

Automatic Choice of the Threshold of a Grain Filter via Galton–Watson Trees: Application to Granite Cracks Detection

**Romain Abraham, Maitine Bergounioux
& Pierre Debs**

Journal of Mathematical Imaging and Vision

ISSN 0924-9907

J Math Imaging Vis

DOI 10.1007/s10851-017-0743-3



Your article is protected by copyright and all rights are held exclusively by Springer Science+Business Media, LLC. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your article, please use the accepted manuscript version for posting on your own website. You may further deposit the accepted manuscript version in any repository, provided it is only made publicly available 12 months after official publication or later and provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be accompanied by the following text: "The final publication is available at link.springer.com".

Automatic Choice of the Threshold of a Grain Filter via Galton–Watson Trees: Application to Granite Cracks Detection

Romain Abraham¹ · Maitine Bergounioux¹  · Pierre Debs¹

Received: 2 September 2016 / Accepted: 5 June 2017
© Springer Science+Business Media, LLC 2017

Abstract The goal of this paper is the presentation of a post-processing method allowing to remove impulse noise in binary images, while preserving thin structures. We use a grain filter. We propose a method to automatically determine the required threshold using Galton–Watson processes. We present numerical results and a complete analysis on a synthetic image. We end the numerical section considering a specific application to granite samples crack detection: Here we deal with X-tomography images that have been binarized via preprocessing techniques and we want to remove residual impulse noise while keeping cracks and micro-cracks structure.

Keywords Image processing · Galton–Watson · Grain filters · Cracks

Mathematics Subject Classification 60J80 · 68U10 · 94A12

1 Introduction

A challenging issue in imaging processing is the identification of thin structures. It may be blood vessels, leaf veins or streets in a satellite or aerial image. From a mathemat-

ical point of view, these structures have a zero Lebesgue measure and we have to use the Hausdorff measure instead. From a practical point of view, these structures may be not viewed once the discretization has been done. There are many methods to achieve this goal both from theoretical and numerical points of view (as the morphological skeletonization for example). However, they may be not very satisfactory since the images are very often corrupted with a high level of noise that, in addition, may not be Gaussian.

The goal of this paper is not the presentation of an image processing model that would identify the structures better than the existing ones but rather to give a *post-processing* method allowing to remove any residual noise effect. Therefore, we do not report on thin structures recovering techniques and do not quote the related works.

The example we have in mind is the recovery of cracks inside a granite block. More precisely, the sample has been imaged¹ via X-tomography process and we have to deal with a 3D image whose size is $1287 \times 1287 \times 99$. For more details on the geological context, one can refer to [14, 21]. Figure 1 shows a 2D slice of such a stack. The right-hand side image is the result of a preprocessing method that we briefly present in Sect. 3 without any focus on it; it allows to get rid of contours that are not cracks. However, there is a residual impulse noise that we would like to remove while keeping cracks, especially micro-cracks.

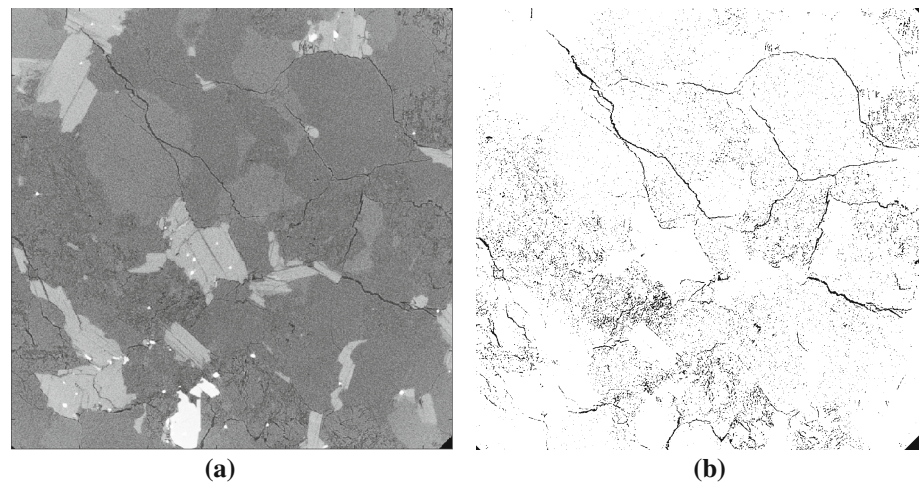
There are few methods devoted to crack identification in geological context. In biological context (angiography or X-ray imaging for example), mathematical morphology (skeletonization) is often used. One can mention the use of fuzzy measures in [12] as well. In material context, there are many works that focus on cracks in roads or concrete. Differ-

✉ Maitine Bergounioux
maitine.bergounioux@univ-orleans.fr
Romain Abraham
romain.abraham@univ-orleans.fr
Pierre Debs
pierre.debs@univ-orleans.fr

¹ Laboratoire MAPMO, CNRS, UMR 7349, Fédération Denis Poisson, FR 2964, Université d'Orléans, B.P. 6759, 45067 Orléans Cedex 2, France

¹ We thank Olivier Rozenbaum, ISTO, Université d'Orléans, CNRS, BRGM.

Fig. 1 Original and binarized 2D slice of a 3D granite sample. **a** Original slice. **b** Binarized slice



ent tools are used as statistical ones (co-occurrence matrices in [15] and anisotropic diffusion with region linkage in [17]), wavelets [23], geodesic contours [6], multiscale approach [7] together with Markov modeling or image-based percolation models [26] and dealing with brightness and connectivity as in [16].

To address this problem, we focus here on a *grain filter*. This filter, introduced in [25] in the framework of mathematical morphology and then generalized in [13], consists in removing all the connected components of a binary image (or of the level sets for a gray-level image) of size smaller than a fixed threshold. It is mainly used for image denoising but also for feature extraction, see [20]. We refer to [5] for more details on this filter. In what follows, we suppose that we have a binary image containing features we want to extract (namely thin structures as cracks). The image is corrupted by an impulse noise where every pixel is 1 with probability p and 0 with probability $1 - p$ independently of the others. One feature that allows to distinguish cracks from the noise is that the cracks size is much larger than the connected components (that we define in the sequel) the noise may create. Therefore, a grain filter can be used in order to suppress the small connected components and keep the ones whose size is greater than some threshold a_0 . The aim of this paper is to give a method to compute the threshold of the grain filter via a statistical hypothesis testing. This method enters the large class of a contrario frameworks whose ideas first appeared in [24] and which is now widely used since the pioneer works [9, 27]. We refer to the book [10] for a complete review of these methods. The general idea of the a contrario method is to compute the probability of appearance of the looked after feature in a pure noise image. If this probability is very small (less than some fixed ε), the feature is said to be “meaningful,” i.e., it cannot appear “accidentally” and thus can be viewed as an important characteristic of the studied image. As explained in [8], the parameter ε is a risk probability in the sense of statistical testing. A typical choice of ε is the

inverse of the number of connected components since, by independence, the mean number of false positives would be 1 in this case.

Throughout the paper, we consider that the image contains only an impulse noise with parameter p (hypothesis \mathcal{H}_0). Under \mathcal{H}_0 , we denote by \mathcal{C}_0 the connected component of the image that contains a given point, say the origin $(0, 0)$, and by $\text{Card}(\mathcal{C}_0)$ its cardinal. We then fix a level $\varepsilon > 0$ for the test and find a_0 such that, under \mathcal{H}_0 ,

$$\mathbb{P}(\text{Card}(\mathcal{C}_0) \geq a_0) \leq \varepsilon.$$

Keeping the connected components of size greater than a_0 , we remove most of those due to the noise. The goal of this paper is to get an upper bound for the looked after probability and hence to automatically compute the threshold a_0 , by comparing the size of \mathcal{C}_0 with the total population size of a Galton–Watson process.

Let us stress that this problem has already been tackled in [8] where an asymptotic (as the size of the image tends to $+\infty$) of the probability $\mathbb{P}(\text{Card}(\mathcal{C}_0) \geq a_0)$ is obtained. This asymptotic contains a combinatorial term a_k (namely the number of connected components one can make with exactly k pixels) which is only known (up to $k = 47$) for the 4-connectivity. Our approach first gives an exact bound of the probability and not an approximation as in [8] (which is not very relevant as the size of our images are quite large and the approximations of [8] are sharp) and second allows to use different features to discriminate cracks from noise. In Sect. 2, we focus on the size of a given connected component for the 8-connectivity and give all the details for the proofs, but the ideas can be easily adapted to handle other definitions of connectivity, the same problem for 3D images or to use the diameter of the connected component instead of its size as developed as well.

We present the computation of the threshold in the next section, and Sect. 3 is devoted to the numerical realization (algorithms and numerical experimentation).

2 Computation of the Threshold

2.1 Notations

If $k < \ell$ are two integers, we denote by $\llbracket k, \ell \rrbracket$ the set $\{n \in \mathbb{Z}, k \leq n \leq \ell\}$. We work with the infinite norm on \mathbb{Z}^2 , i.e., if $x_1 = (i_1, j_1)$ and $x_2 = (i_2, j_2)$ are two points of \mathbb{Z}^2 , we set $\|x_1 - x_2\| = \max(|i_1 - i_2|, |j_1 - j_2|)$.

For $x \in \mathbb{Z}^2$, $\bar{B}(x)$ denotes the closed ball (with respect to the previous norm) centered at x with radius 1 and $B(x)$ is the pointed ball $\bar{B}(x) \setminus \{x\}$. The elements of $B(x)$ are called the neighbors of x .

For any finite subset A of \mathbb{Z}^2 , we denote by $\text{Card}(A)$ the cardinal of A .

Definition 1 A set $A \subset \mathbb{Z}^2$ is said to be connected if, for every $x, y \in A$, there exists a finite sequence (x_0, \dots, x_n) such that

- $x_0 = x, x_n = y$;
- $\forall i \in \llbracket 0, n \rrbracket, x_i \in A$;
- $\forall i \in \llbracket 1, n \rrbracket, x_i \in B(x_{i-1})$.

We fix a positive integer N and consider (for simplicity) the domain $\Omega = \llbracket -N, N \rrbracket^2$. A binary image on Ω will be an application $X : \Omega \mapsto \{0, 1\}$. In all the paper, we always make the confusion between the binary image X and the set $X^{-1}(1) = \{x \in \Omega, X(x) = 1\}$, which is the set of the black pixels. In particular, we talk of the connected components of the image instead of the set $X^{-1}(1)$.

2.2 Galton–Watson Process

A Galton–Watson process is a branching stochastic model introduced in 1873 by Francis Galton to describe population growth and is still the object of active research. We refer to [2] for a comprehensive exposition on this topic. We first give a formal definition of this process.

Definition 2 A Galton–Watson process is a stochastic process $(Z_n)_{n \in \mathbb{N}}$ defined as follows: Consider a probability measure ν on the set of nonnegative integers (called the offspring distribution) and a family $(\xi_i^{(n)}, n \geq 0, i \geq 1)$ of i.i.d. random variables with distribution ν . Then, the process $(Z_n)_{n \in \mathbb{N}}$ is defined recursively by

$$\begin{cases} Z_0 = 1, \\ Z_{n+1} = \sum_{i=1}^{Z_n} \xi_i^{(n)} \text{ for } n \geq 0 \end{cases}$$

with the convention that $Z_{n+1} = 0$ if $Z_n = 0$.

To give an intuitive picture of that process $(Z_n)_{n \in \mathbb{N}}$, let us describe it informally as a population evolution model. We start with a single individual at generation 0. This individual gives birth to a random number $\xi_1^{(0)}$ of offspring with distribution ν which forms generation 1. These offspring reproduce

according to the same distribution ν independently of each other and independently of the past and so on. Hence, the variable Z_n represents the population size at generation n , whereas the variable $\xi_i^{(n)}$ represents the number of offspring of the i -th individual of the n -th generation.

Of course, if the population dies out at some time n , then the population is zero for all further time: 0 is called an absorbing state for the process $(Z_n)_{n \in \mathbb{N}}$.

One of the initial questions raised by Francis Galton was the computation of the probability that the population dies out in finite time. This probability is called the *extinction probability* and is given by

$$\mathbb{P}(\exists n \in \mathbb{N}, Z_n = 0) = \lim_{n \rightarrow +\infty} \mathbb{P}(Z_n = 0).$$

To avoid trivialities, we suppose that $\nu(0) > 0$ (if not, the extinction probability is 0) and that $\nu(0) + \nu(1) < 1$ (if not, the extinction probability is clearly 1).

