Metrics for quantifying verbatim copy in multiple-point geostatistics realisations

Merijn Testroote

0468398

Master Thesis

Credits: 14 EC

MSc Applied Data Science

Utrecht University

First supervisor

dr. Mathieu Gravey

Second supervisor

prof. dr. Derek Karssenberg

June 30, 2022

Abstract

In multiple-point geostatistics, the one-on-one copying of patches from a training image to a realisation in multiple-point geostatistics simulations is called verbatim copy. Verbatim copy is an important quality metric which has to be minimised in relation to other quality metrics. Previous methods used computer vision techniques on the realisation image to quantify verbatim copy. In that way, the problem becomes hard and complex to solve. To get around the complex nature of computer vision, the *index coherence map* (ICM) was used. Various metrics were created to transform the ICM into usable values for analysis. A synthetic dataset of ICMs with a know verbatim copy degree was created to validate and test the metrics. A sliding window method was able to correctly reconstruct the verbatim copy in the synthetic ICMs. Using hierarchical clustering and PCA more metrics were extracted. These metrics provide useful insights into the quantity and shape of verbatim copy, allowing to have a summarised overview of the quality.

Contents

1 Introduction **1**

2 Data **3**
- 2.1 QS Dataset . . . 3
- 2.2 Data exploration . . . 4
- 2.3 Data preparation . . . 5

3 Methods **6**
- 3.1 Sliding Window Matches . . . 6
- 3.2 Grid Based Agglomerative Hierarchical Clustering . . . 8
- 3.3 Principal component analysis of verbatim windows . . . 8
- 3.4 Validation . . . 9

4 Results **10**

5 Discussion **15**

6 Conclusion **17**

1 Introduction

(a) (b)

Figure 1: Two frequently used training images for MPS. Adapted from Mariethoz and Caers, 2014. (a) Stones (b) TI depicting channels (Strebelle, 2002))

In many climate and geological simulations, the starting state of the model is the *training image* (TI). For example, a benchmark training image used in water flow simulations is the strebelle image (Fig 1b). To do the stochastic simulation of random fields, previous research proposed methods to sample multiple realisations from one training image. This class of methods is called *multiple-point geostatistics* (MPS) (Mariethoz & Caers, 2014). MPS takes higher order relations into account and there are algorithms for both discrete and continuous data. MPS has a wide variety of usages including the enhancement of microscopic imaging (Wang et al., 2022), the reconstruction of porous media (Zhang & Du, 2012) and the interpolation of remote-sensing data (Yin et al., 2017; Zakeri & Mariethoz, 2021). MPS algorithms generally come in two types, patch-based and pixel-based. Patch-based algorithms search for an optimal cut of the training image into patches, the realisation image is then the quilting of these patches. In pixel-based algorithms, for every position in the realisation image pixel-based algorithms construct a conditional distribution of the neighbourhood in the training image. A realisation is the sampling from these distributions. When similar patterns are rare and there is only a single option to sample from, one-on-one copying from the training image occurs. This effect is called *verbatim copy*. Some verbatim copy is inevitable. But a good realisation is realistic and has a low verbatim copy.

Abdollahifard et al., performed a quality evaluation of MPS realisations (Abdollahifard et al., 2019). They quantified verbatim copy through pattern innovation. Pattern innovation was measured using computer vision techniques including scale-invariant feature transform (SIFT). SIFT was used to detect keypoints in the training image and the realisation and then find the matching keypoints in the two images. They then validated their methods using (but not exclusively), human participant detection of the verbatim copy in the realisation image. Pattern innovation is not directly translatable to verbatim copy because pattern innovation is dependent on the richness of the training image. Also, when using the realisation image, part of the verbatim copy information has been lost because unless every pixel in the training image has a unique value, the origin cannot be perfectly reconstructed.

Another way to measure verbatim copy is to construct the *index coherence map* (ICM). The ICM is the map corresponding to a training image and realisation set, where every value in the ICM corresponds to the location in the training image from where the pixel in the realisation originates (Mariethoz & Caers, 2014).

When the training image and the realisation are the same, the ICM is the linear index of the training image. In this case, the ICM contains every value between 1 and the number of pixels in the training image. When this is the case the realisation has maximum verbatim copy with respect to the training image. On the other hand, an ICM that is pure noise means that there is no verbatim copy in the realisation image. Local verbatim copy in the realisation can occur when two direct neighbouring pixels have a distance of 1 in the ICM. This means that the two pixels were neighbours in the training image and they are also neighbours in the realisation. When two pixels in the ICM are each other's linear progression by the number of columns and rows, as if the pixels were from the ICM of the maximum verbatim, we also consider it verbatim copy. Because the ICM has a finite number between 1 and the number of pixels in the training image, the ICM possibly allows translating the problem of quantifying verbatim copy to a data-science problem. This is in comparison with using the training image, which is a hard computer vision problem.

With most MPS algorithms, the ICM is not a byproduct of the algorithm. The pixel-based MPS algorithm *Direct Sampling* introduced MPS for continuous images (Mariethoz et al., 2010). Direct Sampling can also produce an ICM without extra computational cost. *Quick Sampling V1.0* (QS), a more recently proposed pixel-based MPS algorithm, improves upon Direct Sampling by allowing for sampling in predictable constant time and can also produce an ICM (Gravey & Mariethoz, 2020). QS needs hyper-parameter tuning to improve various quality metrics one of which is the amount of verbatim copy. There are no reliable and direct ways to quantify verbatim copy yet and as of writing, there is no known labelled data-set of training image and realisation pairs with a known verbatim copy degree. To translate the problem of quantifying verbatim copy into a data-science problem, we propose to construct a synthetic ICM dataset with known verbatim copy that models the verbatim copy in the ICM. This synthetic ICM dataset can then be used to calibrate verbatim copy metrics, which can be used to predict the known verbatim copy in the data.

The methods that are built to work for the synthetic ICM dataset can then be used to quantify the verbatim copy in QS realisations.

