation

In practice, the implementation of Algorithm 2 divides θ into two subsets of parameters, θ_h and θ_s. θ_h contains all parameters that can change the matching itself: here only n the number of neighbors. θ_s includes the parameters related to the sampling – in case of QS, k the quantile, and in case of DS, t the threshold and f the fraction of the training image

that is scanned. This approach allows computing and storing in advance all matches for a given parameterization θ_h. Then the saved matches of θ_h can be used to quickly measure all possibilities for the parameters in $\theta = \theta_h + \theta_s$. This two-step approach allows us to significantly reduce redundant computations. Furthermore, it is possible to improve the speed of this algorithm to easily observe that with few values of V it is often possible to estimate that ε will be too high, and then skip the operation to have a more precise estimation. To realize this last operation, the algorithm increases V for the parameter combinations of interest.

4.3 Result

All experimental tests in this section are done using the stone training image visible in Figure 4.1. The set D of progressions considered are distributed following a log scale, because it is the relative number of informed nodes that constrains the simulation. (e.g., the second pixel is highly constrained by the first one. However, the one before the last pixel, has a limited influence on the last pixel). Furthermore, experimentation shows that the first few percent of the simulation are critical for the overall simulation. The current Python implementation of the proposed method allows getting results in 30 to 40 minutes.

4.3.1 Automatic calibration for QS

In the case of QS, the method finds optimal values for: k the quantile, n the number of neighbors and ω the kernel.

4.3.1.1 Automatic calibration for QS with a uniform kernel

In this first test using QS, we use the configuration $\theta_h = \{n\}$ and $\theta_s = \{k\}$. Results are shown in Figure 4.2. The ignorance threshold represents the point above which no information is extracted from the pattern, and values can be simply drawn from the marginal distribution without introducing bias.

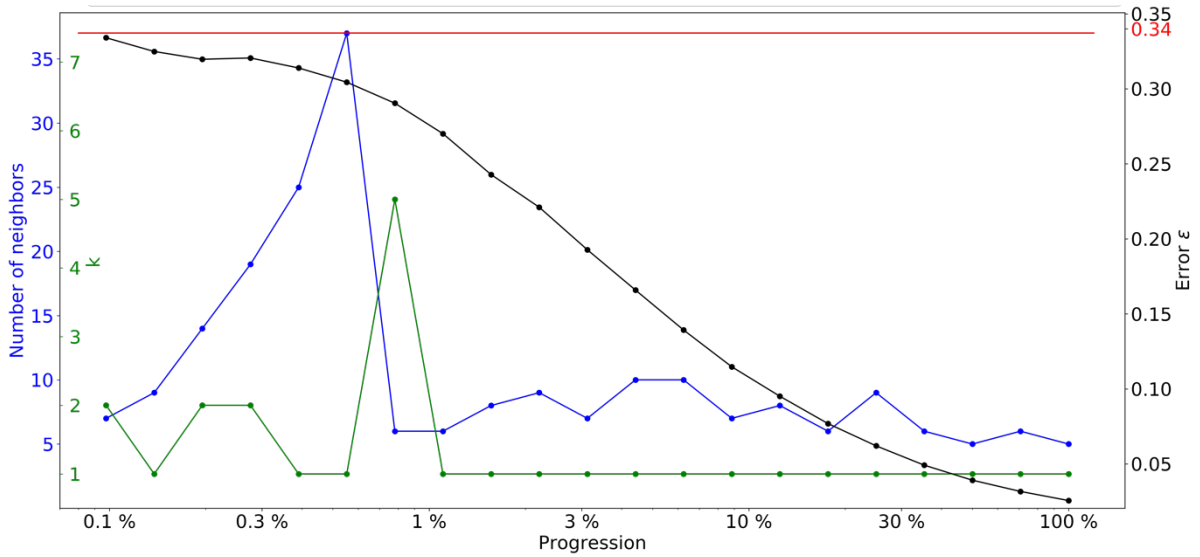

Figure 4.2 Optimal parameters for QS (k and number of neighbors) in function of the progression, with the associated prediction error. The red line represents the ignorance threshold.

Figure 4.2 shows the optimal quantile k and number of neighbors as a simulation of the simulation progression. We can observe that k remains relatively small because the training image is not repetitive enough. At the early stage of the simulation, it seems important to use a large number of neighbors. However, the subsequent drastic reduction in the number of neighbors tends to indicate that once the large structures are informed, only the few direct neighbors are important. We also note that the parametrization is generally difficult to predict. This indicates that the use of a single parametrization for the entire MPS simulation is generally suboptimal. Figure 4.2 also shows that the first few simulated pixels are hardly predictable (close to the ignorance threshold) and therefore can potentially be just drawn from the marginal distribution.

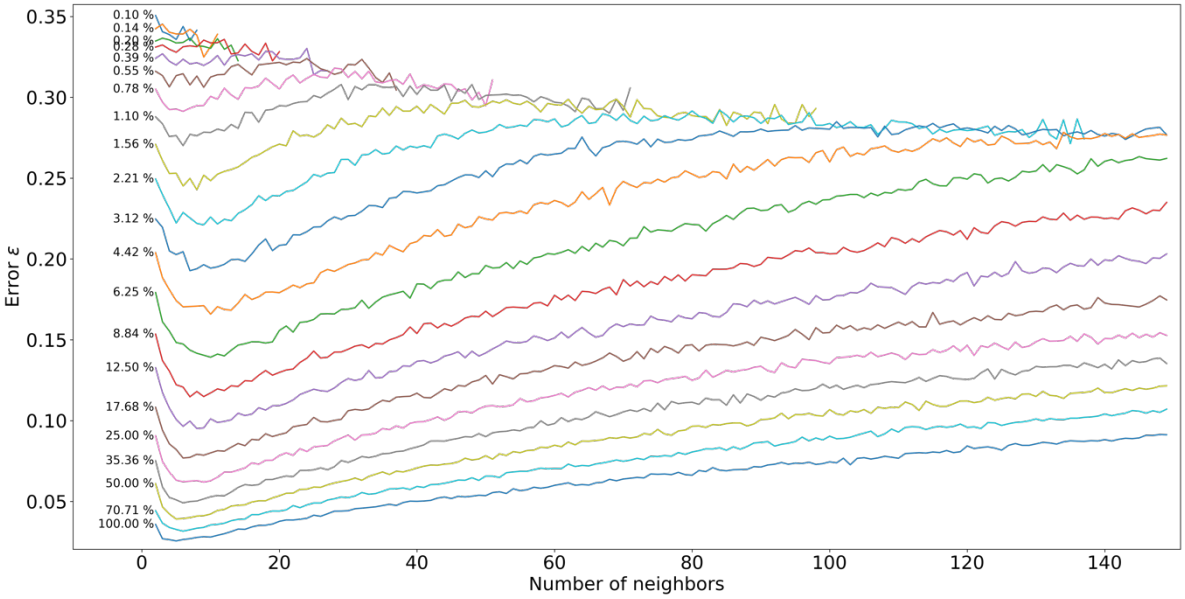

Figure 4.3 Error evolution as a function of the number of neighbors, with k=1, each curve represents the associated density of the neighborhood (which is equivalent to the fraction of the simulation path).

Figure 4.3 shows the adverse effect of taking into account too many neighbors, which results in including non-informative neighbors. It is possible to visualize (for the few first percents (from 0.55% to 1.56%)) the two regimes that exist in the simulation. The first minimum, where the algorithm determines a candidate based on the direct neighbors, that is optimal for the continuity of the surface (short range), the second minimum is optimal for the prediction of structures (long range). This figure shows that the first percent simulated can reach the optimal prediction even with a small number of neighbors. However, it rapidly starts to be impossible, and the importance of a continuous surface takes over. This two-step simulation is expected, as random large-scale features are generated first, then in a second step the MPS algorithm fills the image with a consistent fine-scale structure.

4.3.1.2 Automatic calibration for QS taking into account different kernels

Here, we use the following configuration $\theta_h = \{n,\ \omega\}$ and $\theta_s = \{k\}$. We consider kernels having a radial exponential shape, $w_i = e^{\alpha.d_i}$.

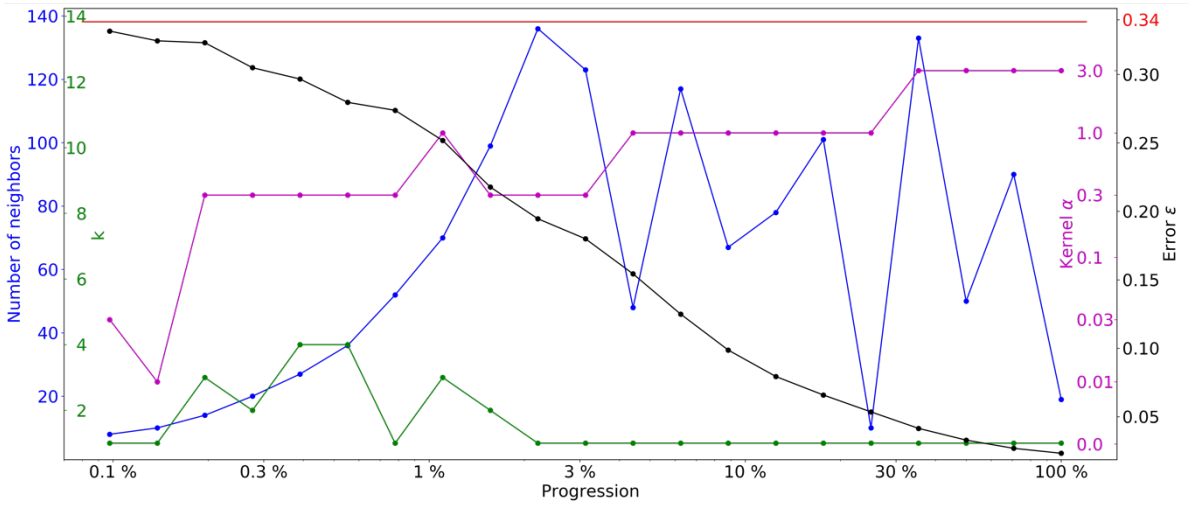

Figure 4.4 Optimal parameters for QS (k and number of neighbors, and the best kernel) In function of the progression, with the associate prediction error.

Figure 4.4 shows the evolution of the parameters of QS, where interferences between the number of neighbors and skewed kernels (high α) are visible. This can be explained by the fact that the last neighbors will receive negligible weights. Therefore, we propose the difference between each configuration is negligible and is more due to random noise on the metric, this effect can be visualized in the graphics in annexes. As expressed in the metrology section, in cases of a similar error, the cheapest solution is considered. In case of QS, lots of neighbors can reduce the efficiency of the path parallelization; therefore, we propose ad hoc solutions that stabilize the result of introducing a small tolerance that results in favoring small n.