The answer to the question is given by the following result (see Chap. 1 Section 5 of [2])

Proposition 3 Let ξ be a random variable with distribution ν . We denote by

$$\mu = \mathbb{E}[\xi] = \sum_{k=0}^{+\infty} k \nu(k)$$

the mean number of offspring and by

$$F(s) = \mathbb{E}[s^\xi] = \sum_{k=0}^{+\infty} \nu(k) s^k \text{ for } s \in [-1, 1],$$

the generating function of ν .

Then, the extinction probability of the process $(Z_n)_{n \geq 0}$ is given by the smallest nonnegative root q of the equation $s = F(s)$.

In particular, if $\mu \leq 1$, $q = 1$ (extinction arises almost surely), whereas if $\mu > 1$, then $q < 1$ (the population has a positive probability for living forever).

The offspring distribution ν and the Galton–Watson process $(Z_n)_{n \geq 0}$ are called subcritical (resp. critical, supercritical) if $\mu < 1$ (resp. $\mu = 1$, $\mu > 1$).

2.3 Stochastic Domination

Let m be the number of neighbors of one pixel. With our definition of $B(x)$, we have $m = 8$, but the method remains valid for a general neighborhood. Let $p \in (0, 1)$ and $(Z_n)_{n \geq 0}$ a Galton–Watson process with the binomial $\mathcal{B}(m, p)$ distribution as offspring distribution ν , that is, for $0 \leq k \leq m$,

$$\nu(k) = \binom{m}{k} p^k (1-p)^{m-k}.$$

According to Proposition 3, in order to obtain a subcritical Galton–Watson process, we assume that $mp < 1$. We denote by $\sharp Z$ the total population size:

$$\sharp Z = \sum_{n=0}^{+\infty} Z_n.$$

Theorem 1 Under \mathcal{H}_0 , the size $\text{Card}(C_0)$ of the connected component C_0 (and thus of any connected component) is stochastically dominated by $\sharp Z$, that is,

$$\forall k \in \mathbb{N}, \quad \mathbb{P}(\text{Card}(C_0) \geq k) \leq \mathbb{P}(\sharp Z \geq k).$$

Proof The main ideas of the proof are taken from [18] where the size of a connected component that arises in continuum percolation is compared to the size of a Galton–Watson process with Poisson offspring distribution.

Let $(X_{(i,j)}, (i,j) \in \mathbb{Z}^2)$ be a family of i.i.d. Bernoulli random variables with parameter p . This models the noise of the image under \mathcal{H}_0 . Remark that if the stochastic domination is true for an infinite image, it is a fortiori true for an image defined on $\llbracket -N, N \rrbracket^2$. If A is a subset of \mathbb{Z}^2 , we denote by

$$X_A = \{(i,j) \in A, X_{(i,j)} = 1\}.$$

Then, $\text{Card}(X_A)$ is distributed according to a binomial distribution with parameters p and $\text{Card}(A)$. Moreover, if A and B are disjoint subsets of \mathbb{Z}^2 , then X_A and X_B are independent.

In order to label the points of some subset in a unique way, we define a total order (denoted by \leq) on \mathbb{Z}^2 . Let $x_1, x_2 \in \mathbb{Z}^2$:

- if $\|x_1\| < \|x_2\|$, then we set $x_1 \leq x_2$.
- If $\|x_1\| = \|x_2\| = n$, we label the $8n$ points of norm n as shown in Fig. 2

and we say that $x_1 \leq x_2$ if the label of x_1 is less than the label of x_2 .

We first construct the connected component C_0 recursively. The process is illustrated in Fig. 3.

We define a sequence $(\tilde{W}_n)_{n \geq 0}$ of subsets of \mathbb{Z}^2 such that $C_0 = \bigcup_{n \geq 0} \tilde{W}_n$ in the following way:

- $\tilde{W}_0 = \{(0,0)\}$.
- Suppose that $\tilde{W}_0, \dots, \tilde{W}_n$ are constructed with $\tilde{W}_n = \{x_1, \dots, x_k\}$ labeled in increasing order. Then, we set

$$\tilde{W}_{n+1} = \bigcup_{\ell=1}^k X_{C_\ell}$$

where

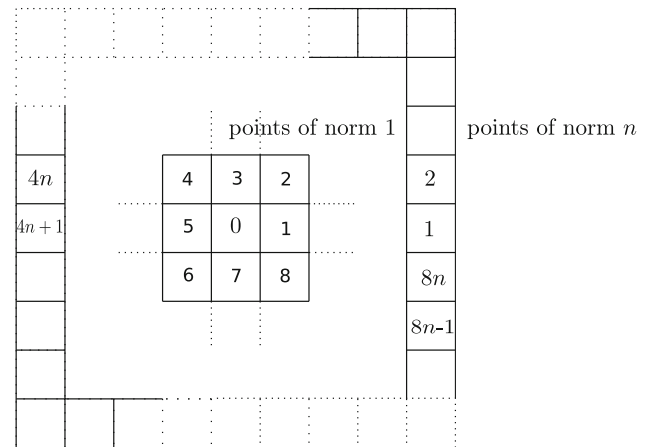


Fig. 2 Labeling of points of norm n

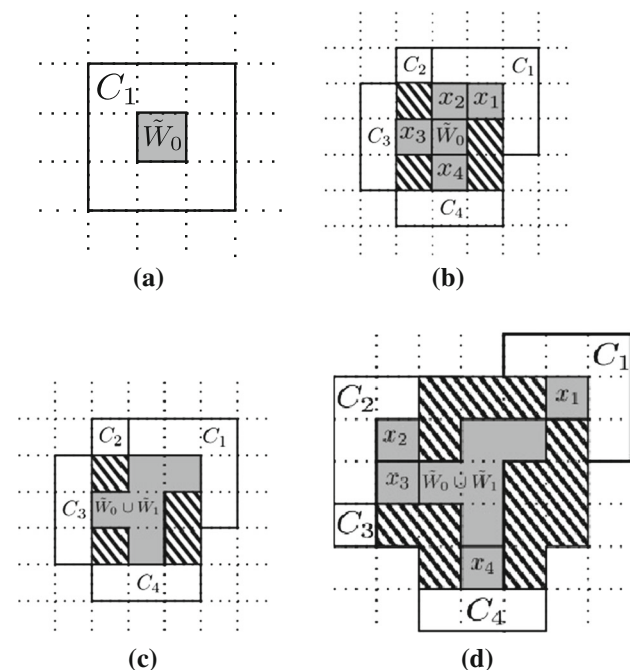


Fig. 3 Generation of the sequence W_n . **a** Initial state. **b** Conditionally on \tilde{W}_0 , $\tilde{W}_1 = \{(1,1), (0,1), (-1,0), (0,-1)\}$. **c** Conditionally on \tilde{W}_0 , $\tilde{W}_1 = \{(1,1), (0,1), (-1,0), (0,-1)\}$. **d** Conditionally on \tilde{W}_0 , $\tilde{W}_1 = \{(1,1), (0,1), (-1,0), (0,-1)\}$, $\tilde{W}_2 = \{(2,2), (-2,1), (-2,0), (0,-2)\}$

$$C_\ell = B(x_\ell) \setminus \left(\bigcup_{j=1}^{\ell-1} \bar{B}(x_j) \bigcup_{i=0}^{n-1} \bigcup_{x \in \tilde{W}_i} \bar{B}(x) \right),$$

with the convention $\bigcup_{j=1}^0 \bar{B}(x_j) = \emptyset$.

- If $\tilde{W}_n = \emptyset$, then we set $\tilde{W}_k = \emptyset$ for every $k > n$.

The set C_ℓ at step n is the set of neighbors of x_ℓ that have not been involved yet in one of the C_k -type sets constructed during the previous steps. Consequently, these sets C_ℓ are

Fig. 4 Labeling of the neighbors of a point x

4	3	2
5	x	1
6	7	8

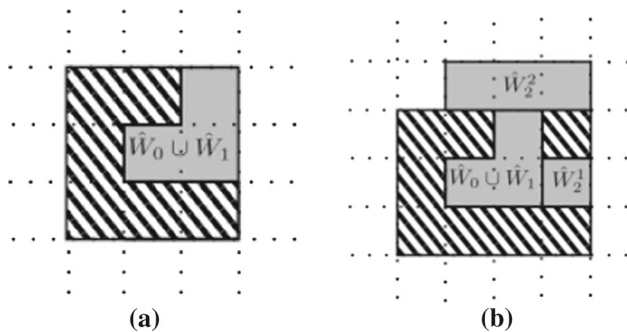


Fig. 5 Example for the two first prunings. **a** $W_1 = \{(0, 1), (1, 1)\}$. **b** $W_2 = W_1^1 \cup W_2^2$ where $W_1^1 = \{(2, 0), (1, 1), (0, 0), (0, -1)\}$ and $W_2^2 = \{(2, 1), (2, 2), (1, 2), (0, 2), (2, 0)\}$

pairwise disjoint. In a second step, we consider a branching random walk $(W_n)_{n \geq 0}$ on \mathbb{Z}^2 (we refer to [22] for more details on branching random walks). This process takes its values in the set $\bigcup_{n \geq 0} (\mathbb{Z}^2)^n$ of finite sequences of points of \mathbb{Z}^2 (with the convention $(\mathbb{Z}^2)^0 = \{\emptyset\}$) and is also defined recursively as follows:

- $W_0 = (0, 0)$.
- Let us suppose that W_n is given and is nonempty, say $W_n = (x_1^n, \dots, x_{k_n}^n)$. For every $i \leq k_n$, we consider the eight neighbors of x_i^n (ranked as shown in Fig. 4) (y_1, \dots, y_8) and consider the subsequence W_{n+1}^i formed by taking each y_j with probability p , independently of the others (and independently of the other points $x_j^n, i \neq j$). Then, W_{n+1} is the concatenation of all the subsequences $(W_{n+1}^i, 1 \leq i \leq k_n)$. Remark that a pixel may appear several times in the sequence W_{n+1} (Fig. 5).
- If $W_n = \emptyset$, then $W_{n+1} = \emptyset$.

The subsequence W_{n+1}^i may be seen as the offspring of the pixel x_i^n and $\text{Card}(W_{n+1}^i)$ is distributed according to a binomial $\mathcal{B}(8, p)$ distribution. Therefore, the process $(\text{Card}(W_{n+1}))_{n \geq 0}$ is distributed as a Galton–Watson process starting from a single individual and with offspring distribution $\mathcal{B}(8, p)$.

We now define a pruning procedure for the process $(W_n)_{n \geq 0}$ which defines a set-valued process $(\hat{W}_n)_{n \geq 0}$ as follows:

- $\hat{W}_0 = \{(0, 0)\}$.
- Suppose that we have constructed $\hat{W}_0, \hat{W}_1, \dots, \hat{W}_n$ where each set \hat{W}_i is composed of elements of W_i , with

$\hat{W}_n = \{\hat{x}_1^n, \dots, \hat{x}_k^n\}$ (ranked in increasing order). To every \hat{x}_i^n , there corresponds a point x_{ji}^n of W_n and we set

$$\hat{W}_{n+1}^i = W_{n+1}^{ji} \left(\bigcup_{\ell=1}^{i-1} \bar{B}(\hat{x}_\ell^n) \bigcup_{r=0}^{n-1} \bigcup_{x \in \hat{W}_r} \bar{B}(x) \right)$$

and then

$$\hat{W}_{n+1} = \bigcup_{i=1}^k \hat{W}_{n+1}^i.$$

- If $\hat{W}_n = \emptyset$, then $\hat{W}_{n+1} = \emptyset$.