2 Data

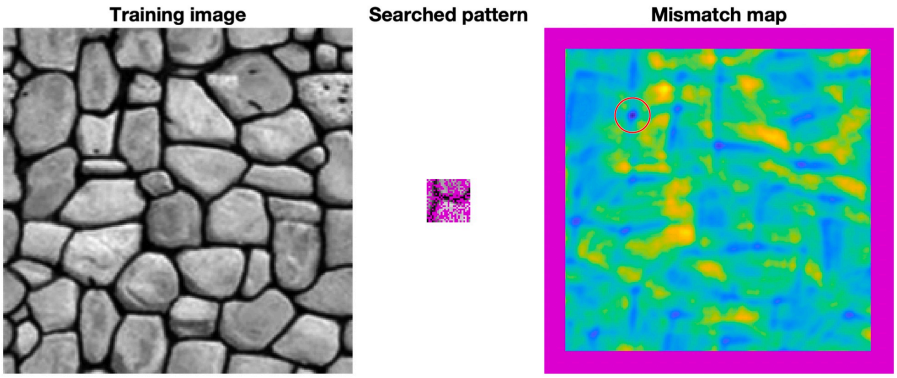


Figure 2: Left: training image, middle: search pattern, right: mismatch map. Adapted from (Gravey & Mariethoz, 2020)

2.1 QS Dataset

The authors of Gravey and Mariethoz, (2020) provided us with QS realisations from two training images. The first is a 250×250 image from (Strebelle, 2002), referred to as strebelle after this (Fig 1a). The second training image "stones" is a 200×200 image depicting stones, much used in MPS benchmarks, (Fig 1b).

QS has a wide set of user-configurable parameters. For a single value in the realisation, QS works by computing a mismatch map between the TI and the current realisation. The mismatch map is constructed based on the neighbourhood around a pixel (Fig. 2). The size of the neighbourhood is defined by the parameter N. Then this pixel is sampled from the k set of best candidates having the lowest values in the mismatch map (Gravey & Mariethoz, 2020). In our dataset we have realisations for every N between 1 and 199 and for every k:

$$k \in [1, 1.01, 1.02, 1.05, 1.10, 1.15, 1.20, 1.30, 1.50, 1.70, 2, 2.50, 3, 5, 10]$$

The realisation and the ICM have the same dimensions as the training image. The ICM has values between 0 and $(n-1)$, where n is the product of the row size and refers to the position from which the pixel in the realisation was sampled from the flattened training image. This flattened training image can be constructed by appending every row of the image to the right of the upper row. The row and column index can be reconstructed using the quotient and the remainder of $\frac{i}{n}$ where i is the value of the ICM and n is the number of values in the matrix.

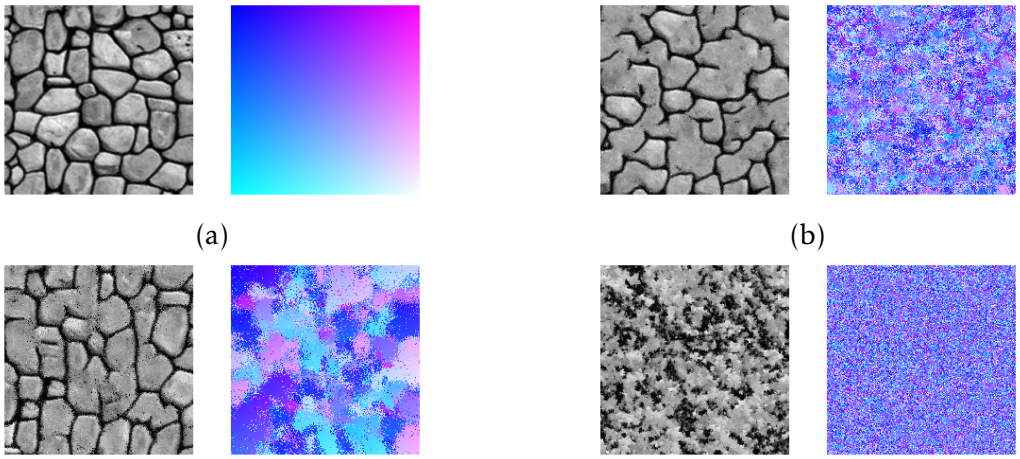


Figure 3: Different samples from the QS-stones dataset. (a) Left: Training image, right: ICM colour gradient, maximum verbatim copy. (b) Low verbatim copy and realistic. (c) High verbatim copy and realistic. (d) Low verbatim copy and non-realistic.

2.2 Data exploration

In high verbatim copy realisations, we expect to see patches from the training image being copied into the realisation. This results in the values of the ICM, corresponding to the verbatim patch, having a low distance between them. For two neighbouring values in the patch in the ICM, we expect them to have a distance of 1. When two values are further apart we expect them to have a distance equal to their distance in the matrix (column/row wise).

In MPS, the goal is to generate realistic samples, keeping the natural structure in the training image, but adding a stochastic component. A consequence of verbatim copy is that the patch is inherently a realistic copy of the original. So in cases with a lot of verbatim copy, we expect to have a realistic realisation. Although, this is not always the case. A realisation with low verbatim can either be random noise, and thus be nonsensical, or be realistic. The best scenario is a low verbatim but realistic realisation.

The ICM can be transformed into a colour image by using a colour gradient image with two varying colour dimensions where maximum verbatim copy results in a perfect gradient (Fig. 3a). We can then randomly take realisations from the dataset, visually searching for cases of verbatim copy.

We select both on the look of the realisation image and the ICM (Fig. 3).

- Figure 3a a training image on the left and a perfect ICM (max. verbatim copy) on the right.
- Figure 3b shows a realistic image, the ICM looks noisy to the human eye.
- Figure 3c shows a realistic realisation, and we can see gradient colour patches in the ICM.
- Figure 3d does not look like a plausibly realistic scenario of the training image, and the ICM looks like noise.

By comparing the ICM in Figure 3c and the original Figure 3a, and then looking at the images, we see whole stones being copied, a kind of unwanted verbatim copy.

2.3 Data preparation

The goal here is to propose metrics to discern between Figure 3b and Figure 3c, realistic and low verbatim, and realistic and high verbatim. To do this a more quantitative method than visual analysis is needed to calibrate the verbatim copy metrics. Using a synthetic dataset that has similar verbatim properties to what can be seen with the human eye in figure 3, verbatim copy could be better quantified. A synthetic dataset of different ICM with a varying verbatim copy was crafted to do this. These ICMs do not have a linked training image or a realisation. They only show similar verbatim patches as were seen in the previous figures.