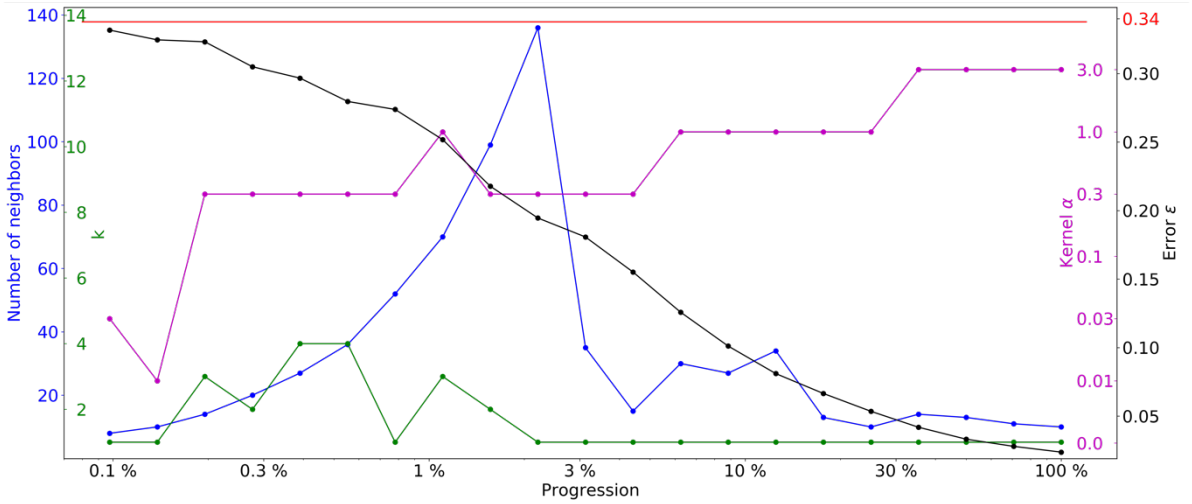

Figure 4.5 Optimal parameters for QS (k and number of neighbors, and the best kernel) using minimum coast approach In function of the progression, with the associate prediction error.

Figure 4.5 shows similar quality (ε curves) as Figure 4.4. However, it reduces drastically the number of neighbors required during the simulation. Compared to Figure 4.2 it is clear that the utilization of kernels reduces the prediction error (ε curves) and therefore should increase the simulation quality.

4.3.2 Automatic calibration for DS

In the case of DS, we determine the optimal values for n the number of neighbors, f the scanned fraction of the training image, and t the threshold. We use the following configuration: $\theta_h = \{n\}$ and $\theta_s = \{t, f\}$.

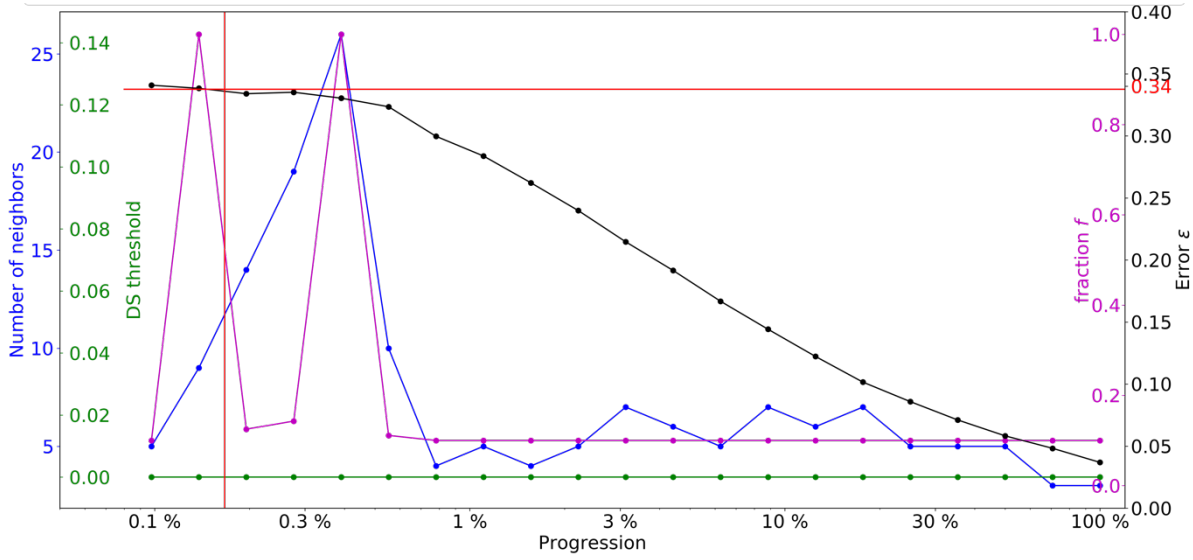

Figure 4.6 Optimal parameters (threshold and number of neighbors) in function of the progression, with the associate prediction error.

Results are shown in Figure 4.6. It is interesting to see that the threshold systematically drops to zero. Here we see the same pattern as for QS, where we start increasing neighbors and abruptly change of regime, and take only a few neighbors into account. Furthermore, similarly as for QS, the early pixels simulated are unpredictable and therefore can just be drawn from the marginal distribution. Here around 0.15% can be drawn without taking into account the neighborhood, which means the first 60 pixels of a 200 x 200 pixels simulation.

4.3.3 Sequential simulation using automatic calibration

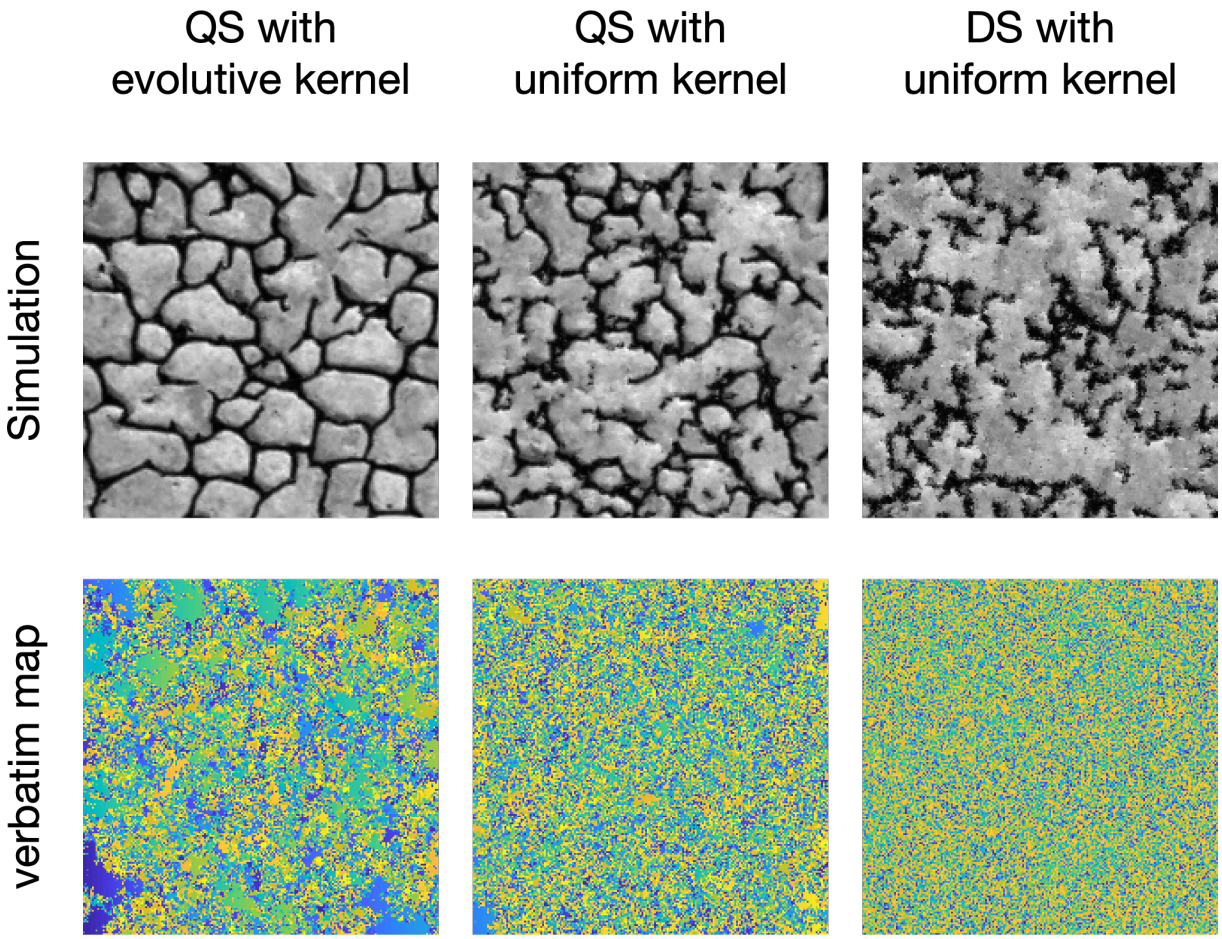

Figure 4.7 Simulation using QS and DS using parameters generated by the automatic calibration .Single realization and the associated Verbatim map.

Figure 4.7 shows qualitative results using the proposed autocalibration. The QS with evolutive kernel refers to the use of different values of alpha for the kernel. In this case the results are similar to state-of-the-art simulations using a manual calibration. Both tests with QS and DS using a uniform kernel have similar results (DS results are noisier), and both have missed some structures in particular the size of the features. Each verbatim map shows very few homogeneous areas; therefore, realizations are produced with a very low rate of verbatim copy.

From a quantitative point of view Figure 4.8 illustrates different metrics over a set of 100 simulations. In regards to Chapter 2 (Figure 2.6), the automatic calibration method proposed here, reaches a similar quality, especially for QS. It is interesting to see that the variogram and connectivity metric are in an acceptable range without having been constrained in the calibration process. The Euler characteristic seems to be highly

discriminative between QS and DS simulations. This can be explained by the sensibility of the Euler characteristic in regards to small-scale noise.