As the process $(\hat{W}_n)_{n \geq 0}$ is distributed as the process $(\tilde{W}_n)_{n \geq 0}$ (since the recursive formulas are the same), we have for every $a \geq 0$

$$\begin{aligned} \mathbb{P}(\mathcal{C}_0 \geq a) &= \mathbb{P}\left(\text{Card}\left(\bigcup \tilde{W}_n\right) \geq a\right) \\ &= \mathbb{P}\left(\text{Card}\left(\bigcup \hat{W}_n\right) \geq a\right). \end{aligned}$$

Furthermore, it is clear by construction (pruning procedure) that $\text{Card}(\hat{W}_n) \leq \text{Card}(W_n)$ for every $n \geq 0$, which yields

$$\begin{aligned} \mathbb{P}\left(\text{Card}\left(\bigcup \hat{W}_n\right) \geq a\right) &\leq \mathbb{P}\left(\text{Card}\left(\bigcup W_n\right) \geq a\right) \\ &= \mathbb{P}(\sharp Z \geq a) \end{aligned}$$

which ends the proof. \square

Remark 1 Looking at Fig. 3, it seems that we could in fact take $m = 5$ except for the first generation of the tree where we must keep $m = 8$. The corresponding tree is not stricto sensu a Galton–Watson tree (as the offspring distribution is not always the same); however, the computations of the next subsection can probably be adapted to this case. For sake of simplicity, we take $m = 8$ in what follows to have a true Galton–Watson process.

2.4 Computation of the Threshold

Let $(Z_n)_{n \geq 0}$ be a subcritical Galton–Watson process with offspring distribution a binomial $\mathcal{B}(m, p)$. For this offspring distribution to be subcritical, recall that we must suppose that

$$mp < 1. \quad (1)$$

We need an upper bound for the probability $\mathbb{P}(\sharp Z \geq k)$. For this purpose, we use Dwass formula that relates the total population size of any subcritical (or critical) Galton–Watson process to the associated Lukasiewicz random walk, see [11]:

Proposition 4 Let $(Z_n)_{n \geq 0}$ be a critical or subcritical Galton–Watson process with offspring distribution ν . Let $(Y_n)_{n \geq 1}$ be a sequence of i.i.d. random variables with distribution ν . We set for every $n \geq 1$

$$S_n = \sum_{i=1}^n Y_i.$$

Then, we have

$$\mathbb{P}(\sharp Z = n) = \frac{1}{n} \mathbb{P}(S_n = n - 1).$$

Recall that ν is the binomial $\mathcal{B}(m, p)$ distribution; then, for every $n \geq 1$, S_n is distributed according to a binomial $\mathcal{B}(mn, p)$ distribution (as the sum of i.i.d $\mathcal{B}(m, p)$ random variables). Therefore,

$$\mathbb{P}(\sharp Z = n) = \frac{1}{n} \binom{mn}{n-1} p^{n-1} (1-p)^{mn-n+1}. \quad (2)$$

This exact formula allows us to compute the looked after threshold:

Proposition 5 We set

$$q = \frac{m^m p(1-p)^{m-1}}{(m-1)^{m-1}}, \text{ and } C = \sqrt{\left(\frac{m}{m-1}\right) \frac{e(1-p)}{pK^2} \frac{1}{1-q}} \quad (3)$$

with $K = e^{\frac{3}{2}(1-\ln \frac{3}{2})}$. Then, for every $\varepsilon > 0$, a threshold is given by

$$a_0 = \frac{\ln \frac{\varepsilon}{C}}{\ln q}, \quad (4)$$

that is, for every $a \geq a_0$, we have

$$\mathbb{P}(\sharp Z \geq a) \leq \varepsilon.$$

Proof By Proposition 4 and Eq. (2), we get for every $a > 0$

$$\begin{aligned} \mathbb{P}(\sharp Z \geq a) &= \sum_{n \geq a} \mathbb{P}(\sharp Z = n) \\ &\leq \sum_{n \geq a} \binom{mn}{n} p^{n-1} (1-p)^{mn-n+1}, \end{aligned}$$

$$\text{since } \frac{1}{n} \binom{mn}{n-1} \leq \binom{mn}{n}. \quad \square$$

We now use the following bounds for the factorial function whose proof is postponed to the end of this section:

Lemma 6 For every positive integer n , we have

$$Kn^{n+\frac{1}{2}}e^{-n} \leq n! \leq n^{n+\frac{1}{2}}e^{-n}e \quad (5)$$

with $K = \exp\left(\frac{3}{2}(1 - \ln \frac{3}{2})\right)$.

Using these bounds in the binomial coefficient, we get

$$\begin{aligned} \mathbb{P}(\sharp Z \geq a) &\leq \sum_{n \geq a} \frac{(mn)^{mn+\frac{1}{2}}e^{-mn}e}{K^2 n^{n+\frac{1}{2}}e^{-n}(mn-n)^{mn-n+\frac{1}{2}}e^{-mn+n}} \\ &= p^{n-1}(1-p)^{mn-n+1} \leq \left(\frac{m}{m-1}\right)^{\frac{1}{2}} \frac{e(1-p)}{pK^2} \\ &\quad \sum_{n \geq a} \left(\frac{m^m p(1-p)^{m-1}}{(m-1)^{m-1}}\right)^n n^{-\frac{1}{2}} \\ &\leq \left(\frac{m}{m-1}\right)^{\frac{1}{2}} \frac{e(1-p)}{pK^2} \sum_{n \geq a} q^n. \end{aligned}$$

Let us remark that the function $x \mapsto x(1-x)^{m-1}$ reaches its maximum on the interval $[0, 1]$ at $x = \frac{1}{m}$. As we supposed that $p < \frac{1}{m}$,

$$q < \frac{m^m \frac{1}{m} (1 - \frac{1}{m})^{m-1}}{(m-1)^{m-1}} = 1.$$

The previous sum is then finite and

$$\mathbb{P}(\sharp Z \geq a) \leq \left(\frac{m}{m-1}\right)^{\frac{1}{2}} \frac{e(1-p)}{pK^2} \frac{q^a}{1-q} := Cq^a.$$

Therefore, (recall that $q < 1$ which gives $\ln q < 0$),

$$a \geq \frac{\ln \frac{\varepsilon}{C}}{\ln q} \Rightarrow Cq^a \leq \varepsilon \Rightarrow \mathbb{P}(\sharp Z \geq a) \leq \varepsilon. \quad \square$$

Proof (Proof of Lemma 6) By the trapezoidal rule, as the logarithm function is concave, we have for every positive integer k

$$\frac{\ln(k+1) + \ln k}{2} \leq \int_k^{k+1} \ln t \, dt.$$

This implies

$$\begin{aligned} \sum_{k=1}^{n-1} \frac{\ln(k+1) + \ln k}{2} &\leq \int_1^n \ln t \, dt \Leftrightarrow \ln n! - \frac{\ln n}{2} \\ &\leq n \ln n - n + 1 \\ &\Leftrightarrow \ln n! \leq \left(n + \frac{1}{2}\right) \ln n - n + 1 \\ &\Leftrightarrow n! \leq n^{n+\frac{1}{2}}e^{-n}e \end{aligned}$$

which gives the upper bound.

Still using the concavity of the logarithm function, its graph is always below its tangent which is given at point k by the equation

$$y = \frac{1}{k}x - 1 + \ln k.$$

This yields

$$\begin{aligned} \int_{k-\frac{1}{2}}^{k+\frac{1}{2}} \ln t \, dt &\leq \frac{\ln k + \frac{1}{2k} + \ln k - \frac{1}{2k}}{2} = \ln k \\ &\Rightarrow \int_{\frac{3}{2}}^{n+\frac{1}{2}} \ln t \, dt \leq \sum_{k=2}^n \ln k = \ln n! \\ &\Rightarrow \left(n + \frac{1}{2}\right) \ln \left(n + \frac{1}{2}\right) - n \\ &\quad - \frac{3}{2} \ln \frac{3}{2} + 1 \leq \ln n! \\ &\Rightarrow \left(n + \frac{1}{2}\right) \ln n + \left(n + \frac{1}{2}\right) \\ &\quad \ln \left(1 + \frac{1}{2n}\right) - n - \frac{3}{2} \ln \frac{3}{2} + 1 \leq \ln n! \end{aligned}$$

To get the lower bound, it suffices to check that

$$\left(n + \frac{1}{2}\right) \ln \left(1 + \frac{1}{2n}\right) \geq \frac{1}{2}$$

which comes immediately from the inequality

$$\forall x \geq 0, \ln(1+x) \geq x - \frac{x^2}{2}.$$

2.5 Other Cases

We gave all the details of the proof with the 8-connectivity for 2D images, but the arguments of the previous subsections can be adapted to many other situations. We present some of them without detailing the proofs of the stochastic domination.

2.5.1 Other Definitions of Connectivity

We focused on the 8-connectivity, but the same arguments work for the 4-connectivity (taking $m = 4$). Numerical tests are performed in the next section to compare the performances of each connectivity.

We can also extend this work to *epsilon connectivity* (see [19] for instance) where two pixels are said to be connected if their ℓ^∞ -distance is less than k . For instance, for $k = 2$, this allows to fill small gaps between cracks that may be created because of the noise and/or the thickness of the cracks. All the previous computations remain valid taking $m = 24$ since a neighborhood of a pixel is now a 5×5 square.

2.5.2 3D Images

We also may extend the results to 3D stacks. Recall that we consider a 3D stack whose size is $1287 \times 1287 \times 99$. If we consider the nearest neighbors of a pixel to define connectivity, the neighborhood of a pixel is now a cube of size 3 and the results apply with $m = 26$. Some numerical experiments are presented for such a case in Sect. 3.3.

2.5.3 Diameter of the Connected Component

Let us go back to 2D-images and 8-connectivity. Let C_0 be the connected component of the image that contains 0. For $x, y \in C_0$, we set $\Gamma_{x,y}$ as the set of connected paths that go from x to y remaining inside C_0 . For $\gamma \in \Gamma_{x,y}$, we denote by $\ell(\gamma)$ its length, i.e., the number of pixels γ is composed with. Finally, we define the diameter of C_0 by

$$\text{diam}(C_0) = \max_{x,y \in C_0} \min_{\gamma \in \Gamma_{x,y}} \ell(\gamma).$$

This diameter can be used to discriminate cracks from noise. It also discriminates cracks (that have a quite large diameter) from ball-like connected components with the same number of pixels. Moreover our method still applies since we have, for every positive integer a ,

$$\mathbb{P}(\text{diam}(C_0) \geq 2a) \leq \mathbb{P}(Z_a > 0)$$

where $(Z_n)_{n \geq 0}$ is a Galton–Watson process with binomial $\mathcal{B}(m, p)$ offspring distribution with $m = 8$. Indeed, if $\text{diam}(C_0) \geq 2a$, there exists two points $x, y \in C_0$ whose shortest path between them has length greater than $2a$. Then, at least one of the shortest paths from the origin to x or y is greater than a and the Galton–Watson process needs at least a generations to attain this point.

To compute the looked after threshold, we use the following lemma.

Lemma 7 *Let $(Z_n)_{n \geq 0}$ be a Galton–Watson process with binomial $\mathcal{B}(m, p)$ offspring distribution. Then, for every positive integer n , we have*

$$\mathbb{P}(Z_n > 0) \leq (mp)^n.$$

Proof Let us denote by $F(z)$ the generating function of the binomial $\mathcal{B}(m, p)$ distribution:

$$F(z) = (pz + 1 - p)^m.$$

Let us also denote for every $n \geq 0$ by $F_n(z)$ the generating function of Z_n . Then, we have the classical recursive formula:

$$\forall n \geq 0, \quad F_{n+1}(z) = F(F_n(z)) = F_n(F(z)).$$

Moreover, we have $\mathbb{P}(Z_n > 0) = 1 - F_n(0)$ and $\mathbb{P}(Z_0 > 0) = 1$. Therefore, we get

$$\begin{aligned}\mathbb{P}(Z_{n+1} > 0) &= 1 - F_{n+1}(0) \\ &= 1 - (1 - p(1 - F_n(0)))^m \\ &\leq mp(1 - F_n(0)) = mp \mathbb{P}(Z_n > 0).\end{aligned}$$

The result follows from an easy induction. \square

In that case, the threshold satisfies

$$a_0 = 2 \frac{\log \varepsilon}{\log(mp)}. \quad (6)$$

This second threshold gives a geometric information, since we only keep structures with large diameters. In our case, this favors the cracks that have anisotropic structure, with respect to the noise that can be viewed as small isotropic elements. Numerical tests are reported in Sect. 3.1.4.

3 Application to Crack Detection

As already mentioned, the application we have in mind is the identification of the cracks that appear in granite samples. Here, we are interested in denoising as a post-processing final step in for a complete crack identification method (based on a variational formulation) that we briefly describe in the sequel. That is why we focus on binary images (though the method works with gray-level images as well) and the impulse noise that has to be deleted. The *grain filter* is a very efficient tool to achieve this goal: That is why we decided to focus on some improvements and automatic thresholding.