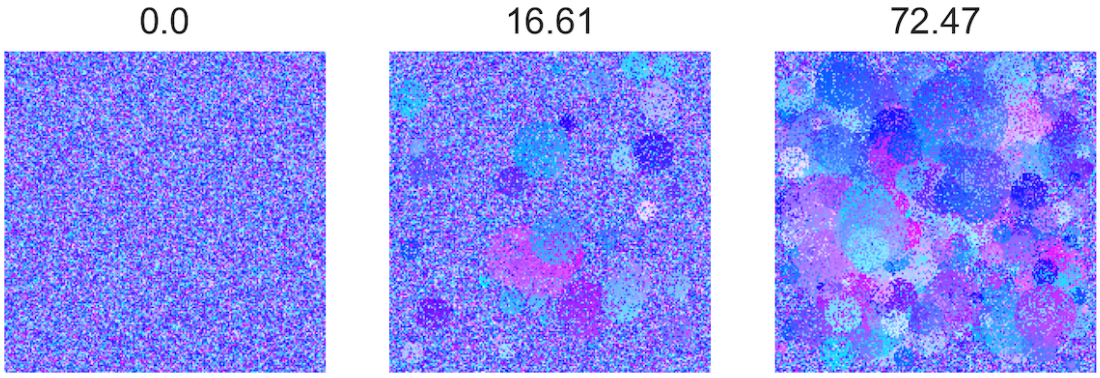


Figure 4: Three samples from the synthetic data. Left: zero verbatim. Middle: 16.61% verbatim. Right: 72.47% verbatim

Circular verbatim patches were added from a perfect ICM (Fig. 3a right) onto a noise background. For example, the perfect ICM of 3×3 is $\left(\begin{smallmatrix}1&2&3\\4&5&6\\7&8&9\end{smallmatrix}\right)$. To create different degrees of verbatim copy average number of verbatim patches p is varied. For every sample a full random ICM of size $m \times m$, uniformly distributed between $\{1, \cdots, m^2\}$ is created. Then n patches from a gradient map where $n \sim Normal(p, 5)$ are added to the full random ICM. Every patch has a random radius r where $r \sim Normal(2, 10)$. For both n and r, negative values are set to zero. The origin and destination location of each patch are both sampled as $(x, y) \sim Uniform(radius, m - radius)$ and are independent. In order to add noise, from every patch pixels are knocked out with a probability $P(knocked\_out = True) = 0.4$.

Using the above method, for every n between $1 \cdots 600$ 10 samples were generated, totalling 6000. The verbatim patches in every sample, were also added to an empty matrix M instead of the full random ICM. M was then used to calculate the percentage of pixels that are verbatim.

3 Methods

The goal is to study different aspects of verbatim. To find where verbatim is located, a sliding window method is proposed. Here the number of neighbours an index in the realisation image has that are from the same place in the training image are counted. This result is then used in a clustering algorithm to find the size and amount of verbatim patches. To measure the shape and density of the verbatim patches, the windows extracted from the sliding window method are transformed to a single metric using *principal component analysis* (PCA).

The metrics above will be validated using the generated synthetic data. The verbatim detection will output a verbatim percentage, this will then be used in a linear regression analysis. A perfect verbatim detection algorithm will output the same verbatim percentage as the truth known from the synthetic data. Linear regression will thus reflect if this is the case.

Every method will result in a single metric which can be further analysed (see table 1).

3.1 Sliding Window Matches

The input to the sliding window matches is an ICM A. Every $A_{i,j}$ refers to the location in the training image the pixel from the realisation image was sampled from. An ICM B of the same size as A is constructed by filling the rows with increasing numbers, from 1 to $(m * n)$, like the right ICM in figure 3a. B is the ICM of a realisation image that is the exact copy of the training image. In MPS one can think of A as being sampled with replacement from B, thus $\{A_{i,j} \in B | i, j \in \mathbb{N}\}$ holds. Verbatim copy is then defined as the number of positional matches between an area around $A_{i,j}$ and $B_{m,n}$ where $A_{i,j} = B_{m,n}$. Intuitively, this is the number of neighbours in A around a pixel that were sampled from the same area in B. Note here that no rotation or scaling of the sampled verbatim is assumed. The number of matches in a window around every value in $A_{i,j}$ is counted, this is then weighted using the number of pixels in a window (Eq. 1).

$$V_{m,n} = \frac{1}{(w*2+1)^2} \sum_{a=-w}^{w} \sum_{b=-w}^{w} \delta_{B_{i+a,j+b}, A_{m+a,n+b}} \tag{1}$$

$$i = \lfloor A_{m,n}/k \rfloor \tag{2}$$

$$j = (A_{m,n} - \lfloor A_{m,n}/k \rfloor) \tag{3}$$

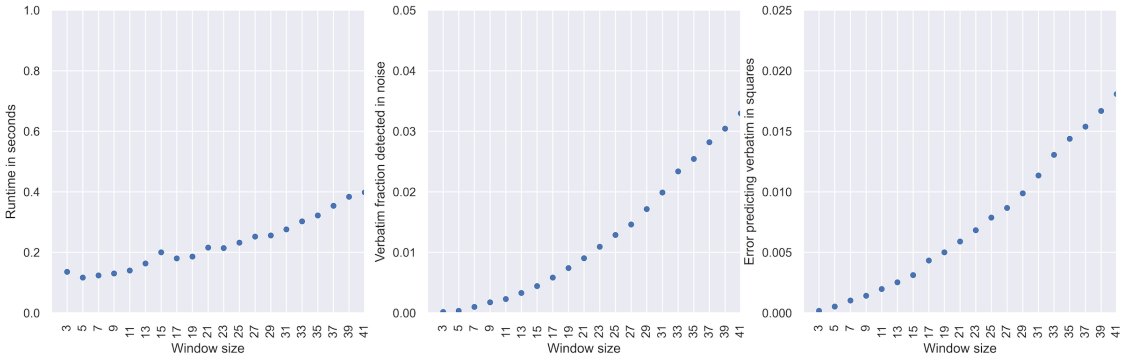
Where $\delta_{i,j}$ is 1 if $i = j$ else 0, V is the verbatim density map, w is the window size around a pixel, $(w * 2 + 1)^2$ is the area of the window, k is the maximum index, $k = m_{max} * n_{max}$, i and j are the indexes of the origin pixel in B.

To test if the method was robust to noise, it was tested on 100 pure noise ICMs. The mean verbatim percentage, calculated as the number of non-zero values in V divided by $m * n$, was on average 0.11% and the maximum was 0.17%. Note that the maximum verbatim percentage is 100. Robustness against noise was also tested for different values of the window size w, see Fig. 5. In addition, run-time was recorded for different w. From these tests, there is no evidence of an increase in performance when increasing window size w. To detect long-range verbatim, w

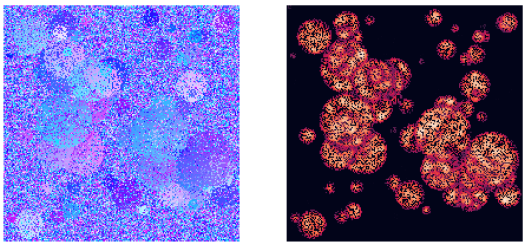
needs to be bigger, thus exponentially increasing run-time. But in some scenarios and implementations, this might not pose a problem.