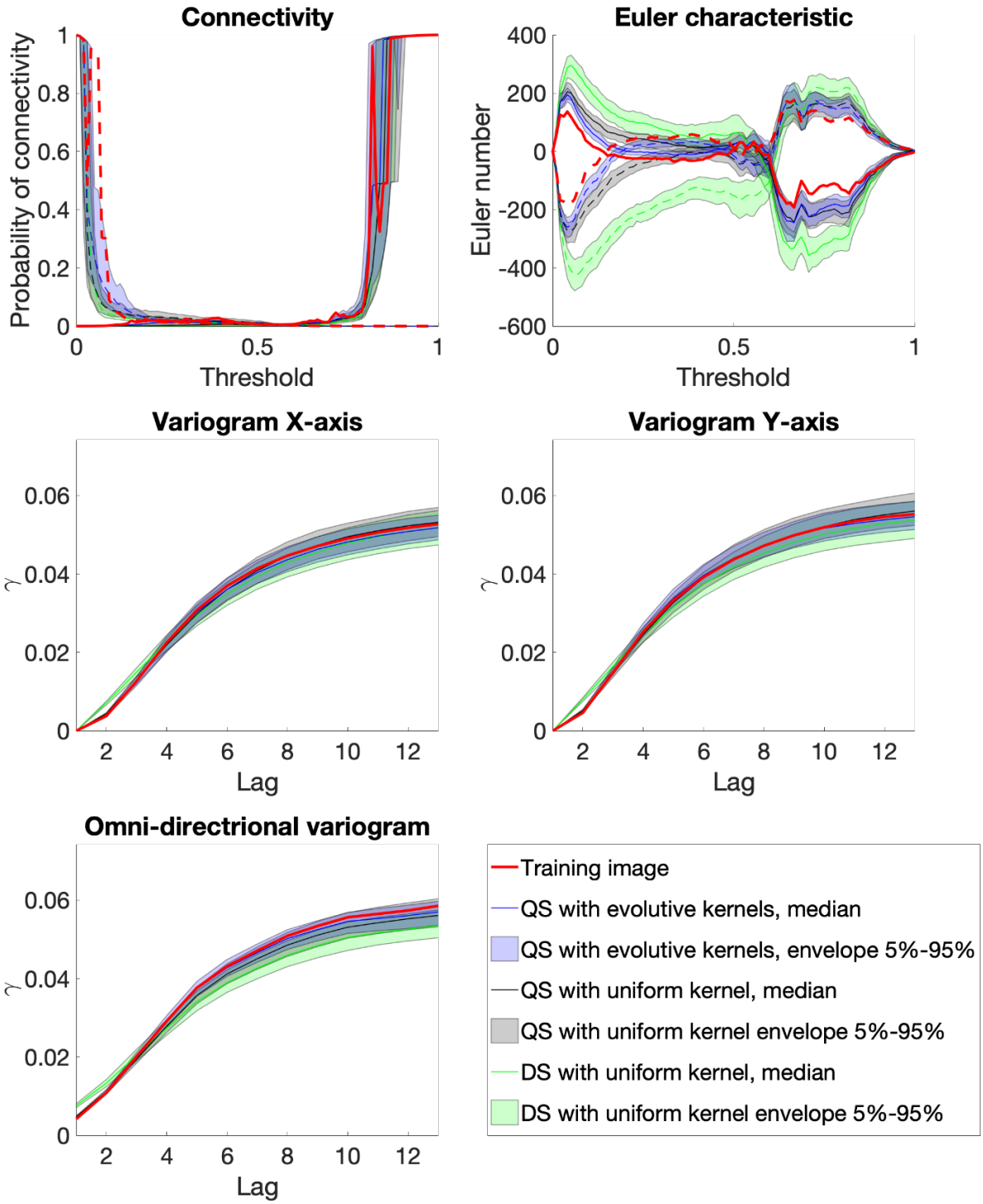

Figure 4.8 Benchmark between QS (with evolutive and uniform kernel) and DS(uniform kernel), over 100 simulations for 5 different metrics.

4.4 Discussion

The proposed method allows automatic calibration of pixel-based MPS, reaching similar quality as manual parameterization from both quantitative and qualitative points of view.

If the metrics confirm the quality of the simulation, the images tend to be less visually pleasing. This can be explained by the fact that this method finds a calibration that avoids verbatim copy.

The major advantage is the absence of a complex objective function, which often requires calibration by itself. The method runs for a predictable maximum time, time that depends on the quality of the calibration that we expect. The calibration can even be refined building on previous results, without running all the processes again.

The current implementation with Python is slow (30-40 min), but we estimate that a C/C++ implementation of the calibration framework would be able to achieve the computation in less than 10 min.

The proposed method relies on the fact that under optimal parameterization the resulting search pattern has a unique solution. This property always looks true for practical applications, but we can suppose that some specific artificial training images, will not guarantee such a property (we searched to construct such a training image, but did not find any examples). The exposed method focuses on the simulation of a single pixel, therefore, resulting in the impossibility of using it to determine the optimal path. Furthermore, the method does not take external considerations such as the computation cost required for a simulation.

4.5 Conclusion

This proposed approach allows us to automatically calibrate pixel-based-MPS. Furthermore, it demonstrates that for optimal results the parametrization cannot remain constant during the simulation, and needs to evolve with the simulation progression. A visually appealing result of complex features without verbatim-copy is hard to simulate, especially when using a uniform kernel. This proposed method allows for calibrating over a set of predetermined kernels; however, we can imagine in a future work, optimizing the kernel for each individual weight w_i using ε as objective function.

The study of the evolution of parameters shows a smooth behavior of the average error. Therefore, the use of a multi-variable fitting approach to estimate the error surface with fewer evaluations could be an interesting solution to speed up the parametrization, by capitalizing on neighbors (in parameter space). The use of machine learning to take advantage of transfer-learn between training images has a high potential too. These two solutions will allow interpolating between parameters like α of the kernel.

4.6 Author contribution

MG proposed the idea, implemented, experiment and wrote the manuscript. GM provided supervision, methodological insights and contributed to the writing of the manuscript.

4.7 Acknowledgments

This research was funded by the Swiss National Science Foundation, grant number 200021_162882. Thanks to Ehsan Baninajar for brainstorming, and to Raphaël Nussbaumer for his comments, and insight in result interpretations, which greatly improved the manuscript.

5 Thesis conclusion

The main objective of this thesis was to develop an efficient framework to realize spectral enhancements. To this end, the second chapter is about developing an efficient MPS algorithm and pattern matching procedure that easily and efficiently allowing a conditional distribution. The third chapter is about embedding this framework in a more general procedure to get high quality spectral enhancements, and to compare it to state-of-the-art approaches. Finally, the chapter 4 is about how to automatically parametrize these methods, such that a non-expert can take advantage of them.

5.1 Contribution to the field

One of the main contributions of this thesis demonstrating the interest and shows the tremendous potential for extensive use of archive imagery.

The main algorithmic contributions of this thesis are the design of QS (chapter 2), which allows realizing MPS simulations quickly and easily, and the NDS (chapter 3) that uses a self-determined path to produce improved MPS simulations in case of fully informed covariables. While methods to calibrate MPS algorithms already exist, the proposed solution in chapter 4 is by design simple and straightforward to use, without the need to design a complex objective function.

From a more practical point of view, an important contribution of this thesis to the field is the high-performance implementation of QS, NDS and the associate interface available for MATLAB, Python and R. This will allow the scientific community to use the same implementation and use it for a wide range of applications.

5.2 Perspectives

The focus of this thesis was to develop a framework to realize spectral downscaling using multiple point statistics. To this end, several methods were designed and implemented, including NDS which proposes a self-determined path to improve simulations. I think this idea should be further explored, including the case of simulating continuous variables without fully informed covariables. Chapter 3 exposes a solution to choose and adapt the kernel as the simulation progression; however, these kernels are still parametric. Even, if the interest of kernels to improve simulation quality is clear, finding the best one remains a challenging task.

I can imagine that the next step is to forge a kernel that is adapted to the training image, not with an optimization approach but with a deterministic algorithm. An alternative could be to generate the kernel on the fly during the simulation.

The number of existing MPS algorithms is considerable, and a new algorithm should be only developed for specific tasks where we can demonstrate a true gain (computational, quality, or simplicity).

There is limited interest for new algorithms to simulate binary channeled structures. Today, MPS methods face new challenges with more complex images than ever before. For example, it would be useful to develop algorithms where multiple variables are not weighted, but only ranked in order of importance, which would be a more robust approach.

During the last few years, MPS has moved forward to unsolved issues like optimal parametrization, or simply how to find the most adapted training image. For this last question, I see great opportunities coming from remote sensing applications, where enough training data is available, and where usually some previous information can guide the selection. The geostatistics field, after switching from two-point approaches to multiple points algorithms, is now slowly moving forward to machine learning. The evolution toward machine learning is inevitable, and will require tremendous work to adapt this solution to perform simulations too and not only estimation as is currently the case.

The colorization solution provided in this thesis shows that some wavelengths carry almost no information. While RGB images are highly used to visualize the information, the amount of information they contain compared to a panchromatic band, is small regarding other bands such as NIR. For numerous applications, having a panchromatic and a NIR band would be much more informative than having RGB. In chapter 3 it was shown that it is possible to convert information from textures to existing spectral bands. The complete spectral information is for sure not included in the textural information, and therefore an exact spectrum cannot be perfectly reconstructed based on it. However, the opposite is true too, the spectrum doesn’t contain all the information included in the texture; therefore, the construction of artificial bands based on this texture to carry extra information that is not already included in the spectrum should be considered.

A. Appendixes

A.1 Threshold vs. Quantile

DS was developed based on the idea that any candidate under a given threshold is an acceptable candidate. At the opposite, QS was developed based on the idea that any candidate within a distance quantile is acceptable. To explore the question of the most appropriated choice, we can examine the mismatch error as a function of the ranking over the evolution of a simulation.

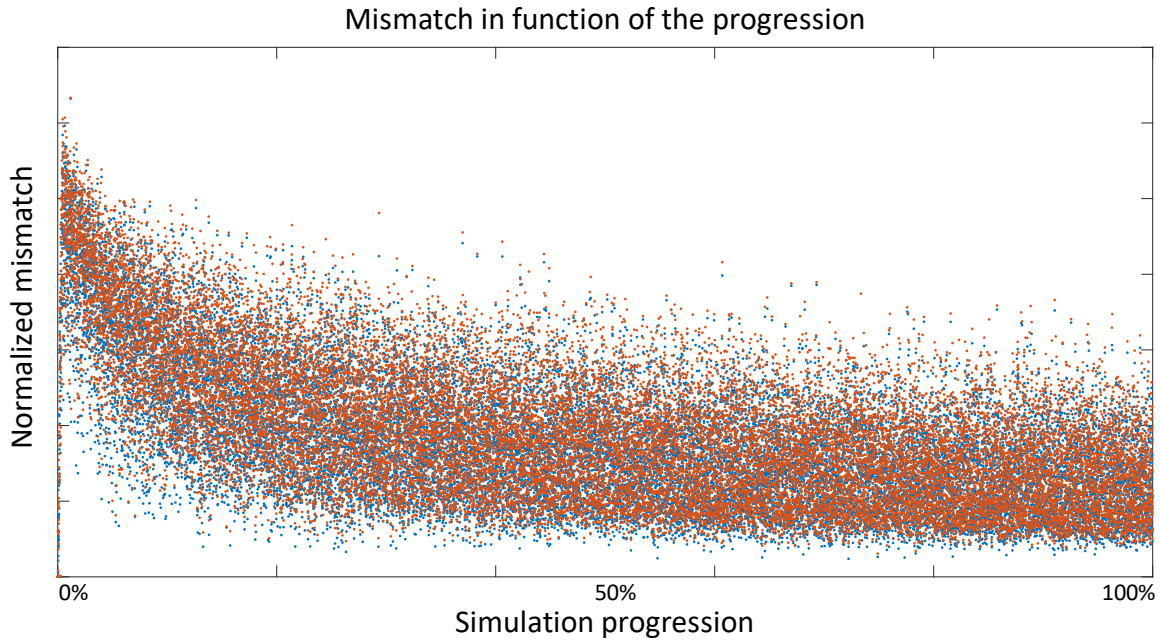

Figure A.1 The 3 lowest mismatches at each step of a simulation, using a uniform kernel. In bleu the lowest and in red the second and third lowest mismatch.