The numerical experimentation has been performed with MATLAB. We do not report on the computation time (we used different machines, and the codes are not optimized, especially for 3D cases). However, we may empha-

size that the most time-consuming step is the computation of the size (or geodesic diameter) of connected components. We used a built in MATLAB function to detect the connected components, namely `bwlabel` and the `inGeodesic` toolbox of <https://github.com/dlegland/matImage/wiki/imGeodesics>. However, this step should/can be improved with ad hoc methods.

3.1 Tests with a 2D Synthetic Black and White Image

We first perform numerical tests on a synthetic image that has been built with the image as shown in Fig. 1a. It has been manually modified to get an image close to real data. This will be considered as the ground truth, and we add impulse noise whose level is p , for $p \in [0.01, 0.35]$. The image is presented in Fig. 6.

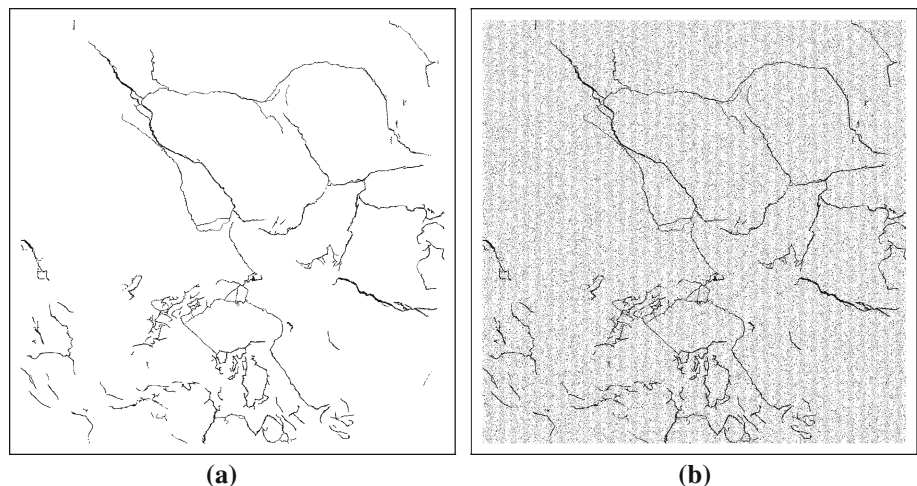
We assume that m (the number of neighbors in the chosen neighborhood) and ε (the test level) are given. A typical choice for m is $m = 3^2 - 1 = 8$ for 2D images (though we shall test $m = 4$ as well) and $m = 3^3 - 1 = 26$ for 3D images (using $3 \times 3 \times 3$ neighborhoods). The parameter ε is chosen *small enough*. We will make this precise in the sequel, when we study the sensitivity with respect to ε .

Let us mention that we should only consider $p < 0.125$ since we have chosen $m = 8$ and we have to fit the assumption (1). However, we have added noise up to level 0.35. In the case where p is known (it is the noise level), the algorithm writes

Algorithm 0 Case where p is known

- Given data:** black and white (noisy) image u , ε and m ; p is known.
1. Compute the threshold value a_0 with (3) and (4).
 2. Find the connected components of u (and their size)
 3. Suppress the connected components of u whose size is less than a_0 to get the significant contours u^*
-

Fig. 6 Ground truth and noisy image. **a** Ground truth (synthetic image). **b** Impulse noise with $p = 0.15$



However, in the real data case, the probability p is unknown and its estimate is an important issue. We may first estimate it using the number of black pixels of the binary image divided by its total size M : $p \simeq \frac{\text{Size}(u=1)}{M}$. This rough method gives an upper bound for p that is too large as an estimate. Indeed, the number of black pixels includes both the noise and the cracks, whereas we only want to count the noisy pixels. So we adopt an iterative strategy to perform successive updates of p once we have performed a more accurate selection of cracks pixels.

Algorithm 1 Algorithm with iterative update of p

Given data: black and white (noisy) image u , ε and m .
Find the connected components of u (and their size)
0. Initialization: $k = 1$, $u_0 = 0$.
1. Iteration k
Compute $u - u_k$ and
$$p_k = \frac{\text{Size}((u - u_k) = 1)}{M}.$$

Compute the threshold value a_k with (3) and (4).
Suppress the connected components of u whose size is less than a_k :
We get u_{k+1}
2. Stopping criterion. Stop when the sequence (p_k) becomes stationary.

We report in Table 1 the values of the computed probability p and the threshold with and without this strategy with respect to the noise level. This corresponds to the computed values at iteration 1 of Algorithm 1 and at the last iteration Algorithm 1. We have set the maximal number of iterations at 20 because we always observe that the convergence is

achieved before iteration 10. The cases where the maximal iterations number is attained correspond to cases where the computed probability does not satisfy (1) which makes no sense.

We can also see in Fig. 7 the behavior of the computed threshold with respect to the noise level for the different approaches. We clearly see the change of behavior for a noise level greater than 0.10. Indeed, the *computed* probability is then not far from the critical value ($0.125 = 1/m$) and the theoretical assumption (1) is no longer fulfilled. We have reported these threshold values in Table 1 to enlighten the effect to the violation of the assumption from a numerical point of view. To overcome this difficulty, we may change the connectivity by setting $m = 4$ which allows the probability p to be less than 0.25. We shall compare the two choices in the sequel.

Figure 8 shows the computed solution for different noise levels. If we compare with the ground truth, we get a quantitative estimate of the performance, with δ_1 which is the ℓ^1 norm of the difference between the ground truth and the solution. Since the images are binary, δ_1 is in fact the ℓ^0 norm of the difference, that is, the number of pixels which are different.

3.1.1 Sensitivity with Respect to ε

We now investigate the sensitivity of the solution computation with respect to ε . As already mentioned a typical choice of ε is the inverse of the number of connected components as, by independence, the mean number of false positives would be 1 in this case. We recall that the threshold is proportional to $\ln \varepsilon$ (Proposition 5).

Table 1 Computed probability and threshold with respect to the noise level, with Algorithm 1 with (last iteration) and without (first iteration) the adaptive strategy $m = 8$, $\varepsilon = 10^{-6}$

Noise level	Computed probability iteration 1	Computed probability last iteration	Threshold (# pixels) iteration 1	Threshold (# pixels) last iteration	Number of iterations in Algorithm 1
0.01	0.032	0.0096	26	11	3
0.02	0.042	0.019	38	16	3
0.03	0.051	0.029	54	23	3
0.04	0.062	0.038	79	33	3
0.05	0.071	0.048	121	48	4
0.06	0.081	0.058	201	71	4
0.07	0.091	0.068	354	107	4
0.08	0.1	0.080	766	191	4
0.09	0.11	0.093	2266	435	5
0.10	0.1203	0.1203	26,284	26,284	2
0.105	0.1249	0.1249	9.2714 e+07	9.2714 e+07	2
0.11	0.13	0.130	2347	2347	2
0.12	0.14	0.131	2648	1633	20 (Max)
0.125	0.14	0.131	1521	1584	20 (Max)

Fig. 7 Evolution of the threshold with respect to the noise level in the range $[0, 0.11]$ — X axis is the noise level and Y -axis is the threshold value (number of pixels) with \log_{10} -scale. The continuous line corresponds to the case where p is computed once (iteration 1) and the dotted curve to the computed threshold with adaptive strategy (last iteration of Algorithm 1)— $m = 8$, $\varepsilon = 10^{-6}$

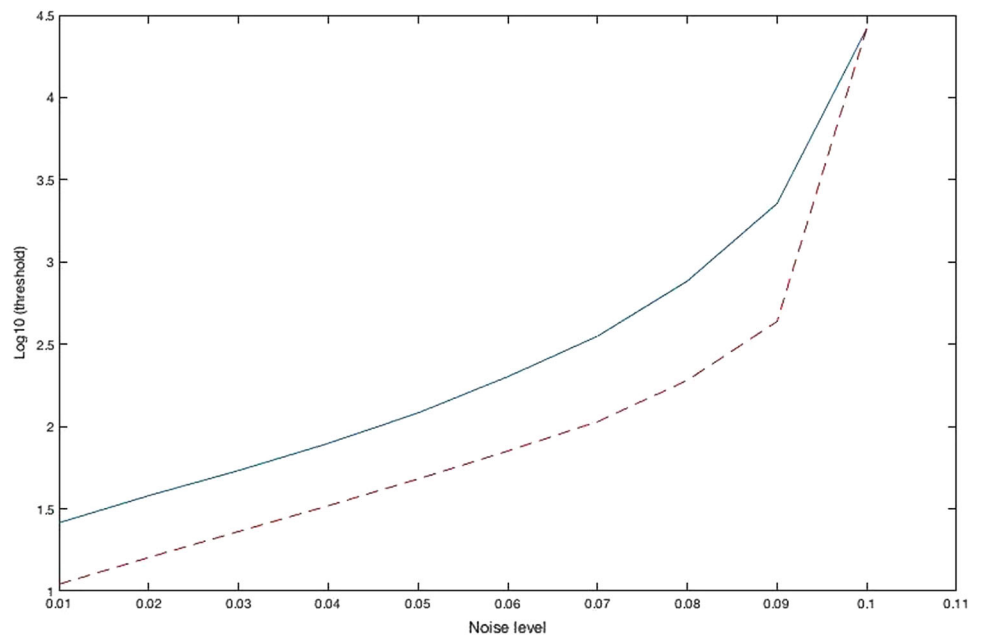


Fig. 8 Noisy image, solution given by Algorithm 1 and difference with the ground truth, for different noise levels— $m = 8$, $\varepsilon = 10^{-6}$ —the differences are presented as negative images for a better insight (white is equal to 0 and black is maximum). The noise level 0.1 corresponds to a computed probability $p = 0.12$ quite close to the critical value. **a** Level noise=0.05. **b** Computed solution. **c** Difference with the data. **d** Difference with the ground truth, $\delta_1 = 3688$. **e** Level noise=0.1. **f** Computed solution. **g** Difference with the data. **h** Difference with the ground truth, $\delta_1 = 18107$

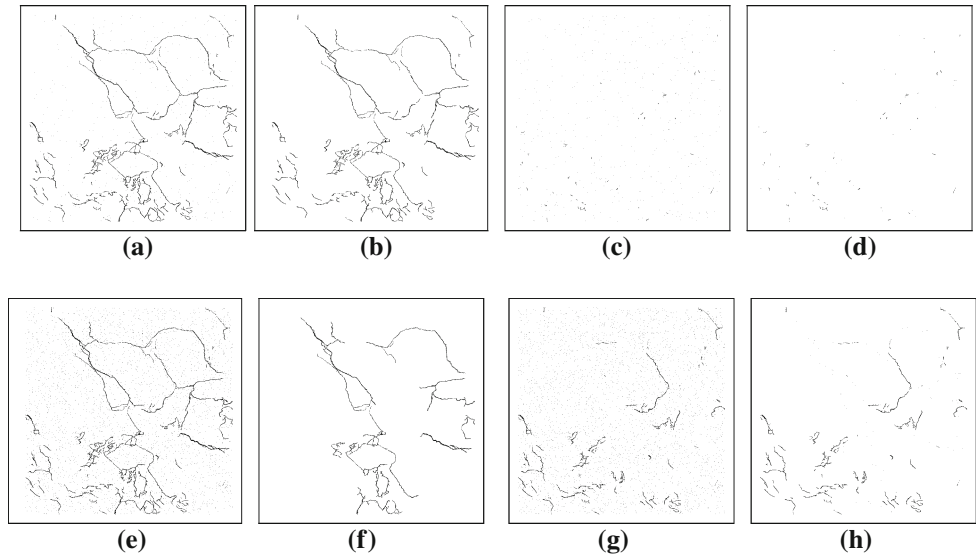
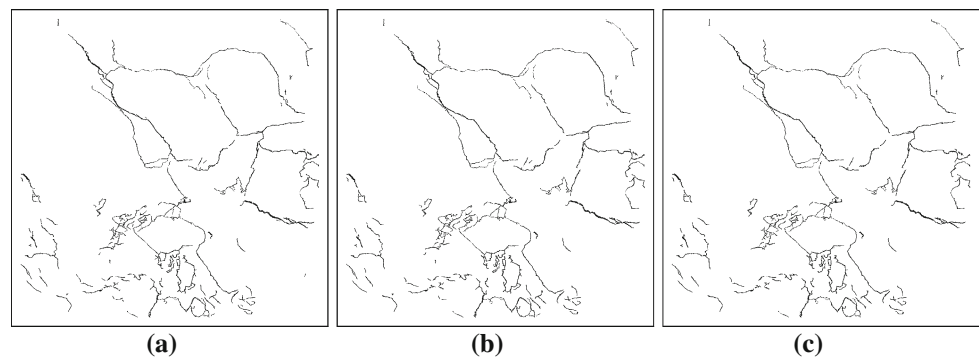


Fig. 9 Examples of solutions given by Algorithm 1 for $\varepsilon = 10^{-2}$, 10^{-4} , 10^{-6} —noise=0.05— δ_1 is the ℓ^1 norm of the difference between the computed solution and the ground truth. **a** $\varepsilon = 10^{-2}$, $\delta_1 = 2869$. **b** $\varepsilon = 10^{-4}$, $\delta_1 = 3279$. **c** $\varepsilon = 10^{-6}$, $\delta_1 = 3688$



This comparison shows that the role of ε has not a big influence (Fig. 9 and Table 2). Indeed, as the image size is $1.653 \cdot 10^6$, the relative value of δ_1 varies from $1.73 \cdot 10^{-3}$ to

$2.23 \cdot 10^{-3}$. In the sequel, we choose $\varepsilon \simeq 1/M$ where M is the size of u that is, roughly speaking, $\varepsilon = 10^{-6}$.