This method is translated to a single metric (V1) by counting the number of values in V that are 1 or higher and then dividing by the number of values in V.

$$V1 = \frac{|V_{m,n} > 0|}{m * n} \tag{4}$$



(a)



(b)

Figure 5: (a) The effect of window size in the sliding window method. Left: The run-time of the method on a 200×200 sample. Middle: The fraction of verbatim predicted in pure noise. Right: The error in predicting the verbatim fraction from a synthetic sample where the true percentage of verbatim pixels was known. (b) Verbatim detection using the sliding window method on a random sample from the synthetic data. Left: index coherence matrix as input. Right: Verbatim density as output.

3.2 Grid Based Agglomerative Hierarchical Clustering

To quantify the size and number of verbatim patches in the verbatim density map constructed in the previous method, *Grid Based Agglomerative Hierarchical Clustering* (GBAHC) was used. Hierarchical clustering in this scenario is ideal because it can find an arbitrary amount of clusters. But, ordinary hierarchical clustering has $\mathcal{O}(n^3)$ complexity. Because of the nature of the verbatim density map, having arbitrary clusters between any pixels does not make sense. We can constrain the clustering algorithm with a grid, only considering the four direct neighbours of a pixel to cluster. This reduces the complexity to $\mathcal{O}(n)$ (Murtagh & Contreras, 2017). Hierarchical clustering takes a distance metric and a linkage criterion. The distance metric depends on the type of linkage. For image clustering in noisy data, only complete and ward linkage is suitable. Ward linkage together with the euclidean distance metric is shown to be effective for image clustering (Bruse et al., 2017). Experiments with the method described below confirm this.

We start with an image V of size $m \times n$. In our case, this is the verbatim density map constructed in section 3.1. We then construct a graph G, which is the list of directly connected pixels. In a 2d image, we have $(m * n) * 4 - m * 2 - n * 2$ connections, corrected for the edges of the image. Then the set of every connected pixel $C_0 = \{\{V_{0,0}, V_{1,0}\} \cdots \{V_{m,n-1}, V_{m,n}\}\}$ is our first set of clusters. We then repeatedly calculate Ward's minimum variance between all clusters that are connected by at least one pixel in the graph G, see Eq. 5 (Ward, 1963). The two clusters with the lowest Ward's minimum variance will be merged. We will continue merging until we reached a pre-defined threshold minimum variance all clusters are merged.

$$d_{i,j} = \|X_i - X_j\|^2 \tag{5}$$

The result C is the set of clusters and the assignment of every value in V to a cluster. From this one can plot a dendrogram, which is useful for the analysis of a single clustering result (Espinoza et al., 2012). To summarise the clustering results, the number of clusters (V2) and the average cluster size (V3) were calculated. V3 is calculated as the number of pixels in the cluster divided by the total number of pixels.

$$V2 = |C| \tag{6}$$

$$V3 = \frac{1}{V2} \sum_{i=0}^{V2} |C_i| \tag{7}$$

3.3 Principal component analysis of verbatim windows

To research the shape of the verbatim *principal component analysis* (PCA) was used on the windows W_i constructed in section 3.1. PCA is a fast and proven dimensionality reduction method. PCA is the eigenvalue decomposition of the covariance matrix $X^T X$ (Pearson, 1901). In this case, every row in X is a flattened window in W. The columns then refer to every position of the window. $X_{i,0}$ being the $W_{i,0,0}$ and $X_{i,n}$ being $W_{i,w,w}$ where n is the amount of pixels in the image, w being the sliding window size. $X^T X$ is then the covariance of each pixel in the index coherence matrix. This is high when two pixels have verbatim in multiple

windows. The eigenvalue decomposition of $X^T X$ (PCA) is then the set of principal components w_i which explain the variance the best. The variance, in this case, is the difference between the verbatim windows. Using $X \cdot w_i$ one can transform all verbatim windows to the single value connected to the principal component. Every principal component explains a part of the variance. The first few principal components usually explain the most variance.

In both the single QS parameter analysis and the multivariate analysis, analysing all windows is not feasible. In the worst case, we have m^2 windows for a $m \times m$ image per parameter configuration. Because of this, n verbatim windows were sampled at random positions from every parameter configuration. The verbatim windows were combined into one $X_i = (v * n) \times b$ matrix, where v is the number of realisations in one parameter configuration, n is the number of samples, b is the area of the window. For every parameter configuration i we ran PCA on X_i resulting in W_i, the set of principal components.

For the multivariate analysis, the number of sampled windows per parameter was decreased. Then the whole process was repeated 20 times while averaging the principal components. Because the QS data was stored in separate files per parameter k, the right order of execution was chosen to speed up input-output time.

$$V4 = X \cdot w_1 \tag{8}$$

3.4 Validation

To validate the previously described methods against the synthetic data, we used linear regression analysis. A linear relation between the number of positive values in V from Eq. 1 and the true verbatim in the synthetic data is expected. The *root mean square error* (RMSE) is also calculated as eq. 9. The RMSE reflects the average error in the unit that was predicted.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}} \tag{9}$$

Verbatim metrics			
Short	Description	Method	Formula
V1	Percentage of pixels that have a neighbour verbatim pixel	Sliding window	4
V2	Number of clusters found	GBAHC	6
V3	Average cluster size	GBAHC	7
V4	Verbatim density (First component)	PCA	8

Table 1: Verbatim metrics

4 Results

The sliding window matches visually shows a correct reconstruction of the verbatim patches in the synthetic data (Fig. 5b). In the full synthetic dataset the V1 metric (7x7 window) correctly predicted the the percentage of pixels that have verbatim with $RMSE = 0.3098$ (Fig. 6e). Based on visual analysis, the density map produced by the sliding window matches method can capture the presence of verbatim copy in a low and high verbatim sample in both training images (Fig. 6. In the QS dataset with the stones TI there is lower verbatim (V1) when the QS parameter k increases when ignoring the N parameter. (Fig. 6f). But for all k values lower than 2 there seems to be no effect.

The verbatim density map was then used in clustering to further analyse the data (Fig. 7a). The RMSE of the predicted number of verbatim patches by the clustering algorithm was $RMSE = 4.34$ (Fig. 7). After fitting a linear regression between the real number of clusters and the predicted number of clusters in the synthetic dataset we find $R^2 = 0.797$. The number of clusters (V2) metric decreases when the QS parameter k increases (Fig. 7d).

For the QS data, the previous metrics were also applied while varying both k and N. In the stones TI, the V1 metric only shows the extreme cases (Fig. 8a). The PCA V4 shows more variance (Fig. 8b). The kernel density of the metrics confirms that the V4 metric is more equally distributed, and can identify more types of verbatim copy (Fig. 8d). The cluster V2 metric shows a similar pattern as V4, but due to computational limitations has fewer data points (Fig. 8c). Analysis on the strebelle TI shows less verbatim copy in both V1 and V2 metric (Fig. 9).