Figure A.1 shows that the mismatch of each pattern globally decreases with the evolution of the simulation. However, the signal is extremely noisy, and it is hard to find an acceptable threshold that is valid for the entire simulation.

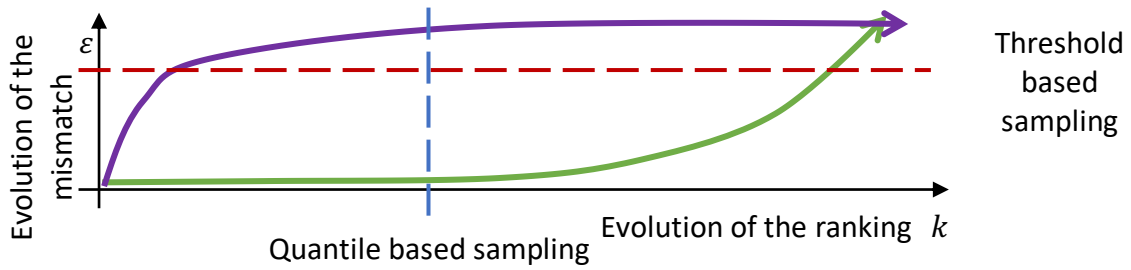

Figure A.2 Duality between threshold and quantile sampling. Here the evolution of two typical cases represented in violet and green.

Another way to visualize the duality between threshold and quantile is to explore the difference at the level of a single pixel simulation. Figure A.2 shows the evolution of the mismatch regarding the ranking. This shows by deduction that a perfect solution doesn't exist that allows one to choose between both strategies. In fact, both strategies seem to be complementary. Experimentation shows that a quantile approach is a safer solution (in terms of time and quality). When threshold sampling can provide a major gain in some easy situations (quick and good simulations), there is always a risk of high computation time (doesn't finding an element under the threshold) or poor resulting quality (threshold to high).

A.2 Implementation challenges

Developing (QS + NDS) and implementing (DS + QS + NDS) MPS algorithms, I faced numerous challenges that are undiscussed in many papers in the field of MPS. The most significant challenges are probably how to handle the missing values in the training image, border effects, and which simulation path to choose.

A.2.1 Missing neighbors in the TI

Computing the mismatch between a pattern and a reference is an easy task.

$$\mathrm{E}\big(N(t), N(s)\big) = \sum_{\boldsymbol{l} \mid N_{\boldsymbol{l}}(t) \; and \; N_{\boldsymbol{l}}(s) \; exist} \omega_{\boldsymbol{l}} \varepsilon\big(N_{\boldsymbol{l}}(t), N_{\boldsymbol{l}}(s)\big) \qquad (A.1)$$

Chapter 2 Equation 2.5, we simply do the sum of all the errors of each pixel that is informed. This works independently of the number of pixels informed. If the training image is complete and we only consider the central part of the training image, then the number of pixel-wise errors is constant. If we consider a training image with missing values, or if we consider incomplete patterns on the border of the image, then the number of pixel-wise-errors fluctuates, and makes any direct sorting (ranking) impossible. Considering this question different possibilities were tried.

- The normalization by the sum of kernel weights (or the number of neighbors in cases of a uniform kernel), seems the most logical and intuitive solution. However, experimentation shows that this is far from optimal. The influence of one neighbor over a pattern of 10 or 100 pixels is significantly different and far from being a linear challenge. (e.g., If we considered a single neighbor, we will always find a perfect match, when considering two neighbors, we can end with no perfect solutions.) Finally, this

approach tends to result in sampling patterns with few pixel-wise-errors, and therefore is not well constrained, resulting in degraded simulations. (e.g. the perfect match with a single value would be always be sampled.)

- Taking into account the bias, it is possible to add extra constraints over the previous situation like a minimum number of informed pixels. This limits but doesn't solve the problem of incomplete patterns. It is possible to compensate the normalization by favoring slightly the candidates with more nodes informed. Currently, QS use a polynomial of degree 4 that tend to be similar to linear for situations with lots of neighbors informed, and to ban the pattern with very few neighbors informed.
- An interesting solution is to add a constant cost for all missing pixels. We can estimate the optimal cost considering the marginal distribution. However, this solution is only optimal in average and results in overestimating the mismatch for simple patterns (→ under sample incomplete patterns) and underestimating the mismatch for complex patterns(→ over sample incomplete patterns).
- It is possible to fill the gaps in the training image (only for mismatch measure), using an interpolator (e.g., for smooth variable), or MPS to simulate gaps (e.g., for chessboard style variable). This solution can provide good results if we know with interpolator to use. However, the generalization is complicated, and the results are unpredictable.
- Finally, the easiest solution is maybe to simply ignore the incomplete patterns. However, with 10% of the training image missing, no (or only extremely bad) patterns are available.

All the previous experimentations were done using controlled gaps that are uniformly distributed. If no solution shows real advantages, the management of non-stationary distribution of missing areas is even more challenging. Recent work (Chen et al., 2019) shows the need for taking into account the layout of a pattern (e.g., cluster of informed pixels). But, still no clear solution is available. This point leads to an interesting research question, with the potential to significantly increase the useable part (including borders) of the training image (which is usually already too small).

A.2.2 Path

The simulation path used can have a tremendous influence on the simulation quality. This is specific to each algorithm: usually SNESIM/Impala give best results with a multigrid path to improve memory management, whereas most patch-based approaches use a row path to allow patch merging. Direct sampling usually uses a random path. As with Sequential Gaussian Simulation, each path type has its own strengths and weaknesses. Using Nusbaumer et al. (2018) terminology, we have:

- Row-by-row: (aka. unilateral) is a path that can be used if not too much conditioning data is available. It tends to respect exceptionally well structures like connectivity; however, the size of the structures tends to be wrong, and parallelization of the path is complicate.

Spiral: this path is usually discouraged because algorithms tend to have difficulty to simulating long-range structures. However, Hoffiman (2017) shows an interesting potential in cases where a secondary variable is available to guide simulations. If enough conditioning points are available, parallelization is possible.

- Multi-grid (and derivatives): This path is often highlighted for its qualities, and is relatively easy to parallelize. It simulates first the large structures, then the smaller ones. Therefore, simulations are usually of relatively good quality. However, conditioning points needs to be migrated at each multi-grid level, which can produce localized inconsistencies.
- Random: The random path is the path that uses no particular assumption. Usually considered as the paths with less bias. It is relatively easy to parallelize.
- Self-determined: this class of path is still poorly used, but is still an interesting. This type of path we don't need to choose a path anymore. The idea is that the algorithm finds the adapted path by itself during the simulation. The first apparition of a self-determined path in MPS was by Liu and Journel(2004), based on the idea of visiting the most determined (lowest entropy) locations first. Using a similar idea, NDS simulated the most certain locations first. Recently, Hansen (2019) proposed using the opposite strategy and simulated underdetermined (high entropy) locations first, in order to reduce the cumulated entropy as quickly as possible and converge to

extremely constrained simulations. However, self-determined simulation paths can potentially reduce the variability over a set of simulations.

A.3 Variability of a set of simulations

In this section I will focus on the question of the variability of realization. To realize the proper estimation at the output, the set of simulations used in input needs to be representative of the different possible senarios. With MPS simulations, the algorithmic parameters can influence the variability of simulations. If the variability of scenarios is hard to measure, the variability at a given location is easy to estimate, and is usually a good proxy. Here we will consider different parametrization cases.

- Extremely constrained (QS → low k; DS → low threshold and high fraction), each solution is unique and variability is only provided through the different path.
- Less constrained (QS → small k (1<k<10); DS →a small threshold and fraction (~20%)), will allow more variability because each pixel is simulated with less constrains.
- Low constraints (QS → high k, DS → a high threshold and minimal fraction), each pixel is simulated almost independently, and will converge to the variability of the marginal distribution.

However, experimentation shows that the variability evolution doesn't follow the expected shape (Figure A.3). The variability starts to decrease immediately with the release of constrains.

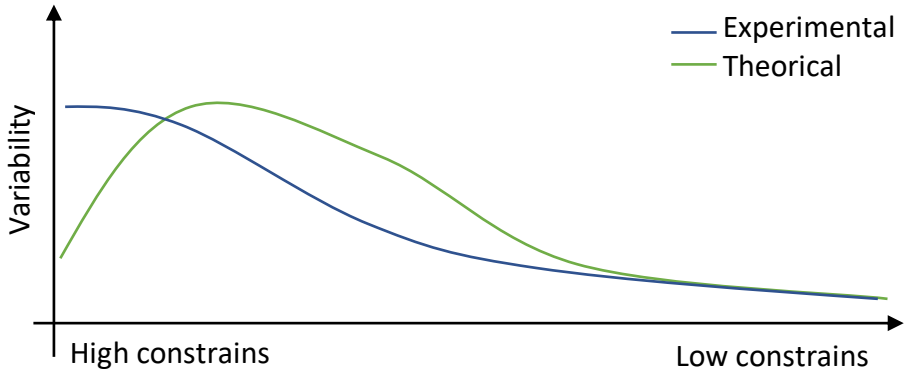

Figure A.3 Schematic evolution of the variability in function of the constrains in the pattern selection, extrapolated over the variation of variability regarding k

That tends to put all the weight of the variability in the selection of the path. One example of this phenomenon is (Figure A.4) when the increase of k (relaxes of constrains), reduces systematically the variability.