Table 2 Computed probability and threshold with respect to the noise level, with the adaptive strategy Algorithm 1 and different values of $\varepsilon - m = 8$

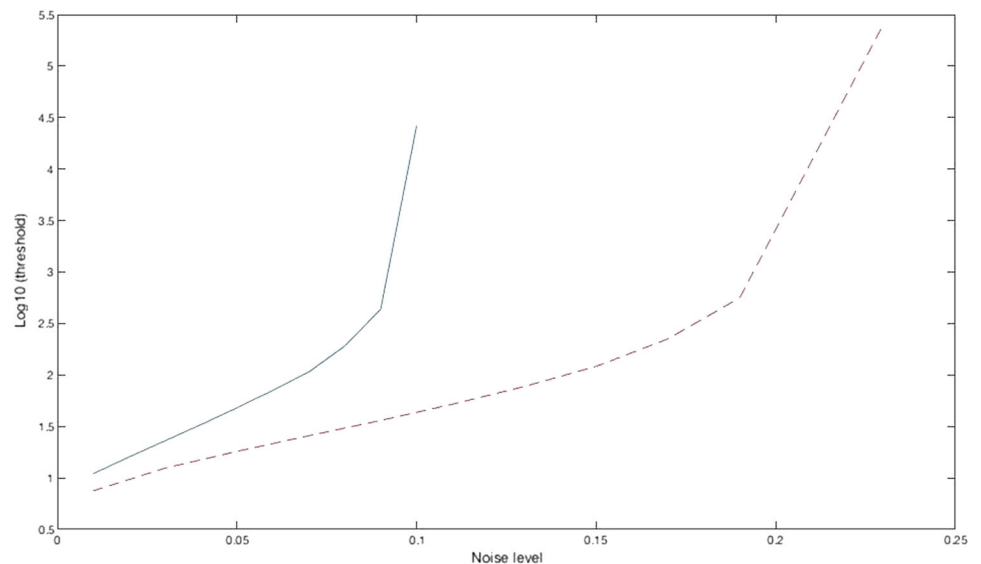
Noise level	Probability			Threshold (# pixels)		
	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-6}$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-6}$
0.01	0.0095	0.0095	0.0096	5	8	11
0.03	0.0287	0.0288	0.0290	11	17	24
0.05	0.0477	0.0480	0.0483	22	35	48
0.10	0.1203	0.1203	0.1203	158	215	271
0.15	0.1496	0.1509	0.1517	476	635	792

Note that the last line is not meaningful since in that case the computed probability does not satisfy (1)

Table 3 Computed probability and threshold with respect to the noise level, with Algorithm 1 for $m = 4$ and $m = 8$, $\varepsilon = 10^{-6}$

Noise level	$m = 8$		$m = 4$	
	Computed probability	Threshold	Computed probability	Threshold
0.01	0.0096	11	0.01	7
0.03	0.029	23	0.029	12
0.05	0.048	48	0.048	18
0.07	0.068	107	0.068	26
0.09	0.093	435	0.088	36
0.11	—	—	0.108	52
0.15	—	—	0.148	121
0.19	—	—	0.198	563

Fig. 10 Evolution of the threshold with Algorithm 1 with respect to the noise level—X axis is the noise level and Y-axis is the threshold value (number of pixels) with log10- scale. The continuous line corresponds to the case where $m = 4$ and the dotted curve to $m = 8$, $\varepsilon = 10^{-6}$



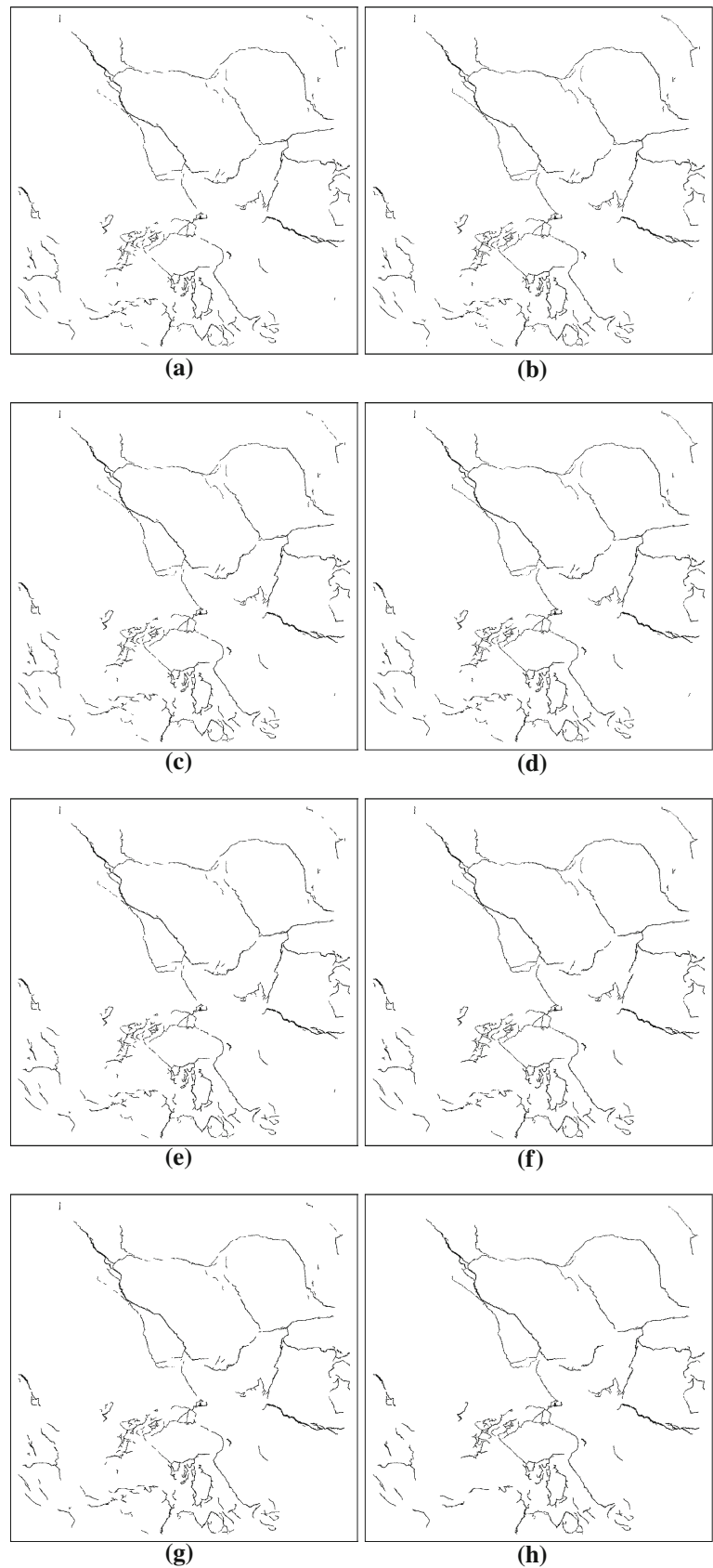
3.1.2 Sensitivity with Respect to the Connectivity

We compare here the results obtained with the two connectivities corresponding to $m = 4$ and $m = 8$. As expected, the behavior of the threshold value for $m = 4$ allows to consider higher noise levels: The computed probability has to be less than 0.25 to be consistent with assumption (1). Table 3 and Fig. 10 show the behavior of the threshold with respect to the noise level for the two connectivities.

Figure 11 presents the solutions that are obtained with Algorithm 1 for the same noisy data and $m = 4, 8$, respec-

tively. Figure 11 shows that the use of 4-connectivity gives a better result when we consider the difference with the ground truth, when the noise level becomes high. However, these results have to be handled from a qualitative and/or geometrical point of view as well. Indeed, the use of a 4-neighborhood does not take the diagonal connections into account. Therefore, thin cracks are lost and thick ones may be disconnected. However, the 4-connectivity is useful in case of high-level noise since it allows assumption (1) to be satisfied even if the noise level is high.

Fig. 11 Comparison of the solutions computed with Algorithm 1 for $m = 4$ (left) and $m = 8$ (right) and $\varepsilon = 10^{-6}$. δ_1 is the ℓ^1/ℓ^0 norm of the difference between the computed solution and the ground truth. **a** Noise: 0.01, $m = 4$, $\delta_1 = 981$. **b** Noise: 0.01, $m = 8$, $\delta_1 = 604$. **c** Noise: 0.03, $m = 4$, $\delta_1 = 1981$. **d** Noise: 0.03, $m = 8$, $\delta_1 = 1983$. **e** Noise: 0.05, $m = 4$, $\delta_1 = 3082$. **f** Noise: 0.05, $m = 8$, $\delta_1 = 3688$. **g** Noise: 0.07, $m = 4$, $\delta_1 = 4241$. **h** Noise: 0.07, $m = 8$, $\delta_1 = 6351$



3.1.3 Comparison with Other Filters and/or Other Thresholds

Though there are a lot of filters to deal with impulse noise, we limit our comparison to the median one and a morphological one, namely an opening filter with two different structural elements: a disk of radius 1 (which corresponds to the 4-connectivity) and a square 3×3 (which corresponds to the 8-connectivity). Results are reported in Fig. 12 for a noise level of 0.05. Visually speaking the solution that we get with Algorithm 1 is the best one.

Next, we compare our threshold computation with the one of [8]. Figure 13 shows the behavior of the threshold for both approaches, in the case where $\varepsilon = 10^{-2}$ and $m = 4$. We see that the threshold estimates of [8] are better (lower value), at least for high-level noise. However, we would emphasize the fact that these estimates are obtained for $m = 4$ (and we are mostly interested in 8-connectivity) and that their computations involve constants that have been computed (<https://oeis.org/A000105>) for connected components whose size is less than 47. In addition, we have to mention that our estimate

is rough and can be improved with sharper upper estimates. Our main concern was to give a method to automatically tune the threshold parameter.

3.1.4 Use of the Connected Components Diameter

As mentioned in Sect. 2.5.3, we can use Algorithm 1 and consider the geodesic diameter of connected components instead of the size. The modified algorithm writes

Algorithm 2 Use of the geodesic diameter

Given data: black and white (noisy) image u , ε and m .

Find the connected components of u and their **diameter**.

0. **Initialization:** $k = 1$, $u_0 = 0$.

1. **Iteration k**

Compute $u - u_k$ and

$$p_k = \frac{\text{Size}((u - u_k) = 1)}{M}.$$

Compute the threshold value a_k with (6).

Suppress the connected components of u whose diameter is less than a_k : We get u_{k+1}

2. **Stopping criterion.** Stop when the sequence (p_k) becomes stationary.