The PCA V4 metric, for windows randomly sampled over N for a given k shows a logarithmic relation with the k-parameter. (Fig. 10a). The V4 metric on the synthetic data shows a linear relation with the known verbatim percentage, but it overestimates for most values (Fig. 10b). The first PCA component window shows a relation with the k parameter (Fig 10c). The second and the third projected component do not show a clear relation.

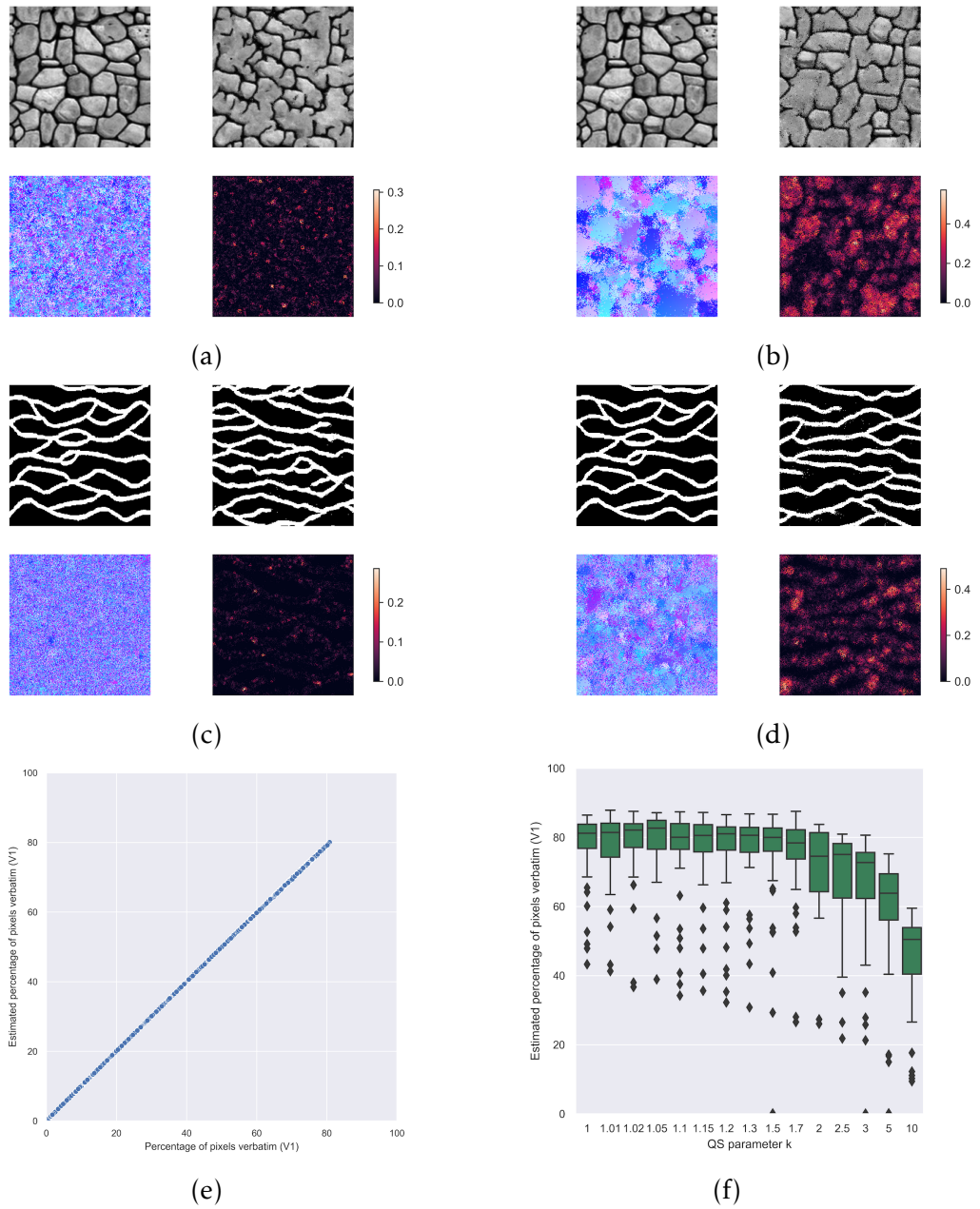
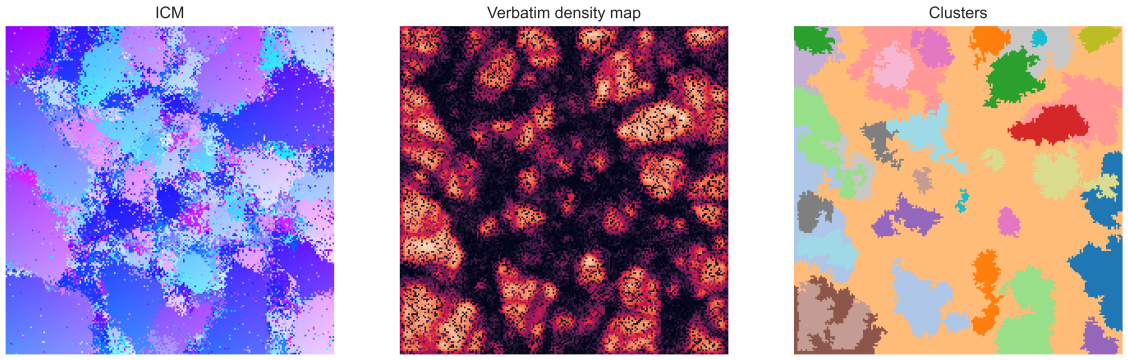
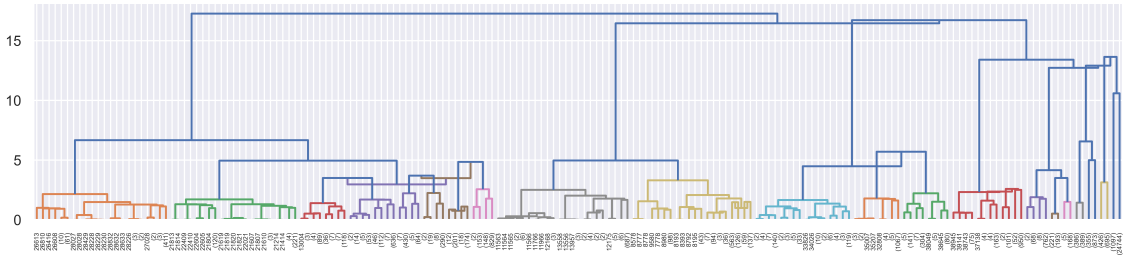