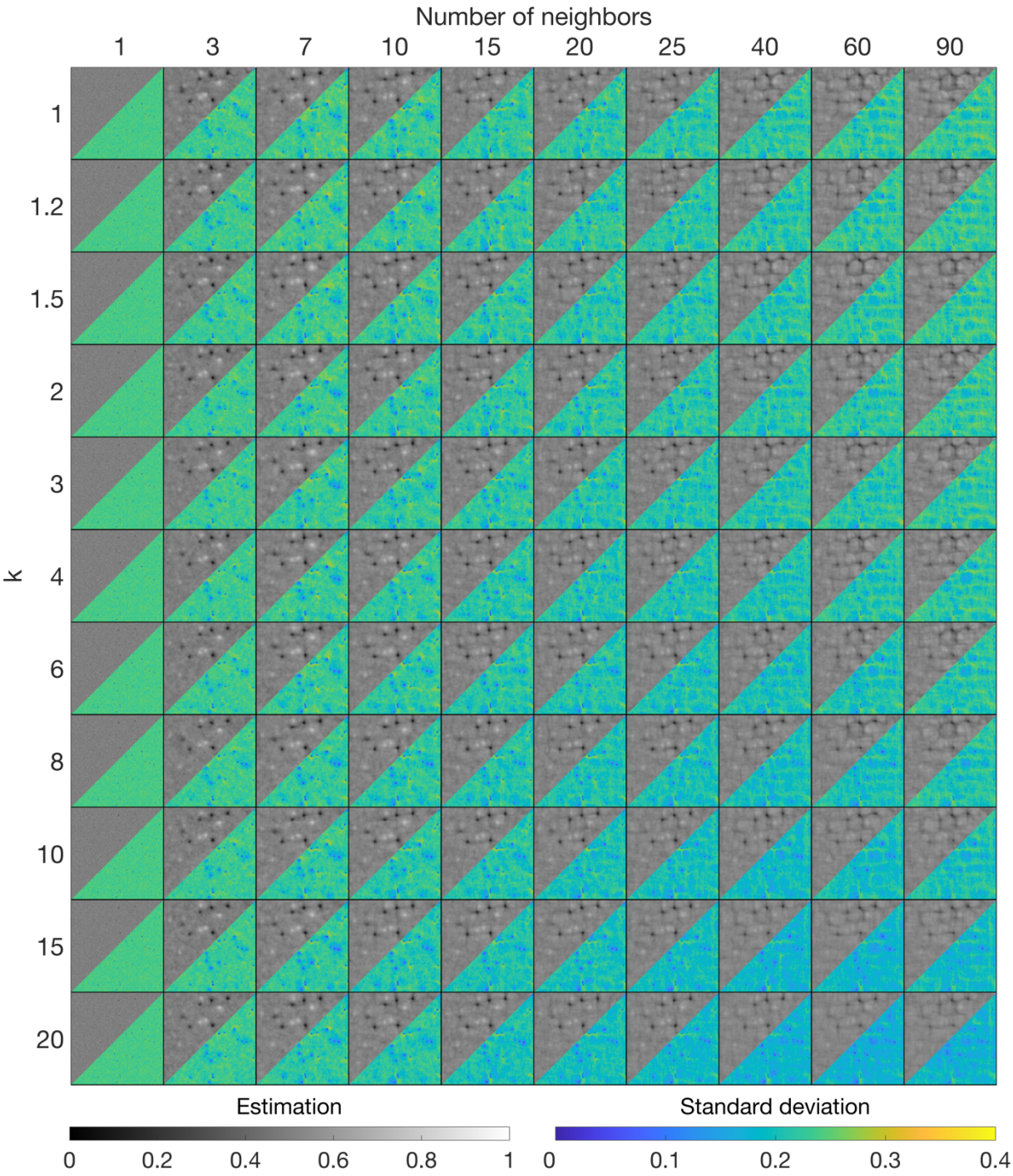

Figure A.4 Sensitivity analysis of the variability over a set of 100 realizations for the two main parameters of QS. For simulation refer to Figure A.4.3. The variability decreases with the increases of k.

A.4 Constructing a kernel based on optimization

Before developing the solution of the Chapter 4, I performed parameters optimization tests using approaches based on the minimization of objective functions. The goal of this work was to find a unique and optimal setup for the complete simulation. The framework was developed using genetic algorithm, which allows a simple parallelization of each iteration (each candidate of the population can be simulated independently). This project was

interrupted, due to the extreme computation required, that is hard to justify in regard to the improvement, and the difficulty of control over fitting, to the condition as well as to the objective function.

The objective function was composed of variogram errors (RMSE between variograms), connectivity errors (dynamic time warping (DTW) between connectivity functions) or SSIM. First a complete but coarse (in parameter space) exploration was done to explore variation regarding kernel parameters. Figure A.5 show the result in terms of metrics and simulations. The optimal for moth metrics is far from being visually pleasant. The optimization in case of a single variable shows (Figure A.6) its potential in case with better structured simulations with the same error. However, it is hard to know when to interrupt the optimization. Figure A.7 shows the risk of an over fitting of the objective function; here the simulation looks comparable to a Gaussian simulation, for with the variogram is an adapted metric. Figure A.8 shows the potential of a multivariate with fully informed covariables. In this last case, the results are much more interesting, showing promising outcomes, especially we demonstrate the complex structure of an optimal kernel. However, the computation cost to get acceptable parameter is extremely high, and justifies a change of approach. The potential next step is to merge the framework of chapter 3 with an optimization approach to get similar result at lower cost.

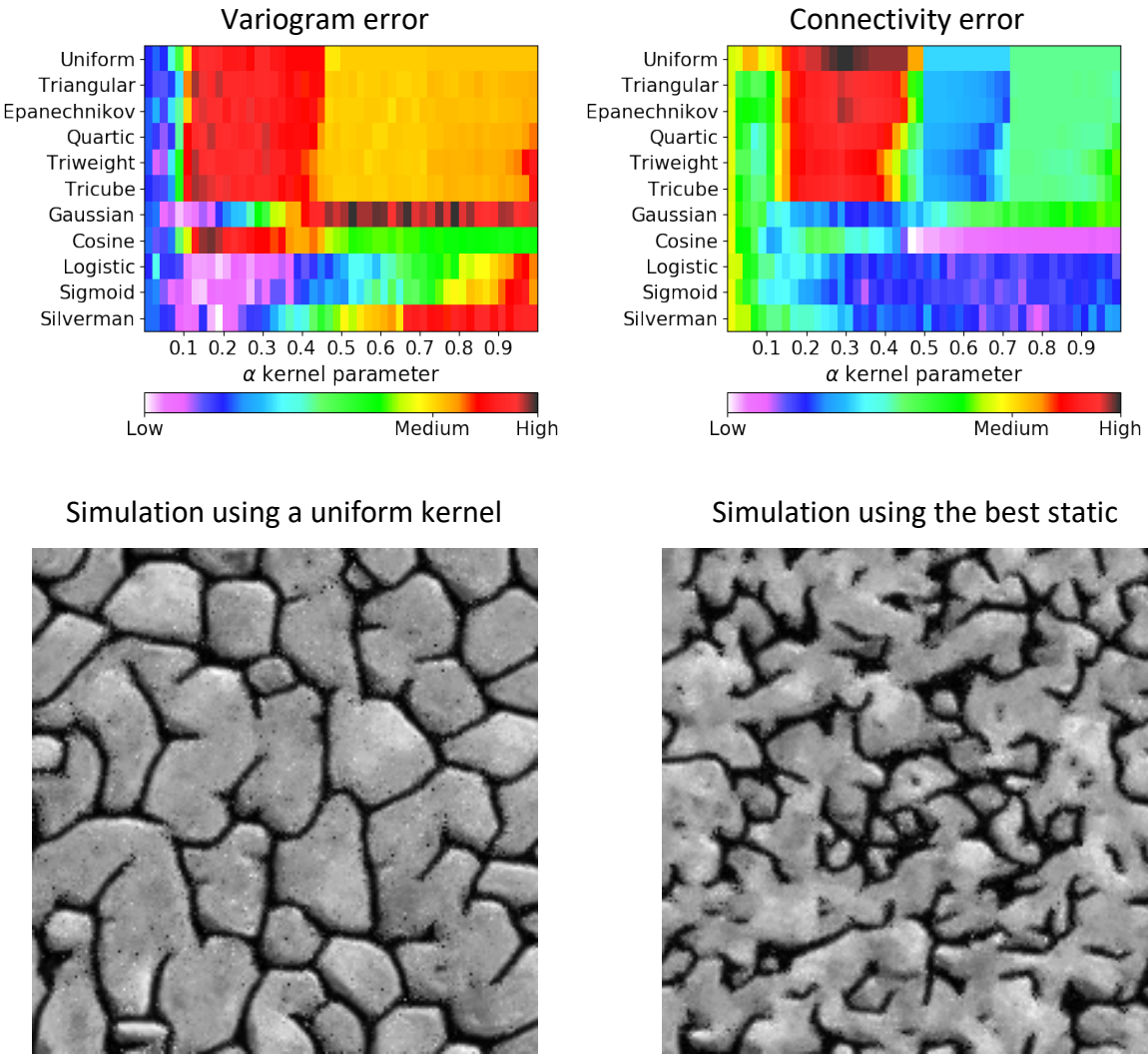

Figure A.5 Variogram and connectivity error of complete simulations, regarding the static kernel and α parameter associated. The best static kernel refers to an optimal setting in theme of variogram and connectivity metric.

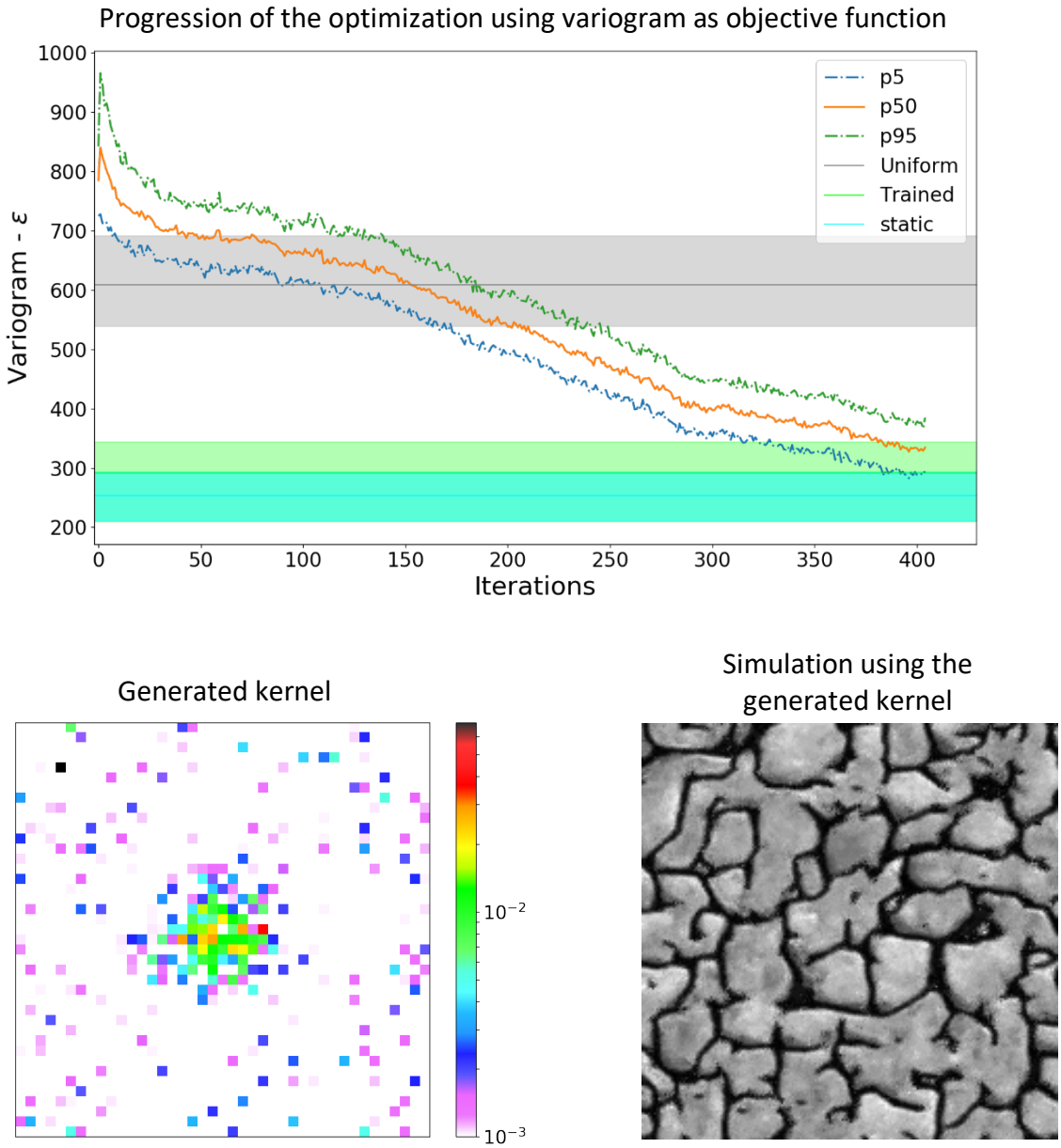

Figure A.6 Result of kernel optimization after 500 iteration (~1.5 days using 5 computer)