Fig. 12 Comparison of the solutions computed with different filters, for a noise level equal to 0.05. Here δ_1 is the ℓ^1 norm of the difference between the ground truth and the solution, that is, the number of pixels that are different. **a** Median filter 3×3 , $\delta_1 = 6371$. **b** Opening with disk of radius 1, $\delta_1 = 9900$. **c** Opening with square of size 3×3 , $\delta_1 = 21344$. **d** Algorithm 1, $m = 8$, $\varepsilon = 10^{-6}$, $\delta_1 = 3688$

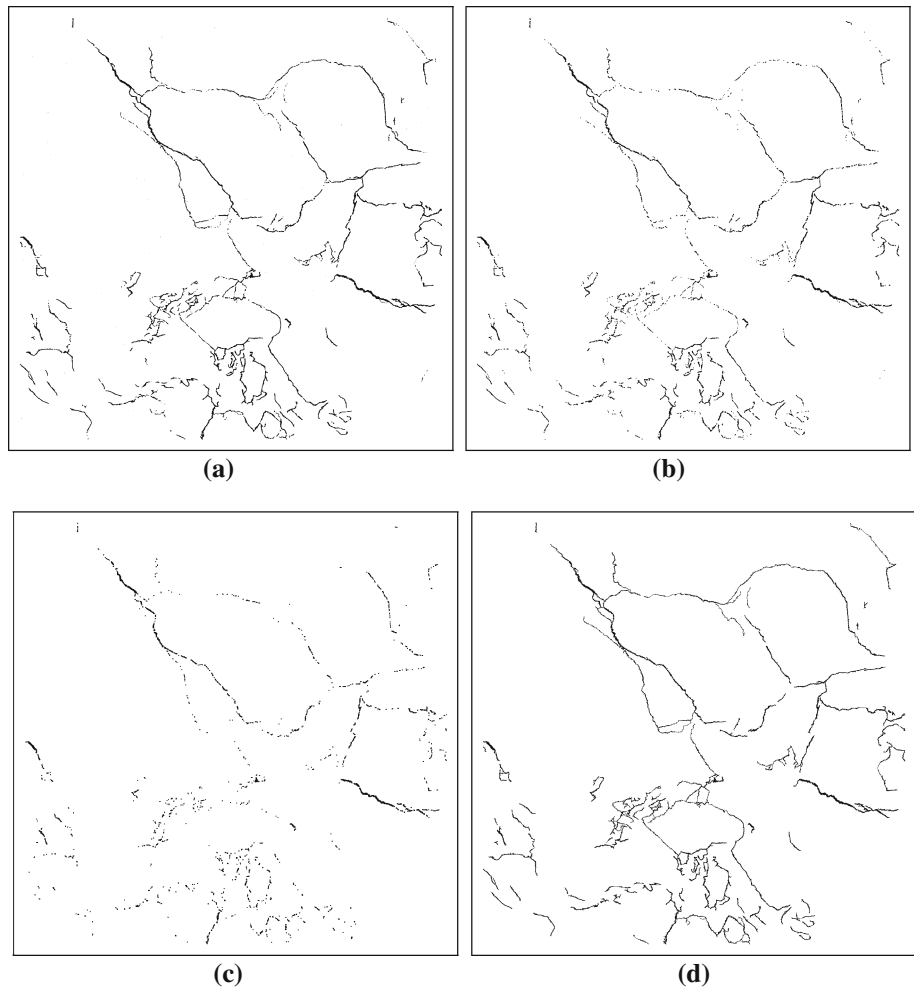


Fig. 13 Behavior of the theoretical threshold value with respect to the noise level $\varepsilon = 10^{-2}$ and $m = 4$. The continuous line is the curve obtained from [8] data, and the dotted line comes from estimates (3) and (4)

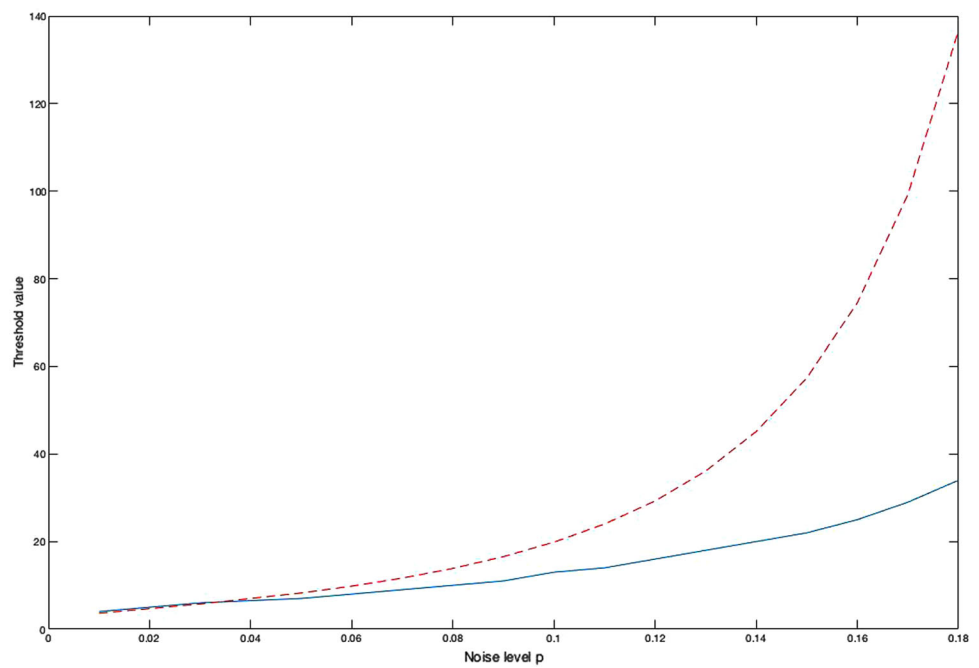


Fig. 14 Solutions computed with Algorithm 2 for a noise level equal to 0.05, $m = 8$ and different values of ε . **a** Ground truth. **b** $\varepsilon = 10^{-2}$ Threshold = 9, $\delta_1 = 2725$. **c** $\varepsilon = 10^{-4}$ Threshold = 19, $\delta_1 = 3614$. **d** $\varepsilon = 10^{-6}$ Threshold = 29, $\delta_1 = 4451$

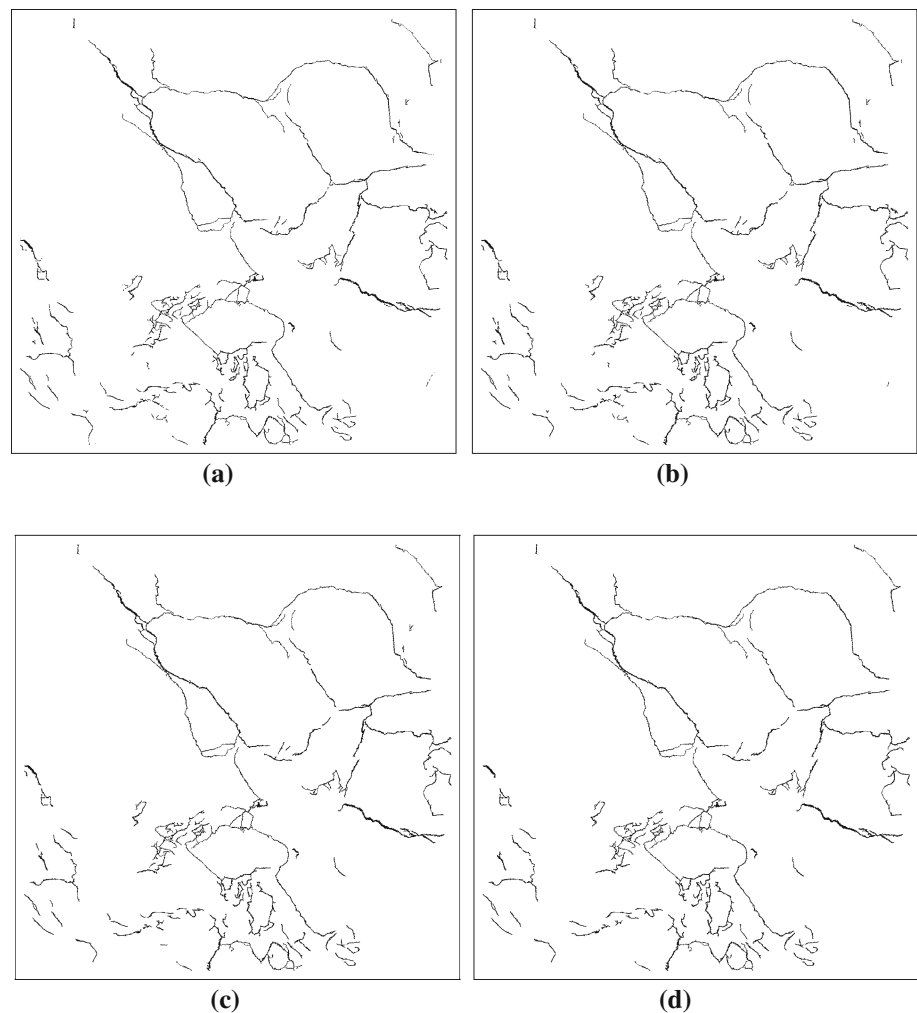


Fig. 15 Comparison of the solutions computed with Algorithms 1 and 2 for $m = 8$ and $\varepsilon = 10^{-6}$; difference between the two solutions for different noise levels. **a** Noise level: 0.01 ℓ^1 norm = 492. **b** Noise level: 0.05 ℓ^1 norm = 893. **c** Noise level: 0.1 ℓ^1 norm = 2215

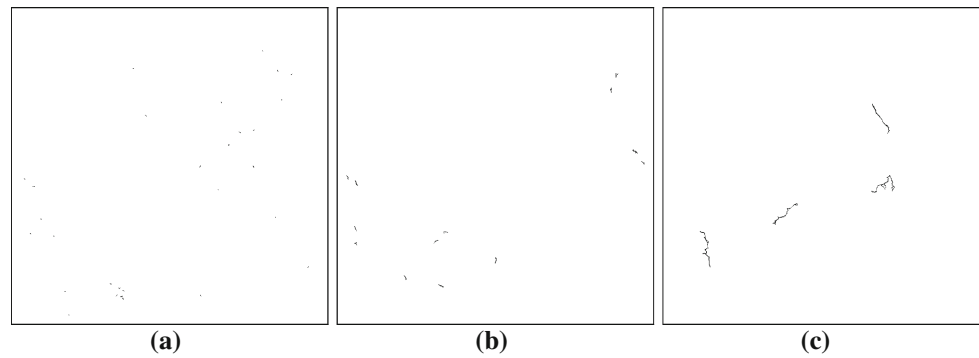
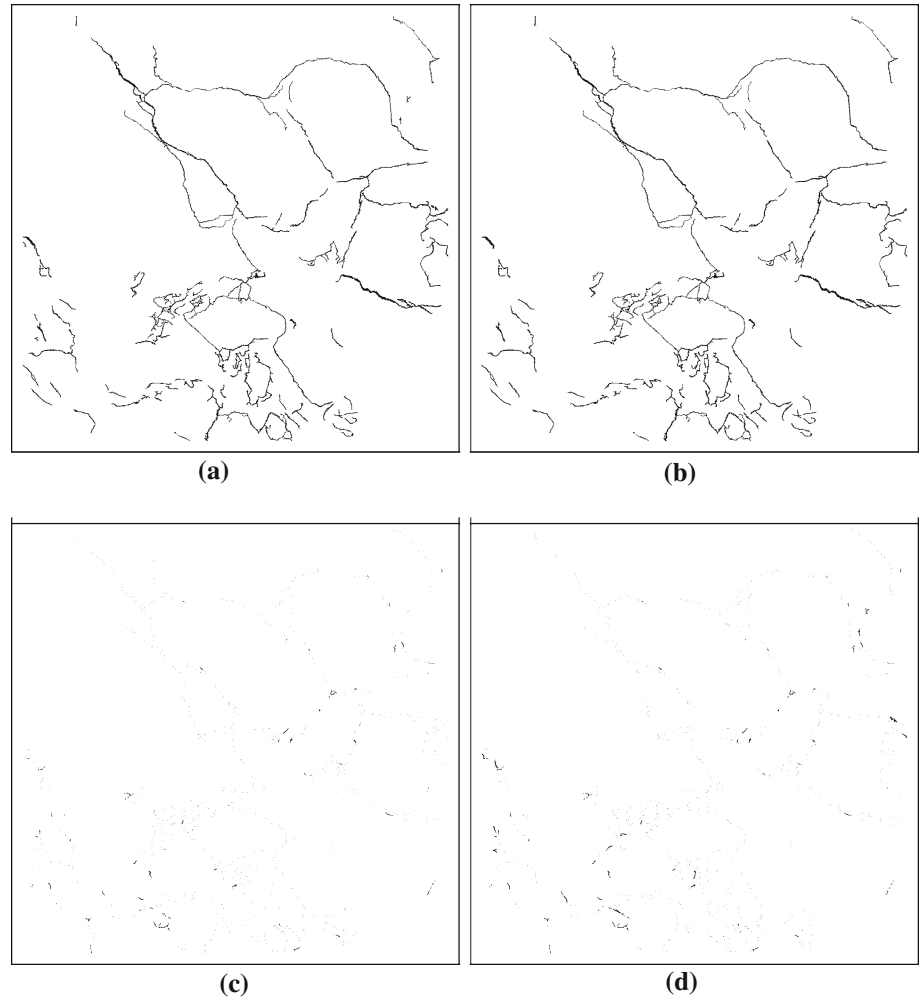