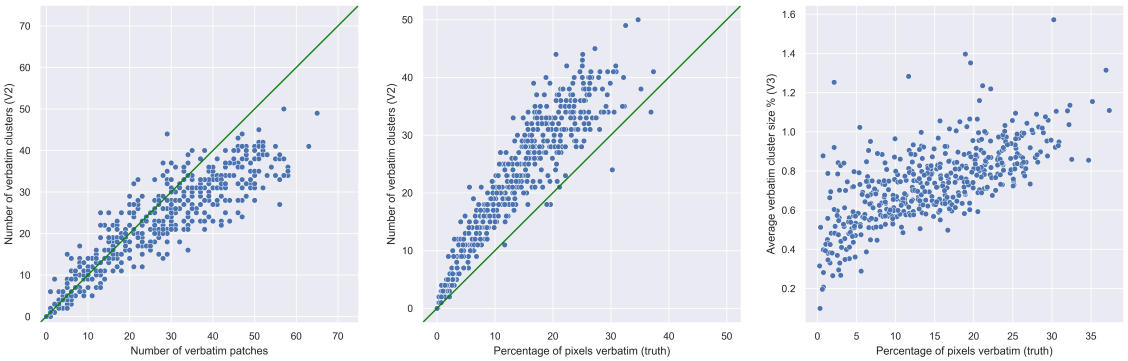
Figure 6: Verbatim detection. (a, b) A low and high verbatim copy sample from the QS dataset (stones, parameter='3.0'). In the subplot: Top-left is the training image, top-right is the QS realisation, bottom-left is the ICM and bottom-right is the detected verbatim density. (c,d) Low and high verbatim copy sample with the strebelle training image. (e) The fraction of pixels that have a positive verbatim fraction. Sampled from the synthetic dataset (f) Boxplot showing the distribution over the percentage of pixels verbatim for different values of k. Sampled from the QS dataset (stones training image), 50 samples per parameter.



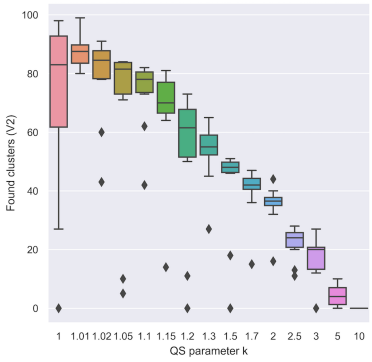
(a)



(b)



(c)



(d)

Figure 7: (a) QS stones clustering example. Threshold=3.5. Disconnected clusters with the same colour are separate clusters. (b) Dendrogram of clusters in (a) (c) Clustering on synthetic data results. The green line shows a perfect linear relation (y=x) Left: The number of verbatim patches (circles) against the number of found clusters. Middle: The truth verbatim factor against the number of found clusters. Right: The known percentage of verbatim pixels against cluster size. (d) Clustering on the stones training image with varying parameter k.

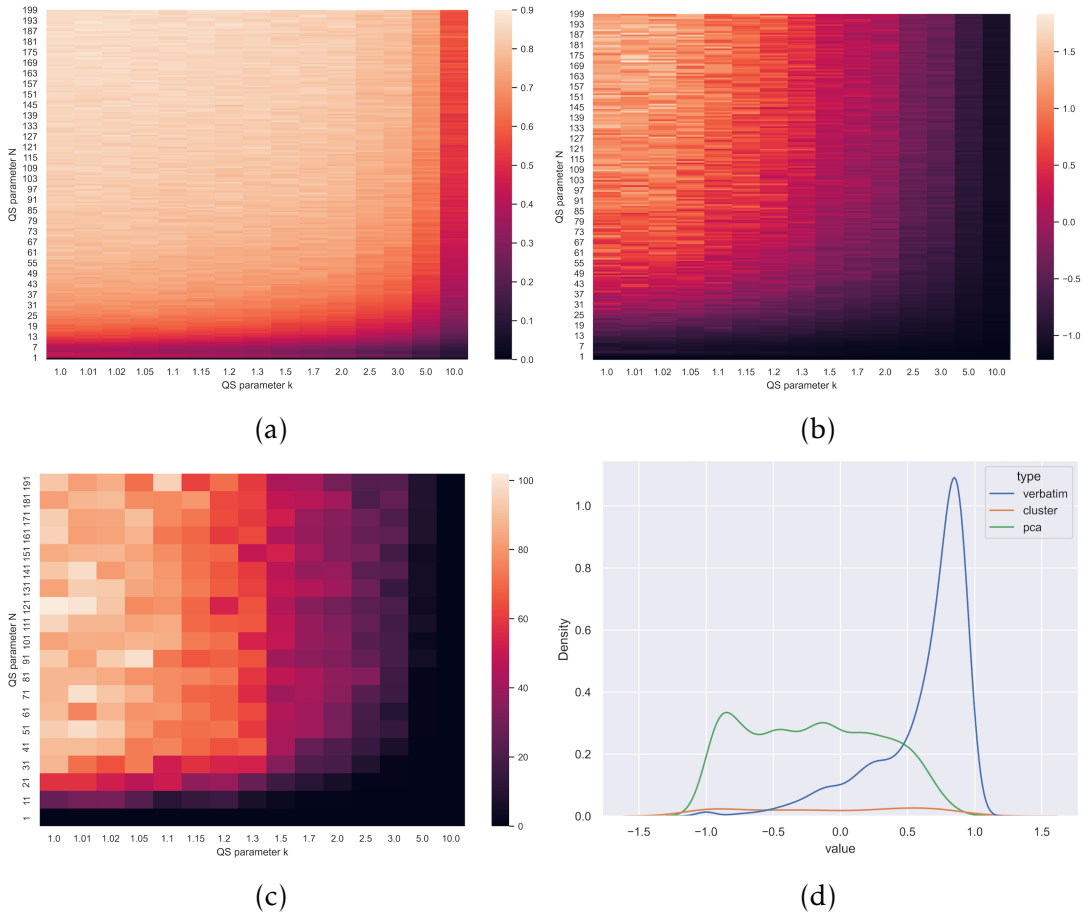


Figure 8: Verbatim metric for both k and N on the stones training image. (a) Verbatim metric (V1) for both k and N extracted from every realisation, (b) PCA metric (V4) for both k and N by sampling verbatim windows from realisations. (c) Cluster metric (V2) (d) distribution of the values in (a,b,c) showing the amount of information present in the metrics.