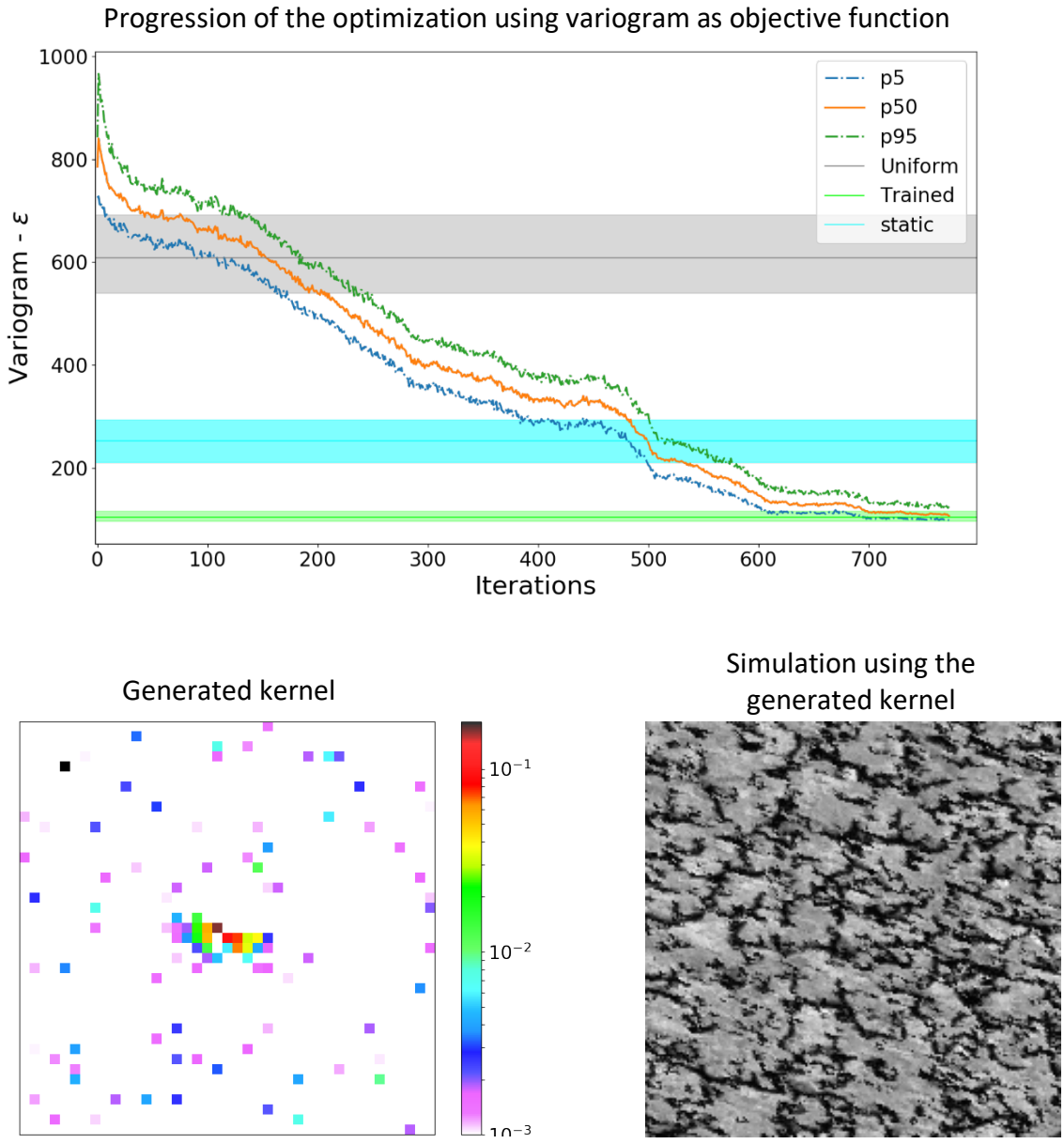

Figure A.7 Result of kernel optimization after 800 iteration (Overfitting the objective function) (~2.5 days using 5 computer)

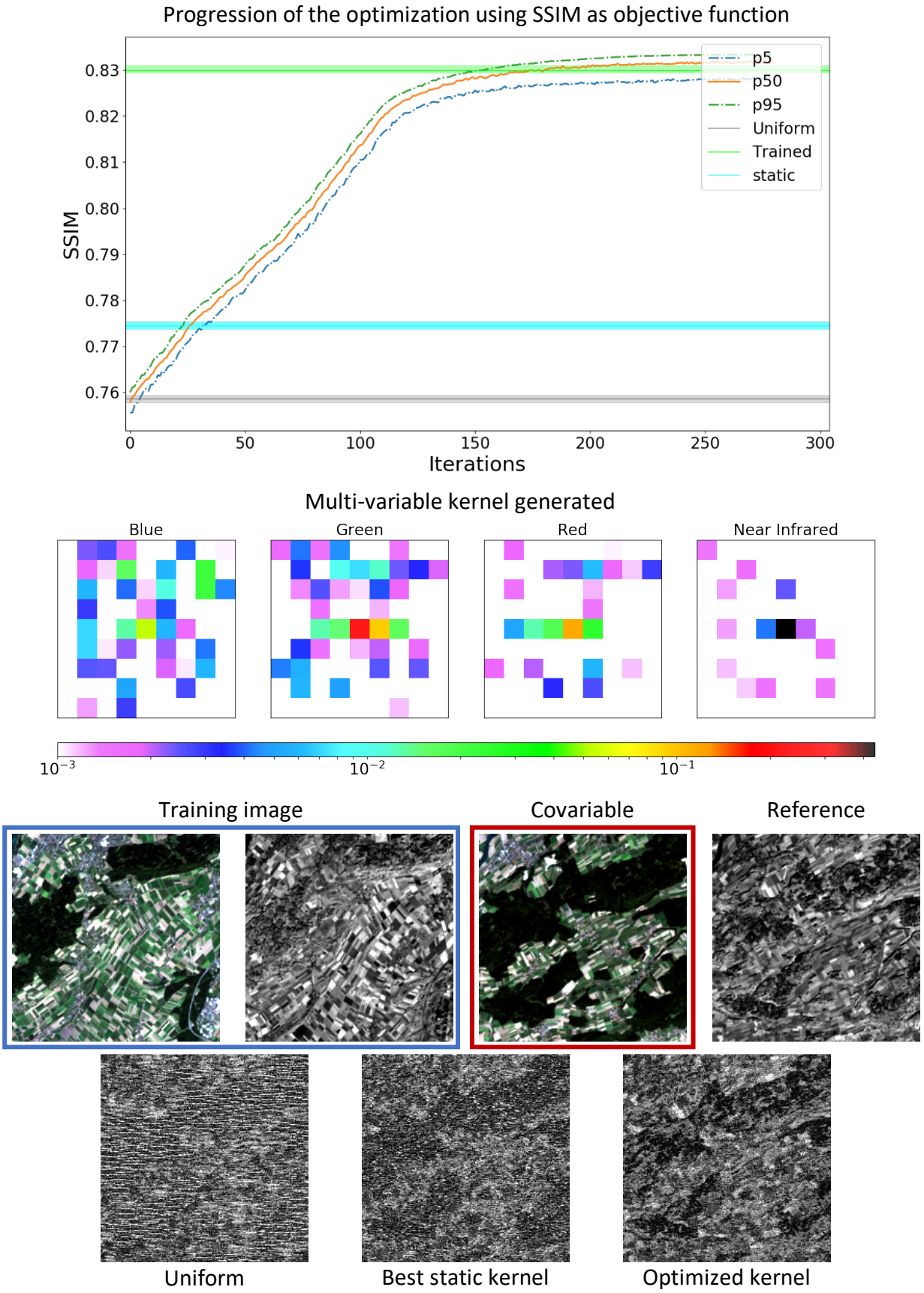

Figure A.8 Result of multi variable kernel optimization after 280 iteration (~1 days using 4 computer). Simulating NIR knowing RGB.

Bibliography

Alphan, H., Yilmaz, K.T., 2005. Monitoring environmental changes in the Mediterranean coastal landscape: the case of Cukurova, Turkey. Environ Manage 35, 607–619. doi:10.1007/s00267-004-0222-7

Arpat, G.B., Caers, J., 2007. Conditional Simulation with Patterns. Mathematical Geology 39, 177–203. doi:10.1007/s11004-006-9075-3

Bancheri, M., Serafin, F., Bottazzi, M., Abera, W., Formetta, G., Rigon, R., 2018. The design, deployment, and testing of kriging models in GEOframe with SIK-0.9.8. Geosci. Model Dev. 11, 2189–2207. doi:10.5194/gmd-11-2189-2018

Baninajar E., Sharghi Y.,Mariethoz G.: MPS-APO: A Rapid and Automatic Parameter Optimizer for Multiple-point Geostatistics, accepted in Stochastic Environmental Research and Risk Assessment, Manuscript Number: SERR-D-19-00106

Barfod, A.A.S., Vilhelmsen, T.N., Jørgensen, F., Christiansen, A.V., Høyer, A.-S., Straubhaar, J., Møller, I., 2018. Contributions to uncertainty related to hydrostratigraphic modeling using multiple-point statistics. Hydrol. Earth Syst. Sci. 22, 5485–5508. doi:10.5194/hess-22-5485-2018

Beck, R., 2010. EO-1 User Guide v. 2.3, 2003. doi:10.1016/j.rse.2004.11.012

Boisvert, J.B., Pyrcz, M.J., Deutsch, C.V., 2010. Multiple Point Metrics to Assess Categorical Variable Models. Nat Resour Res 19, 165–175. doi:10.1007/s11053-010-9120-2

Borgman, L., Taheri, M., Hagan, R., 1984. Three-Dimensional, Frequency-Domain Simulations of Geological Variables, in: Geostatistics for Natural Resources Characterization. Springer, Dordrecht, Dordrecht, pp. 517–541. doi:10.1007/978-94-009-3699-7_30

Boucher, A., 2009. Sub-pixel Mapping of Coarse Satellite Remote Sensing Images with Stochastic Simulations from Training Images. Math Geosci 41, 265–290. doi:10.1007/s11004-009-9215-7

Bruzzone, L., Demir, B., 2014. A Review of Modern Approaches to Classification of Remote Sensing Data, in: Manakos, I., Braun, M. (Eds.), Land Use and Land Cover Mapping in Europe, Remote Sensing and Digital Image Processing. Springer Netherlands, Dordrecht, pp. 127–143. doi:10.1007/978-94-007-7969-3_9

Campbell, J.B., Wynne, R.H., 2011. Introduction to Remote Sensing, Fifth Edition. Guilford Press.