Fig. 16 Comparison of the solutions computed with Algorithms 1 and 2 for $m = 8$, $\varepsilon = 10^{-6}$ and a noise level 0.05. **a** Solution with Algorithm 1. **b** Solution with Algorithm 2. **c** Difference with the ground truth, $\delta_1 = 3688$. **d** Difference with the ground truth, $\delta_1 = 4451$



The use of the geodesic diameter is not appropriate if the noise is too large. Indeed, if mp is too close to 1, then the threshold $a_0 = 2 \frac{\log \varepsilon}{\log mp}$ becomes quite large and we lose most of the crack information. If the noise is small enough, then we get good results provided ε is not too large. However, the method seems to be robust with respect to small ε . Figures 14 and 16 present the results for a noise level equal to 0.05 and $m = 8$ (Figs. 15, 16)

3.2 Tests with a Real 2D Black and White Image

3.2.1 Preprocessing

We present some results on a 2D slice of a granite sample stack, for the sake of simplicity (see Fig. 1a). Though, it is not our purpose here, we briefly present the preprocessing method which is illustrated in Fig. 17:

Fig. 17 Image decomposition with $\lambda = 5$ and $\mu = 7$ —the contrast and lightning have been emphasized to provide a better view of the images. **a** Original (2D) image. **b** u component. **c** v component. **d** w component

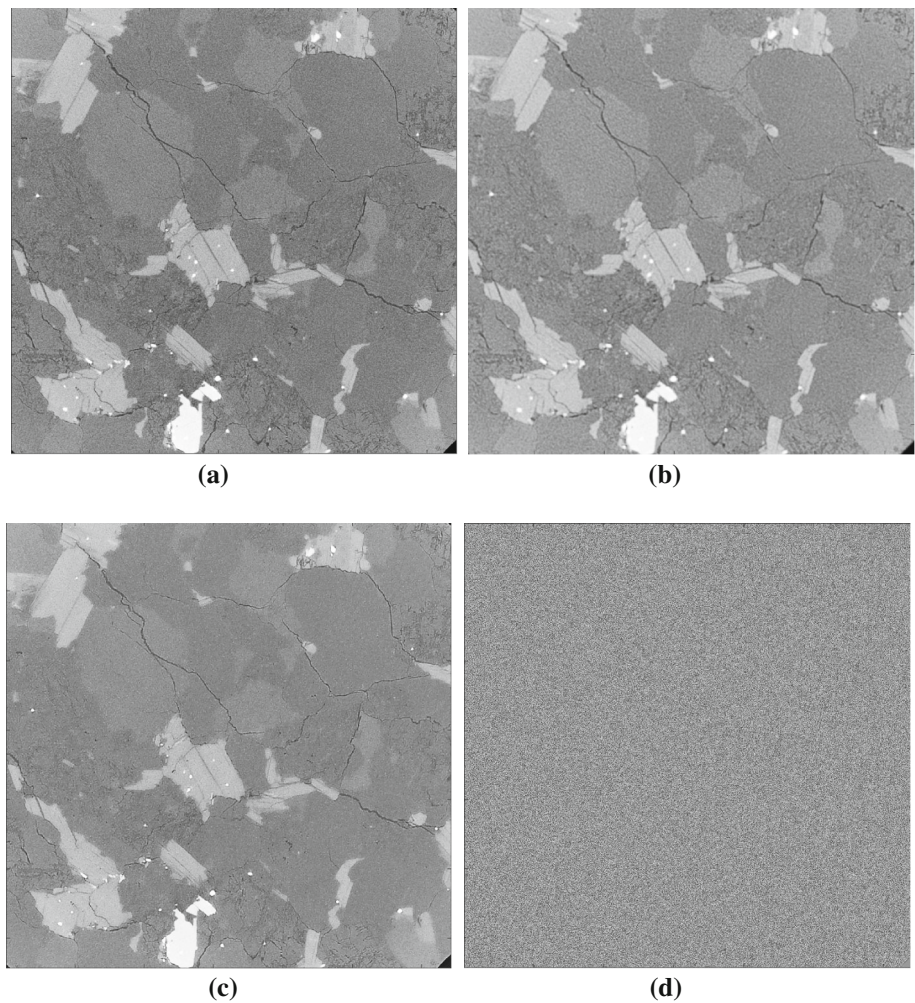
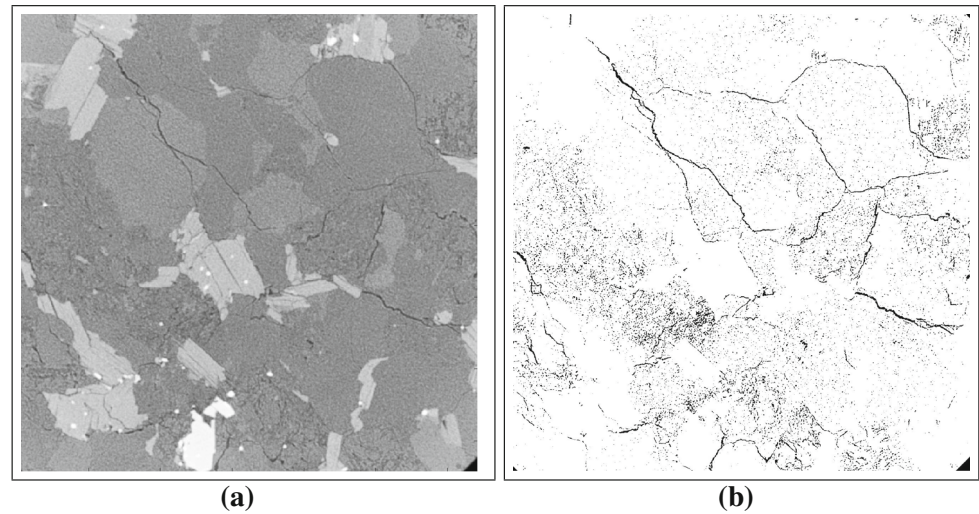


Fig. 18 Thresholded component used as the binary data. **a** u component. **b** Thresholded u component



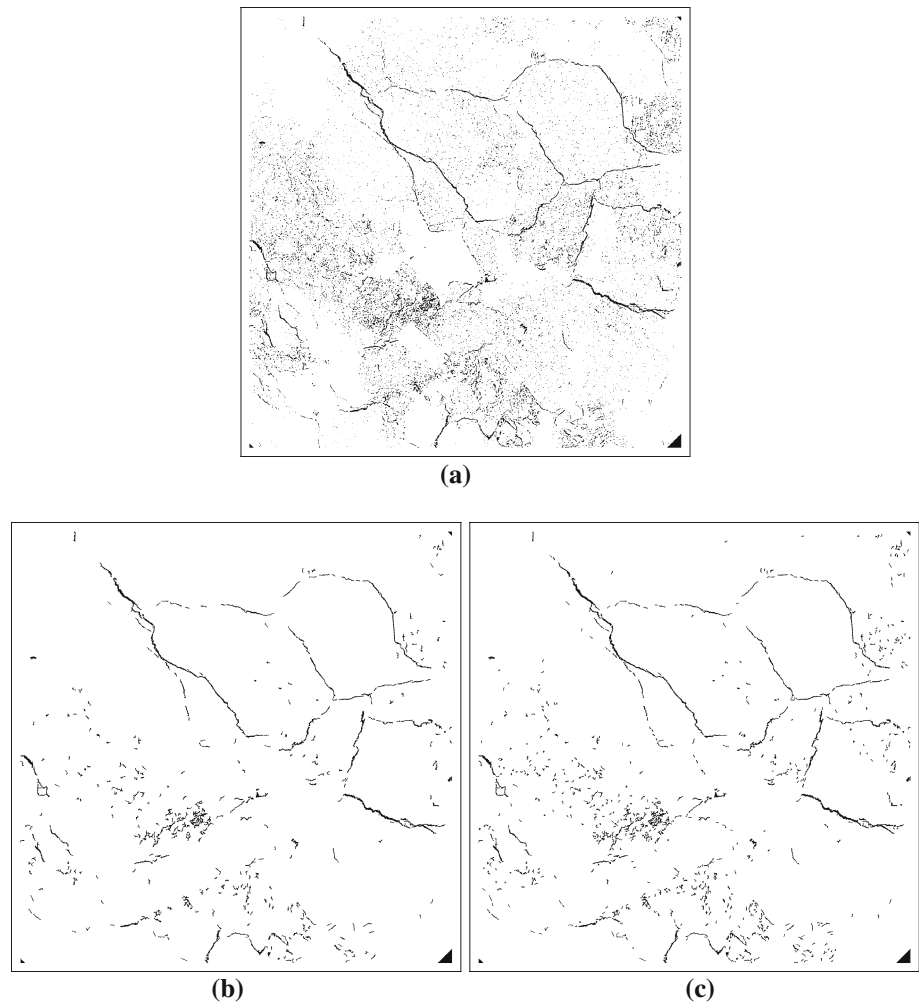
1. Original (2D or 3D) images are preprocessed with the method of [3,4]. More precisely, we use a variational model that allows to split the image in three components denoted v , u and w :

- the v component is a smooth (continuous) function whose distributional derivative is of bounded variation: It represents the image dynamic;
- the u component is a bounded variation function and ideally should be piecewise constant. It is called a

Table 4 Computed probability p_k and corresponding threshold a_k during the iterations— $m = 8$ and $\varepsilon = 10^{-6}$

Iteration	1	2	3	4	5
p_k	4.6928×10^{-2}	3.3656×10^{-2}	3.0146×10^{-2}	2.9171×10^{-2}	2.9171×10^{-2}
a_k	45	27	24	23	23

Fig. 19 Solutions computed by Algorithms 1 and 2 with $m = 8$ and $\varepsilon = 10^{-6}$. **a** Noisy data. **b** Solution with Algorithm 1. **c** Solution with Algorithm 2



cartoon component: The image contours are given by the jump set of u ;

- the w component is a L^2 function that represents the details and/or the noise.

The main inconvenient of variational methods is their dependence to some parameters. The result of the above decomposition depends on the choice of such parameters. We refer to [4] to get tuning rules for the two parameters λ and μ to be used in this context. From a practical point of view, we choose $\mu \simeq 1.2\lambda$. The tests we present here use $\lambda = 5$, $\mu = 7$.

The preprocessing method is robust enough with respect to the choice of parameters, and we can get rid of most noise (involved in the w component). So we focus on the cartoon component u that involves all the

contours. However, though the *Gaussian* part of the noise has been removed, impulse noise still remains. The challenge is to distinguish the cracks that are thin *structures* from the noise (which is *unstructured*). However, it will be still quite difficult to recover micro-cracks.

2. Next, the u component has to be binarized. This is a delicate step since the threshold parameter is not easy to find. We decided to choose $s = m(u) - \sigma(u)$ where m and σ are the mean value and the standard deviation, respectively, of u . This is an heuristic choice that we cannot fully justify by now.
3. Last, we use the thresholded (binary) image and use Algorithm 1.