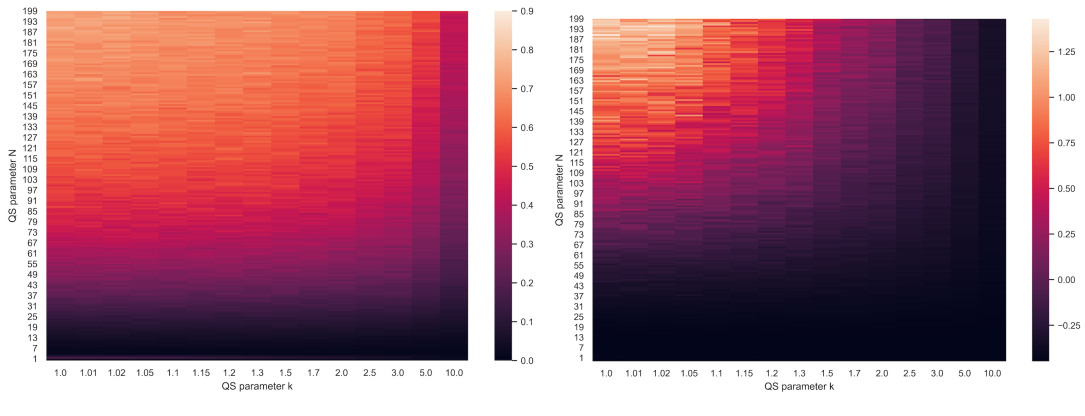


Figure 9: Verbatim metric for both k and N on the strebelle training image. (a) Verbatim metric (V1) for both k and N extracted from every realisation, (b) PCA metric (V4) for both k and N by sampling verbatim windows from realisations.

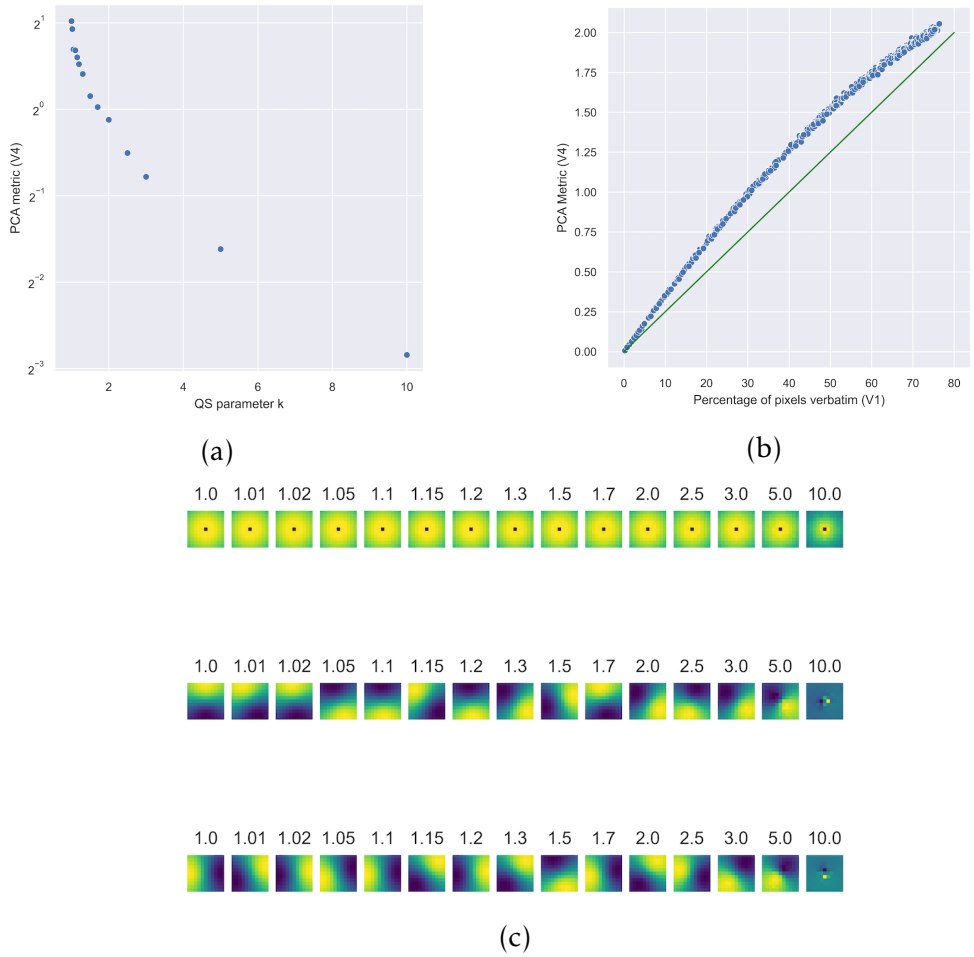


Figure 10: PCA results. (a) Left: First PCA component fitted on the QS dataset. Right: Verbatim fraction as output. (b) First PCA component fitted on the synthetic dataset against the known percentage of verbatim pixels. The green line shows expected linear increase. (c) PCA fitted on the QS dataset, the top row is the first principal component, the middle is the second and the bottom is the third row. The columns are the k-parameter.

5 Discussion

A classic data-science problem has predictor variables and target variables. For quantifying verbatim copy there are no proven existing metrics and there are no target variables to learn from. Based on expert domain knowledge and qualitative analysis of the realisations, a synthetic dataset with known degrees of verbatim copy was constructed. This allows the use of the data-science paradigm for validating new metrics. The hypothesis about what verbatim copy is and how to create it in the synthetic data directly affects the effectiveness of the results.

The sliding window matches method, generating the verbatim density map, when visually analysed, shows to correctly highlight the verbatim patches in both the synthetic data and the QS data. The usefulness of this method on its own is unclear. Because when tuning an MPS algorithm there are too many realisations generated to be visually analysed. QS realisations and the synthetic data, show no rotation or scaling of the verbatim patches. Future research can verify or adapt these methods under rotation or scaling if needed. The sliding window method likely does not work under scaling of the verbatim patches because an equal distance between the pixels is assumed in the method. A convolutional neural network can be a possible solution, as it can create many different windows for different scenarios.

The V1 metric, the percentage of positive pixels in the verbatim density map, was very effective in scoring the synthetic data. But in the QS data, did not find much information. Only a slight downtrend in the V1 in the QS data is noticeable. There seems to be no sensitivity towards lower k values, only after $k = 2$ does the V1 metric change. Which could mean that the higher the k-parameter, the lower the verbatim fraction (Fig. 6f).

The V1 metric assumes that when a pixel has a neighbour verbatim pixel, the pixel is part of a verbatim patch. Under the synthetic data, this holds and it is statistically unlikely that this happens in noise. This was verified in the tuning of the method. But in the QS-stones data, a pixel likely has a few verbatim neighbours without there being noticeable verbatim. Also, the V1 metric does not take into account how many of a pixel's neighbours are verbatim (verbatim density). Future research can see if it is possible to create other metrics based on the sliding window method that is more sensitive to verbatim copy. This metric may include a threshold on the number of verbatim pixels before adding to the total percentage.