Charpiat, G., Bezrukov, I., Hofmann, M., 2009. Machine Learning Methods for Automatic Image Colorization 1–20. doi:10.1201/b10284-16

Cheng, G., Han, J., 2016. A survey on object detection in optical remote sensing images. ISPRS Journal of Photogrammetry and Remote Sensing 117, 11–28. doi:10.1016/j.isprsjprs.2016.03.014

Cohen, W.B., Spies, T.A., Alig, R.J., Oetter, D.R., Maiersperger, T.K., Fiorella, M., 2002. Characterizing 23 Years (1972-95) of Stand Replacement Disturbance in Western Oregon Forests with Landsat Imagery. Ecosystems 5, 122–137. doi:10.2307/3659042?ref=search-gateway:e1903d9c988897adcb86d25852b606a6

Cooley, J. W., & Tukey, J. W., 1965. An algorithm for the machine calculation of complex Fourier series. Mathematics of Computation, 19(90), 297–297. https://doi.org/10.1090/s0025-5718-1965-0178586-1

Cooley, J.W., computation, J.T.M.O., 1965, 1965. An algorithm for the machine calculation of complex Fourier series. JSTOR

Dagasan, Y., Renard, P., Straubhaar, J., Erten, O., Topal, E., 2018. Automatic Parameter Tuning of Multiple-Point Statistical Simulations for Lateritic Bauxite Deposits. Minerals 8, 220–20. doi:10.3390/min8050220

Davis, M.W., 1987. Production of conditional simulations via the LU triangular decomposition of the covariance matrix. Mathematical Geology 19, 91–98. doi:10.1007/BF00898189

Dewidar, K.M., 2002. Landfill detection in Hurghada, North Red Sea, Egypt, using Thematic Mapper images. International Journal of Remote Sensing 23, 939–948. doi:10.1080/01431160110070645

Dimitrakopoulos, R., Luo, X., 2004. Generalized Sequential Gaussian Simulation on Group Size ν and Screen-Effect Approximations for Large Field Simulations. Mathematical Geology 36, 567–591. doi:10.1023/B:MATG.0000037737.11615.df

Fraser, R.H., Olthof, I., Carrière, M., 2011. Detecting long-term changes to vegetation in northern Canada using the Landsat satellite image archive. Environmental Research Letters 6, 045502. doi:10.1088/1748-9326/6/4/045502

Frigo, M., Johnson, S.G., 2018. FFTW.

Gauss, C.F., 1799. Demonstratio nova theorematis omnem functionem algebraicam.

Ge, Y., 2013. Sub-pixel land-cover mapping with improved fraction images upon multiple-point simulation. International Journal of Applied Earth Observation and Geoinformation 22, 115–126. doi:10.1016/j.jag.2012.04.013

Ge, Y., Bai, H., 2011. Multiple-point simulation-based method for extraction of objects with spatial structure from remotely sensed imagery. International Journal of Remote Sensing 32, 2311–2335. doi:10.1080/01431161003698278

Gnyp, M.L., Miao, Y., Yuan, F., Ustin, S.L., Yu, K., Yao, Y., Huang, S., Bareth, G., 2014. Hyperspectral canopy sensing of paddy rice aboveground biomass at different growth stages. Field Crops Research 155, 42–55. doi:10.1016/j.fcr.2013.09.023

Graeler, B., Pebesma, E., Heuvelink, G., 2016. Spatio-Temporal Interpolation using gstat. R Journal 8, 204–218.

Gravey, M., Rasera, L.G., Mariethoz, G., 2019. Analogue-based colorization of remote sensing images using textural information. ISPRS Journal of Photogrammetry and Remote Sensing 147, 242–254. doi:10.1016/j.isprsjprs.2018.11.003

Guardiano, F.B., Srivastava, R.M., 1993. Multivariate Geostatistics: Beyond Bivariate Moments, in: Geostatistics Tróia '92, Quantitative Geology and Geostatistics. Springer, Dordrecht, Dordrecht, pp. 133–144. doi:10.1007/978-94-011-1739-5_12

Gómez-Hernández, J.J., Journel, A.G., 1993. Joint Sequential Simulation of MultiGaussian Fields, in: Geostatistics Tróia '92, Quantitative Geology and Geostatistics. Springer, Dordrecht, Dordrecht, pp. 85–94. doi:10.1007/978-94-011-1739-5_8

Hamming, R.W., 1950. Error detecting and error correcting codes. The Bell system technical 29, 147–160. doi:10.1002/j.1538-7305.1950.tb00463.x

Hansen, M.C., Stehman, S.V., Potapov, P.V., Loveland, T.R., Townshend, J.R.G., DeFries, R.S., Pittman, K.W., Arunarwati, B., Stolle, F., Steininger, M.K., Carroll, M., Dimiceli, C., 2008. Humid tropical forest clearing from 2000 to 2005 quantified by using multitemporal and multiresolution remotely sensed data. Proc. Natl. Acad. Sci. U.S.A. 105, 9439–9444. doi:10.1073/pnas.0804042105

Hansen, T. M., 2019. Entropy of Multiple-point statistical models. In: IAMG 2019, State College, Pennsylvania, US.

Hildebrandt, A., 1907. Die Luftschiffahrt 1–446.

Hoang, N.T., Koike, K., 2017. Transformation of Landsat imagery into pseudo-hyperspectral imagery by a multiple regression-based model with application to metal deposit-related minerals mapping. ISPRS Journal of Photogrammetry and Remote Sensing 133, 157–173. doi:10.1016/j.isprsjprs.2017.09.016

Hoang, N.T., Koike, K., 2018. Comparison of hyperspectral transformation accuracies of multispectral Landsat TM, ETM , OLI and EO-1 ALI images for detecting minerals in a geothermal prospect area. ISPRS Journal of Photogrammetry and Remote Sensing 137, 15–28. doi:10.1016/j.isprsjprs.2018.01.007

Hoffimann, J., Scheidt, C., Barfod, A., & Caers, J., 2017. Stochastic simulation by image quilting of process-based geological models. Computers & Geosciences, 106, 18–32. https://doi.org/10.1016/j.cageo.2017.05.012

Hoffimann, J., Scheidt, C., Barfod, A., Caers, J., 2017, 2017. Stochastic simulation by image quilting of process-based geological models. Elsevier

Honarkhah, M., Caers, J., 2010. Stochastic Simulation of Patterns Using Distance-Based Pattern Modeling. Math Geosci 42, 487–517. doi:10.1007/s11004-010-9276-7

Houborg, R., McCabe, M., 2018. Daily Retrieval of NDVI and LAI at 3 m Resolution via the Fusion of CubeSat, Landsat, and MODIS Data. Remote Sensing 10, 890–23. doi:10.3390/rs10060890

Hwang, J., Zhou, Y., 2016. Image Colorization with Deep Convolutional Neural Networks.

Intel Corporation, 2019. Intel® Math Kernel Library Reference Manual - C 1–2606.

Jha, S.K., Mariethoz, G., Evans, J., McCabe, M.F., Sharma, A., 2015. A space and time scale-dependent nonlinear geostatistical approach for downscaling daily precipitation and temperature. Water Resources Research 51, 6244–6261. doi:10.1002/2014WR016729

John Paul Shen, M.H.L., 2018. Modern Processor Design: Fundamentals of Superscalar Processors 1–658.

Kessler, T.C., Comunian, A., Oriani, F., Renard, P., Nilsson, B., Klint, K.E., Bjerg, P.L., 2012. Modeling Fine-Scale Geological Heterogeneity-Examples of Sand Lenses in Tills. Groundwater 51, 692–705. doi:10.1111/j.1745-6584.2012.01015.x

Latombe, G., Burke, A., Vrac, M., Levavasseur, G., Dumas, C., Kageyama, M., Ramstein, G., 2018. Comparison of spatial downscaling methods of general circulation model results to study climate variability during the Last Glacial Maximum. Geosci. Model Dev. 11, 2563–2579. doi:10.5194/gmd-11-2563-2018

Le Ravalec, M., Noetinger, B., Hu, L.Y., 2000. The FFT moving average (FFT-MA) generator: An efficient numerical method for generating and conditioning Gaussian simulations. Mathematical Geology 32, 701–723.

Levin, A., Lischinski, D., Weiss, Y., 2004. Colorization using optimization. ACM Transactions on Graphics 23, 689–694. doi:10.1145/1015706.1015780

Li, J., Heap, A.D., 2014. Spatial interpolation methods applied in the environmental sciences: A review. Environ Model Softw 53, 173–189. doi:10.1016/j.envsoft.2013.12.008

Li, L., Zhang, Q., Huang, D., 2014. A Review of Imaging Techniques for Plant Phenotyping. Sensors 14, 20078–20111. doi:10.3390/s141120078

Li, X., Mariethoz, G., Lu, D., Linde, N., 2016. Patch-based iterative conditional geostatistical simulation using graph cuts. Water Resources Research 52, 6297–6320. doi:10.1002/2015WR018378

Liu, Y., Journel, A., 2004. Improving Sequential Simulation with a Structured Path Guided by Information Content. Mathematical Geology 36, 945–964. doi:10.1023/B:MATG.0000048800.72104.de

Lochbühler, T., Pirot, G., Straubhaar, J., Linde, N., 2013. Conditioning of Multiple-Point Statistics Facies Simulations to Tomographic Images. Math Geosci 46, 625–645. doi:10.1007/s11004-013-9484-z

Lu, D., Mausel, P., Brondízio, E., Moran, E., 2004. Change detection techniques. International Journal of Remote Sensing 25, 2365–2401. doi:10.1080/0143116031000139863

Mahlein, A.K., Steiner, U., Dehne, H.W., Oerke, E.C., 2010. Spectral signatures of sugar beet leaves for the detection and differentiation of diseases. Precision Agric 11, 413–431. doi:10.1007/s11119-010-9180-7

Mahmud, K., Mariethoz, G., Caers, J., Tahmasebi, P., Baker, A., 2014. Simulation of Earth textures by conditional image quilting. Water Resources Research 50, 3088–3107. doi:10.1002/2013WR015069

Mandanici, E., Bitelli, G., 2016. Preliminary Comparison of Sentinel-2 and Landsat 8 Imagery for a Combined Use. Remote Sensing 8, 1014–10. doi:10.3390/rs8121014

Mariethoz, G., 2010. A general parallelization strategy for random path based geostatistical simulation methods. Computers and Geosciences 36, 953–958. doi:10.1016/j.cageo.2009.11.001

Mariethoz, G., Caers, J., 2014. Multiple-point geostatistics: stochastic modeling with training images. Wiley.