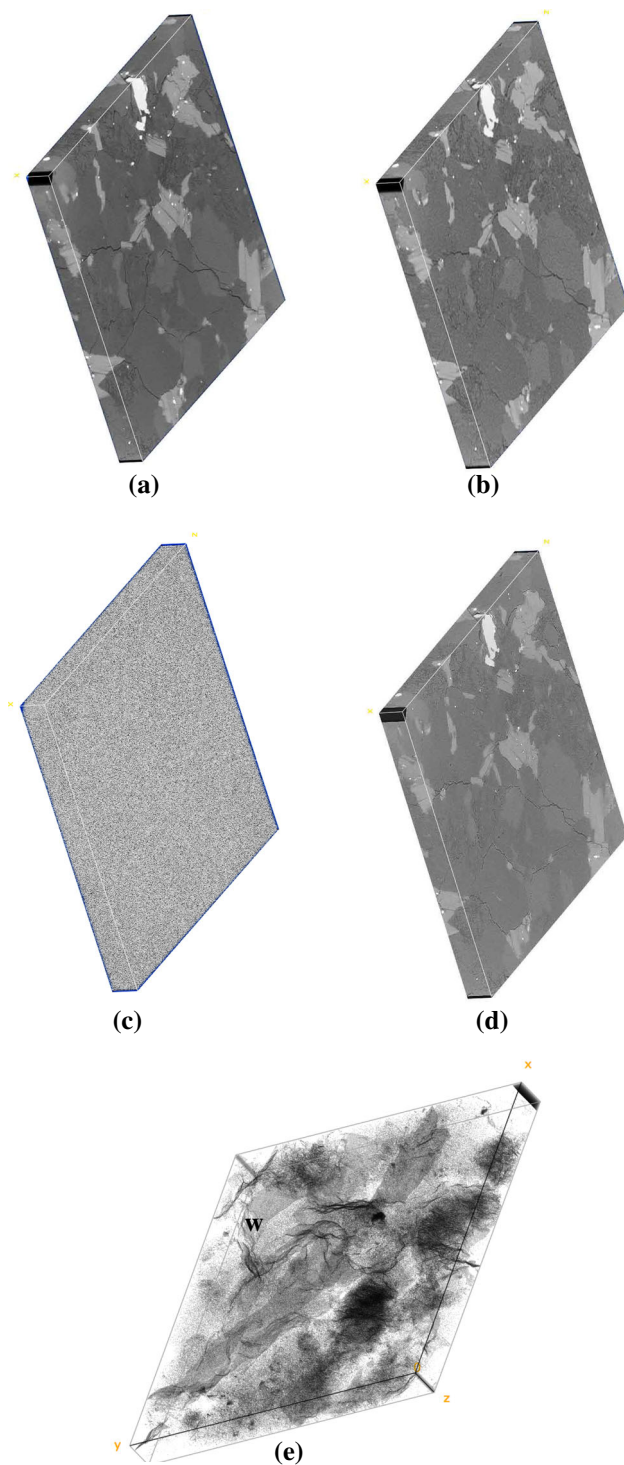


Fig. 20 Decomposition of the 3D stack and binarized cartoon component. **a** Original image. **b** Smooth component v . **c** Noise component w . **d** Cartoon component u . **e** Binarized cartoon component

3.2.2 2D Results

From now, we use the binary thresholded cartoon component as the data (see Fig. 18). We report in Table 4 the evolution

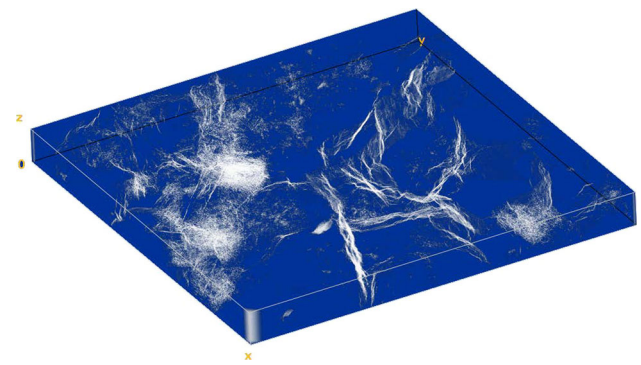


Fig. 21 Solution

of p estimate. We see that the iterative process becomes stationary very quickly (5 iterations).

Figure 19 shows the solution with Algorithms 1 and 2:

3.3 3D Images

Though, we have presented the method with 2D images (whose domains are $\llbracket -N, N \rrbracket^2$), we may extend the result to 3D stacks. Figure 20 shows the preprocessing result.

Recall that we consider a 3D stack whose size is $1287 \times 1287 \times 99$. If we consider the nearest neighbors of a pixel to define connectivity, the neighborhood of a pixel is now a cube of size 3 and the results apply with $m = 26$ and Algorithm 1 works similarly in the 3D case. The main difference was the choice of the built in MATLAB function to compute the connected components directly with the 3D stack (and $m = 26$).

The preprocessing was performed directly on the 3D stack (and not slice by slice) as explained in [4], and it was natural to use the 3D stack to perform binarization and denoising with Algorithm 1. Of course, the computation time is quite large since we have to estimate the size of a great number of connected components. Therefore, we have to develop further strategies to improve the 3D method as for example parallelization.

Figure 21 shows the solution computed with Algorithm 1, $m = 26$ and $\varepsilon = 10^{-6}$.

4 Conclusion

The automatic threshold we propose seems to be performing, and we are able to recover a lot of information from the binary images. Though, we have presented the method on a 2D material image, it can be applied to any binary images arising from appropriate preprocessings. One may think of as angiography images, for example.

However, we could say that the size of the connected components is rather a crude criterion for distinguishing cracks

for noise and some other geometric characteristic (as the diameter) would better discriminate our features. The key point is that we need an upper bound for the tail probability of the looked after characteristic (we obtained such a bound for $\mathbb{P}(\sharp Z \geq k)$ in the case of the total population size and for $\mathbb{P}(Z_n > 0)$ in the case of the diameter) which is not obvious in general (see, for instance, [1] for recent works in this direction).

Another point that we did not really focus on is the estimation of the parameter p as it was not the central point of this work. Our method (especially after iterations) seems rather satisfactory, but always overestimates this probability. Some more sophisticated methods have been developed in the literature and may be used to improve the results.

Acknowledgements We would like to thank all the anonymous referees for their fruitful comments that helped to improve this paper a lot.

References

- Addario-Berry, L., Devroye, L., Janson, S.: Sub-gaussian tail bounds for the width and height of conditioned Galton–Watson trees. *Ann. Probab.* **41**(2), 1072–1087, 03 (2013)
- Athreya, K.B., Ney, P.E.: *Branching Processes*. Dover Publications, Inc., Mineola, NY, 2004. Reprint of the 1972 original [Springer, New York; MR0373040]
- Bergounioux, M.: Mathematical analysis of a inf-convolution model for image processing. *J. Optim. Theory Appl.* **168**(1), 1–21 (2016)
- Bergounioux, M.: Second order decomposition model for image processing: numerical experimentation. In: Bergounioux, G., Peyré, M., Schnörr, C., Caillaud, J.-B., Haberkorn, T. (eds) *Variational Methods In Imaging and Geometric Control*, vol. 18 of *Radon Series on Computational and Applied Mathematics*, pp. 5–34. Degruyter (2017)
- Caselles, V., Monasse, P.: Grain filters. *J. Math. Imaging Vision* **17**(3), 249–270 (2002)
- Chambon, S.: Detection of points of interest for geodesic contours: application on road images for crack detection. In: *International Joint Conference on Computer Vision Theory and Applications, VISAPP*, p. sp (2011)
- Coudray, N., Karathanou, A., Chambon, S.: Multi-resolution approach for fine structure extraction: application and validation on road images. In: *International Joint Conference on Computer Vision Theory and Applications, VISAPP*, p. sp (2010)
- Coupier, D., Desolneux, A., Ycart, B.: Image denoising by statistical area thresholding. *J. Math. Imaging Vis.* **22**, 183–197 (2005)
- Desolneux, A., Moisan, L., Morel, J.-M.: Meaningful alignments. *Int. J. Comput. Vis.* **40**(1), 7–23 (2000)
- Desolneux, A., Moisan, L., Morel, J.-M.: *From Gestalt Theory to Image Analysis: A Probabilistic Approach*, vol. 34. Springer, London (2007)
- Dwass, M.: The total progeny in a branching process. *J. Appl. Probab.* **6**, 682–686 (1969)
- Linda, C.H., Jiji, G.W.: Crack detection in X-ray images using fuzzy index measure. *Appl. Soft Comput.* **11**(4), 3571–3579 (2011)
- Masnou, S., Morel, J.-M.: Image restoration involving connectedness. In: *Sixth International Workshop on Digital Image Processing and Computer Graphics*, pp. 84–95. International Society for Optics and Photonics (1998)
- Maurício, A., Figueiredo, C., Pereira, M.F.C., Alves, C., Bergounioux, M., Rozenbaum, O.: Assessment of stone heritage decay by X-ray computed microtomography: I-a case study of portuguese braga granite. *Microsc. Microanal.* **21**(S6), 160–161 (2015)
- Moysan, J., Benoist, P., Corneloup, G., Magnin, I.: Crack-like defect detection and sizing from image segmentation through co-occurrence matrix analysis. *Ultrasonics* **30**(6), 359–363 (1992)
- Nguyen, T.S., Begot, S., Duculty, F., Avila, M.: Free-form anisotropy: a new method for crack detection on pavement surface images. In: *2011 18th IEEE International Conference on Image Processing*, pp. 1069–1072. IEEE (2011)
- Oliveira, H., Correia, P.L.: Automatic crack detection on road imagery using anisotropic diffusion and region linkage. In: *Signal Processing Conference, 2010 18th European*, pp. 274–278. IEEE (2010)
- Penrose, M.D.: Continuum percolation and Euclidian minimal spanning tree in high dimensions. *Ann. Appl. Probab.* **6**(2), 528–544 (1996)
- Popov, A.T.: Approximate connectivity and mathematical morphology. In: Goutsias, J., Vincent, L., Bloomberg, D.S. (eds) *Mathematical Morphology and its Applications to Image and Signal Processing*, pp. 149–158. Springer, Boston (2000)
- Pritchard, A.J., Sangwine, S.J., Horne, R.E.N.: Area thresholding and silhouette extraction of simple coloured objects using hue. In: *Morphological and Nonlinear Image Processing Techniques, IEE Colloquium on*, pp. 3–1. IET (1993)
- Rozenbaum, O., Bergounioux, M., Maurício, A., Figueiredo, C., Alves, C., Barbanson, L.: Versatile three-dimensional denoising and segmentation method of X-ray tomographic images: applications to geomaterials characterizations. Preprint (2016)
- Shi, Z.: *Branching random walks*. Springer 2015. École d'Été de Probabilités de Saint-Flour XLII (2012)
- Subirats, P., Dumoulin, J., Legeay, V., Barba, D.: Automation of pavement surface crack detection using the continuous wavelet transform. In: *2006 International Conference on Image Processing*, pp. 3037–3040. IEEE (2006)
- Tenenbaum, J.M., Witkin, A.P.: On the role of structure in vision. In: *Human and Machine Vision*, pp. 481–543 (1983)
- Vincent, L.: Grayscale area openings and closings, their efficient implementation and applications, pp. 22–27. In: *First Workshop on Mathematical Morphology and its Applications to, Signal Processing* (1993)
- Yamaguchi, T., Nakamura, S., Saegusa, R., Hashimoto, S.: Image-based crack detection for real concrete surfaces. *IEEJ Trans. Electr. Electron. Eng.* **3**(1), 128–135 (2008)
- Zhu, S.-C.: Embedding gestalt laws in markov random fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(11), 1170–1187 (1999)



Romain Abraham is a professor of Mathematics at University of Orléans, France. He was born in Saint-Dizier (France) in 1968. After studying at Ecole Normale Supérieure, he received his Ph.D. degree in 1993 and became Assistant Professor at University Paris Descartes in 1994. He moved to a Full Professor position at University of Orléans in 2004. His main research concerns probability theory, in particular stochastic processes and random trees, but he is also interested in image processing.



Pierre Debs is an Assistant Professor of Mathematics at University of Orléans, France. He was born in Metz (France) in 1981. He studied in Nancy and received his Ph.D. degree in 2007. He became Assistant Professor in 2008 at the MAPMO (Mathématiques-Analyse, Probabilités, Modélisation- Orléans). He is a specialist of probabilities, more precisely random walks, random walks in random environment, penalization and random trees.



Maïtine Bergounioux is a Professor of Mathematics at University of Orléans, France. She was born in Bayonne (France) in 1956, received her Ph.D. degree in 1985 and was appointed professor in 1998. She was a former student in Ecole Normale Supérieure. Her research interests in image processing relate to variational methods, texture analysis and segmentation processes. She is also specialist of inverse problems and parameter identification, especially in the context of medical imaging (PET reconstruction, thermo-acoustic tomography).