The clustering method was able to find the number of clusters in the synthetic dataset. In the QS-stones dataset, we see that there are fewer clusters for higher parameter values. This is in line with the previous QS-stones finding. We see a slight increase in the average cluster size for higher k-parameters. But in most higher parameters there were no found clusters (Fig. 5). These results are promising and the insights provided by the hierarchical clustering paradigm are interesting. One of these is the dendrogram, the dendrogram can be a useful tool in analysing single realisations (Fig. 7b). It is possible that the distances read from the dendrogram can lead to more insights into the nature of verbatim copy in a specific problem. But the threshold distance in hierarchical clustering is problematic. With the threshold parameter, we do not need to specify the number of clusters before clustering. But future research should focus on how the parameter

changes for different training images and MPS settings. Also, the computational costs involved with this clustering method can be problematic for large sets of realisations. Future research can look into optimising the clustering algorithm specifically for this problem.

The average of the first PCA component linearly correlates with the known verbatim percentage in the synthetic dataset. When looking at the shape of the first component one can see that it is always radial. This means that the average first component is just the sum of the pixels around a verbatim pixel, discounting far away pixels. This results in a combination of the number of verbatim pixels and the density in their neighbourhood. The average first component in the QS-stones dataset shows a negative correlation with the k parameter, agreeing with our findings above. The second and third components show no direct correlation to the k-parameter value. But future research can look at the distribution of these PCA components as they might hold information about the shape, direction and distance of the verbatim copy in a dataset.

The multivariate plots provide a useful tool for analysing verbatim copy under the two QS parameters. The PCA metric holds more information than the V1 metric. The cluster metric was too computationally expensive to run for every parameter combination, but it might still hold valuable information.

The problem of translating verbatim detection to a data-science or machine-learning problem is that there is no labelled data, no right or wrong. In this thesis, we constructed a synthetic dataset with a known verbatim copy degree. We propose that future research focuses on building realistic synthetic verbatim copy data sets of high resolution. This can then be used in a *Common Task Framework* (CTF) setting (Donoho, 2017). CTF is the basis for a data-science competition. A publicly available labelled dataset is shared, participants try to train a model and predict on an unlabelled testing dataset, and a general committee then scores the results using the labelled testing dataset.

6 Conclusion

The *index coherence map* (ICM) allows for precise quantification of the verbatim copy in synthetic data through various metrics. In contrast with previous research, where verbatim copy was quantified using the realisation image, the ICM verbatim metrics allow for direct quantification of the verbatim copy. The ICM verbatim metrics provide a tool to tune and benchmark the *Quick Sampling* (QS) and related algorithms.

A sliding window metric was used to transform the ICM into the verbatim density map. This shows a direct visual representation of where verbatim copy is in the realisation. Through hierarchical clustering, the verbatim patches were detected in the verbatim density map. The clustering method is computationally expensive and requires tuning. More work is needed to optimise the clustering algorithm and tuning.

The PCA method showed useful insights into the behaviour of verbatim copy under different QS parameters. Future research can look at the other PCA components, which possibly encode the shape and range of verbatim.

The need for a labelled verbatim copy dataset was shown. Future research is needed on creating such data sets. When they become standardised, the problem becomes more accessible to the data science and machine-learning community.

References

Abdollahifard, M. J., Mariethoz, G., & Ghavim, M. (2019). Quantitative evaluation of multiple-point simulations using image segmentation and texture descriptors. *COMPUTATIONAL GEOSCIENCES*, *23*(6), 1349–1368. https://doi.org/10.1007/s10596-019-09901-z

Bruse, J. L., Zuluaga, M. A., Khushnood, A., McLeod, K., Ntsinjana, H. N., Hsia, T.-Y., Sermesant, M., Pennec, X., Taylor, A. M., & Schievano, S. (2017). Detecting clinically meaningful shape clusters in medical image data: Metrics analysis for hierarchical clustering applied to healthy and pathological aortic arches. *IEEE Transactions on Biomedical Engineering*, *64*(10), 2373–2383. https://doi.org/10.1109/TBME.2017.2655364

Donoho, D. (2017). 50 years of data science. *Journal of Computational and Graphical Statistics*, *26*(4), 745–766. https://doi.org/10.1080/10618600.2017.1384734

Espinoza, F. A., Oliver, J. M., Wilson, B. S., & Steinberg, S. L. (2012). Using Hierarchical Clustering and Dendrograms to Quantify the Clustering of Membrane Proteins. *Bulletin of Mathematical Biology*, *74*(1), 190–211. https://doi.org/10.1007/s11538-011-9671-3

Gravey, M., & Mariethoz, G. (2020). Quicksampling v1.0: A robust and simplified pixel-based multiple-point simulation approach. *Geoscientific Model Development*, *13*(6), 2611–2630. https://doi.org/10.5194/gmd-13-2611-2020

Mariethoz, G., & Caers, J. (2014). Multiple-point geostatistics algorithms. *Multiplepoint geostatistics* (pp. 155–171). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781118662953.ch9

Mariethoz, G., Renard, P., & Straubhaar, J. (2010). The direct sampling method to perform multiple-point geostatistical simulations. *Water Resources Research*, *46*(11). https://doi.org/10.1029/2008WR007621

Murtagh, F., & Contreras, P. (2017). Algorithms for hierarchical clustering: An overview, ii. *WIREs Data Mining and Knowledge Discovery*, *7*(6), e1219. https://doi.org/10.1002/widm.1219

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 559–572. https://doi.org/10.1080/14786440109462720

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

Wang, H., Treble, P., Baker, A., Rich, A. M., Bhattacharyya, S., Oriani, F., Akter, R., Chinu, K., Wainwright, I., & Marjo, C. E. (2022). Sulphur variations in annually layered stalagmites using benchtop micro-xrf. *Spectrochimica Acta Part B: Atomic Spectroscopy*, *189*, 106366. https://doi.org/10.1016/j.sab.2022.106366

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, *58*(301), 236–244. https://doi.org/10.1080/01621459.1963.10500845

Yin, G., Mariethoz, G., & McCabe, M. F. (2017). Gap-filling of landsat 7 imagery using the direct sampling method. *Remote Sensing*, *9*(1). https://doi.org/10.3390/rs9010012

Zakeri, F., & Mariethoz, G. (2021). A review of geostatistical simulation models applied to satellite remote sensing: Methods and applications. *Remote*

Sensing of Environment, *259*, 112381. https://doi.org/10.1016/j.rse.2021.112381

Zhang, T., & Du, Y. (2012). Reconstructing porous media using mps. In F. L. Wang, J. Lei, R. W. H. Lau, & J. Zhang (Eds.), *Multimedia and signal processing* (pp. 341–348). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-35286-7_43