Mariethoz, G., Kelly, B.F.J., 2011. Modeling complex geological structures with elementary training images and transform-invariant distances. Water Resources Research 47, 959–14. doi:10.1029/2011WR010412

Mariethoz, G., Lefebvre, S., 2014. Bridges between multiple-point geostatistics and texture synthesis_ Review and guidelines for future research. Computers and Geosciences 66, 66–80. doi:10.1016/j.cageo.2014.01.001

Mariethoz, G., Renard, P., Straubhaar, J., 2010. The Direct Sampling method to perform multiple-point geostatistical simulations. Water Resources Research 46. doi:10.1029/2008WR007621

Matheron, G., 1973. The intrinsic random functions and their applications. Advances in Applied Probability 5, 439–468. doi:10.2307/1425829

Meerschman, E., Pirot, G., Mariethoz, G., Straubhaar, J., Van Meirvenne, M., Renard, P., 2013. A practical guide to performing multiple-point statistical simulations with the Direct Sampling algorithm. Computers and Geosciences 52, 307–324. doi:10.1016/j.cageo.2012.09.019

Munyati, C., 2010. Wetland change detection on the Kafue Flats, Zambia, by classification of a multitemporal remote sensing image dataset. International Journal of Remote Sensing 21, 1787–1806. doi:10.1080/014311600209742

Nadar, 1900. Quand j'Étais Photographe.

Noda, H., Korekuni, J., Niimi, M., 2006. A colorization algorithm based on local MAP estimation. Pattern Recognition 39, 2212–2217. doi:10.1016/j.patcog.2006.03.015

Oriani, F., Ohana-Levi, N., Marra, F., Straubhaar, J., Mariethoz, G., Renard, P., Karnieli, A., Morin, E., 2017. Simulating Small-Scale Rainfall Fields Conditioned by Weather State and Elevation: A Data-Driven Approach Based on Rainfall Radar Images. Water Resources Research 15, 265. doi:10.1002/2017WR020876

Pekel, J.-F., Cottam, A., Gorelick, N., Belward, A.S., 2016. High-resolution mapping of global surface water and its long-term changes. Nature Publishing Group 540, 418–422. doi:10.1038/nature20584

Pirot, G., Straubhaar, J., Renard, P., 2014. Simulation of braided river elevation model time series with multiple-point statistics. Geomorphology 214, 148–156. doi:10.1016/j.geomorph.2014.01.022

Plaza, A., Benediktsson, J.A., Boardman, J.W., Brazile, J., Bruzzone, L., Camps-Valls, G., Chanussot, J., Fauvel, M., Gamba, P., Gualtieri, A., Marconcini, M., Tilton, J.C., Trianni, G., 2009. Recent advances in techniques for hyperspectral image processing. Remote Sensing of Environment 113, S110–S122. doi:10.1016/j.rse.2007.07.028

Pratt, W.K., 2007. Digital Image Processing. Wiley-Interscience.

Rasera, L. G., Gravey, M., Lane, S. N., & Mariethoz, G., 2019. Downscaling Images with Trends Using Multiple-Point Statistics Simulation: An Application to Digital Elevation Models. Mathematical Geosciences. https://doi.org/10.1007/s11004-019-09818-4

Renard, P., Allard, D., 2013. Connectivity metrics for subsurface flow and transport. Advances in Water Resources 51, 168–196. doi:10.1016/j.advwatres.2011.12.001

Rodriguez V., P., 2002. A radix-2 FFT algorithm for Modern Single Instruction Multiple Data (SIMD) architectures. In IEEE International Conference on Acoustics Speech and Signal Processing. IEEE. https://doi.org/10.1109/icassp.2002.5745335

Rodríguez, P., 2002. A radix-2 FFT algorithm for modern single instruction multiple data (SIMD) architectures. ieeexplore.ieee.org

Semary, N.A., Hadhoud, M.M., Kilani, El, W.S., Ismail, N.A., 2007. Texture recognition based natural gray images coloring technique, in:. Presented at the National Radio Science Conference, NRSC, Proceedings, IEEE, pp. 1–12. doi:10.1109/NRSC.2007.371374

Shannon, 1948. A mathematical theory of communication. Wiley Online Library

Shen, J. P., 2013. Modern processor design: fundamentals of superscalar processors. Long Grove: Waveland Press.

Shendryk, I., Broich, M., Tulbure, M.G., McGrath, A., Keith, D., Alexandrov, S.V., 2016. Mapping individual tree health using full-waveform airborne laser scans and imaging spectroscopy: A case study for a floodplain eucalypt forest. Remote Sensing of Environment 187, 202–217. doi:10.1016/j.rse.2016.10.014

Silverman, B.W., 1986. Density Estimation for Statistics and Data Analysis. CRC Press.

Srivastava, R.M., 2018. The Origins of the Multiple-Point Statistics (MPS) Algorithm. Springer International Publishing. doi:10.1007/978-3-319-78999-6_32

Straubhaar, J., Renard, P., Mariethoz, G., Froidevaux, R., Besson, O., 2011. An Improved Parallel Multiple-point Algorithm Using a List Approach. Math Geosci 43, 305–328. doi:10.1007/s11004-011-9328-7

Strebelle, S., 2002. Conditional simulation of complex geological structures using multiple-point statistics. Mathematical Geology 34, 1–21. doi:10.1023/A:1014009426274

Strebelle, S., Cavelius, C., 2013. Solving Speed and Memory Issues in Multiple-Point Statistics Simulation Program SNESIM. Math Geosci 46, 171–186. doi:10.1007/s11004-013-9489-7

Strebelle, S., Payrazyan, K., Caers, J., 2002. Modeling of a Deepwater Turbidite Reservoir Conditional to Seismic Data Using Multiple-Point Geostatistics, SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers. doi:10.2118/77425-MS

Tadić, J.M., Qiu, X., Miller, S., Michalak, A.M., 2017. Spatio-temporal approach to moving window block kriging of satellite data v1.0. Geosci. Model Dev. 10, 709–720. doi:10.5194/gmd-10-709-2017

Tadić, J.M., Qiu, X., Yadav, V., Michalak, A.M., 2015. Mapping of satellite Earth observations using moving window block kriging. Geosci. Model Dev. 8, 3311–3319. doi:10.5194/gmd-8-3311-2015

Tadjudin, S., Landgrebe, D., 1998. Classification of high dimensional data with limited training samples.

Tahmasebi, P., 2017. Structural Adjustment for Accurate Conditioning in Large-Scale Subsurface Systems. Advances in Water Resources 1–52. doi:10.1016/j.advwatres.2017.01.009

Tahmasebi, P., Hezarkhani, A., Sahimi, M., 2012. Multiple-point geostatistical modeling based on the cross-correlation functions. Computational Geosciences 16, 779–797. doi:10.1007/s10596-012-9287-1

Tahmasebi, P., Sahimi, M., Mariethoz, G., Hezarkhani, A., 2012. Accelerating geostatistical simulations using graphics processing units (GPU). Computers and Geosciences 46, 51–59. doi:10.1016/j.cageo.2012.03.028

Tan, X., Tahmasebi, P., Caers, J., 2013. Comparing Training-Image Based Algorithms Using an Analysis of Distance. Math Geosci 46, 149–169. doi:10.1007/s11004-013-9482-1

Tang, Y., Atkinson, P.M., Zhang, J., 2015. Downscaling remotely sensed imagery using area-to-point cokriging and multiple-point geostatistical simulation. ISPRS Journal of Photogrammetry and Remote Sensing 101, 174–185. doi:10.1016/j.isprsjprs.2014.12.016

USGS, 2008. Declassified Intelligence Satellite Photographs 1–2.

Vaduva, C., Costachioiu, T., Patrascu, C., Gavat, I., Lazarescu, V., Datcu, M., 2013. A Latent Analysis of Earth Surface Dynamic Evolution Using Change Map Time Series. IEEE Trans. Geosci. Remote Sensing 51, 2105–2118. doi:10.1109/TGRS.2012.2219316

van der Meer, F. D., van der Werff, H. M. A., van Ruitenbeek, F. J. A., Hecker, C. A., Bakker, W. H., Noomen, M. F., … Woldai, T., 2012. Multi- and hyperspectral geologic remote sensing: A review. International Journal of Applied Earth Observation and Geoinformation, 14(1), 112–128. https://doi.org/10.1016/j.jag.2011.08.002

Vannametee, E., Babel, L. V., Hendriks, M. R., Schuur, J., de Jong, S. M., Bierkens, M. F. P., & Karssenberg, D., 2014. Semi-automated mapping of landforms using multiple point geostatistics. Geomorphology, 221, 298–319. https://doi.org/10.1016/j.geomorph.2014.05.032

Varshney, P.K., Arora, M.K., 2013. Advanced Image Processing Techniques for Remotely Sensed Hyperspectral Data. Springer Science & Business Media, Berlin, Heidelberg. doi:10.1007/978-3-662-05605-9

Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Trans. on Image Process. 13, 600–612. doi:10.1109/tip.2003.819861

Wei, L.-Y., Levoy, M., 2000. Fast texture synthesis using tree-structured vector quantization, the 27th annual conference. ACM Press/Addison-Wesley Publishing Co., New York, New York, USA. doi:10.1145/344779.345009

Welsh, T., Ashikhmin, M., Mueller, K., 2002. Transferring color to greyscale images. SIGGRAPH 277. doi:10.1145/566570.566576

Weng, Q., 2001. A remote sensing–GIS evaluation of urban expansion and its impact on surface temperature in the Zhujiang Delta, China. International Journal of Remote Sensing 22, 1999–2014. doi:10.1080/01431160118847

Wojcik, R., McLaughlin, D., Konings, A. G., & Entekhabi, D., 2009. Conditioning Stochastic Rainfall Replicates on Remote Sensing Data. IEEE Transactions on Geoscience and Remote Sensing, 47(8), 2436–2449. https://doi.org/10.1109/tgrs.2009.2016413

Wojcik, R., McLaughlin, D., on, A.K.I.T., 2009, 2009. Conditioning stochastic rainfall replicates on remote sensing data. ieeexplore.ieee.org

Yang, J., Gong, P., Fu, R., Zhang, M., Chen, J., Liang, S., Xu, B., Shi, J., Dickinson, R., 2013. The role of satellite remote sensing in climate change studies. Nature Climate change 3, 875–883. doi:10.1038/nclimate1908

Yin, G., Mariethoz, G., McCabe, M., 2017. Gap-Filling of Landsat 7 Imagery Using the Direct Sampling Method. Remote Sensing 9, 12. doi:10.3390/rs9010012

Zanter, K., 2005. Landsat 8 (L8) data users handbook. LSDS-1574 Version.