

Journal of Electronic Imaging

JElectronicImaging.org

Robust crack detection for unmanned aerial vehicles inspection in an *a-contrario* decision framework

Emanuel Aldea
Sylvie Le Hégarat-Masclé

Robust crack detection for unmanned aerial vehicles inspection in an *a-contrario* decision framework

Emanuel Aldea* and Sylvie Le Hégarat-Mascle

University Paris-Sud, Institute of Fundamental Electronics, Department of Autonomous Systems, Orsay cedex 91405, France

Abstract. We are interested in the performance of currently available algorithms for the detection of cracks in the specific context of aerial inspection, which is characterized by image quality degradation. We focus on two widely used families of algorithms based on minimal cost path analysis and on image percolation, and we highlight their limitations in this context. Furthermore, we propose an improved strategy based on *a-contrario* modeling which is able to withstand significant motion blur due to the absence of various thresholds which are usually required in order to cope with varying crack appearances and with varying levels of degradation. The experiments are performed on real image datasets to which we applied complex blur, and the results show that the proposed strategy is effective, while other methods which perform well on good quality data experience significant difficulties with degraded images. © 2015 SPIE and IS&T [DOI: [10.1117/1.JEI.24.6.061119](https://doi.org/10.1117/1.JEI.24.6.061119)]

Keywords: nondestructive examination; crack detection; image processing; visual inspection; a-contrario modeling.

Paper 15520SSP received Jun. 28, 2015; accepted for publication Nov. 30, 2015; published online Dec. 30, 2015.

1 Introduction

Either for road quality assessment, for the examination of buildings or for the inspection of critical infrastructure, an automated framework aimed at the detection of surface defects (cracks, pitting corrosion, rutting, and so on) may have a significant positive impact in terms of cutting down maintenance costs and enhancing security. One major reason supporting automation is that human supervision is prohibitively costly for large scale, frequent tasks required by infrastructure surveillance. Detecting and monitoring the degradation level in order to decide upon the necessity of corrective measures are also subjective processes requiring further deliberation, and a solution operating with minimal human intervention is vital for scaling up the maintenance cycle.

Among surface defects, cracks exhibit a high level of pervasiveness on different mediums which are affected by various phenomena such as hygrothermal aging and fatigue. However, their relatively similar appearance makes image processing an ideal tool for detection, all the more so as cameras are affordable sensors which are also ideal for non-destructive examination (NDE).

For these reasons, there is a large body of work aimed at automating the problem of crack detection.¹⁻⁵ More recently, robotic systems and unmanned aerial vehicles (UAVs) have further increased the quantity of data to be processed for two main reasons. First of all, the operating cost for road network inspection is potentially lower than that of using monitoring vehicles, especially if the navigation process is greatly automated. Also, UAVs have the ability to approach scenes which would otherwise be difficult to access or even unreachable under safe circumstances.

The main challenges in this field are related to navigation since different levels of positioning precision are required for

the detection of different types of defects. Currently, GPS and inertial sensors allow for navigating accurately enough in order to highlight surface deformations of the order of 1 cm, or in order to follow a desired trajectory with an error bound of 0.5 to 1 m.^{6,7} In order to detect finer cracks in asphalt or in concrete, a better localization of the embedded imaging sensor is necessary, particularly with respect to the analyzed surface. A closer working distance using an aerial device presents the main disadvantage of degrading the image quality as a result of various phenomena specific to image recording in the presence of imprecise relative localization and/or significant camera dynamics, such as motion blur, defocus, or fast illumination variations. These degradations represent a significant drawback since they have a direct impact on the salience of cracks, which are often challenging to detect.

Our work addresses the current gap between robust pattern recognition algorithms which are suited for complex detection tasks, and their evaluation in conditions which are more representative for NDE using automated systems exhibiting fast, imprecise dynamics. Our main assumption is that while there are multiple hardware- and software-based strategies available for coping with high frequency vibrations, out of focus positioning, and motion blur, a significant level of degradation will continue to be present. It is, therefore, important to identify crack detection strategies which are resilient to degradations specific to UAV-borne image recording.

In order to further emphasize the interest of our investigation, we note in the literature a recent number of works which focus on validating the functionality of a robotic examination system.⁸⁻¹² Very few works (e.g., Ref. 13) consider in their experimental validation the correlation between acquisition factors and the deterioration in detection performance; the experimental evaluation is usually performed using basic processing tasks on images acquired in favorable

*Address all correspondence to: Emanuel Aldea, E-mail: emanuel.aldea@u-psud.fr

conditions. In contrast, a distinct and significant body of work deals with crack detection as a pattern recognition problem (see, e.g., the following section, or work¹⁴ for a review), and assumes ideal or at least adequate image quality.

Recently, some approaches denoted as *a-contrario* strategies have been proposed in order to detect structures in an image by modeling the absence of structures rather than the structures themselves (hence the name *a-contrario*). The basic idea is to detect what contradicts a naive model representing the signal in the absence of structures, i.e., representing noise or background. The advantage is that this naive model may be very approximate provided that it is false in the presence of a structure. Then due to the approximate nature of the naive model, a subsequent advantage of these approaches is their robustness to the model parameter(s). Finally, reformulating the problem of detection in terms of a problem of maximizing the contradiction relative to a naive model allows us to avoid relying on any threshold parameter. In this paper, we propose such an approach, based on *a-contrario* decisions, in order to detect and to reconstruct the crack, taking advantage of their robustness. Since we avoid tuning various method parameters, in order to grasp defect properties which may be less obvious according to the extent of image quality loss, we are able to formulate the detection as a structural search at image level, as opposed to methods which cluster local detections.

In Sec. 2, we first introduce some classical approaches originating from the pattern recognition community which address the problem of crack detection, and we highlight their limitations. Then we present in Sec. 3 the framework for our strategy and enumerate the steps of the detection algorithm we propose. Section 4 presents the results we obtain in the presence of image degradation compared to reference methods, and finally, the conclusion is presented in Sec. 5.

2 Related Work

Crack detection remains a challenging task mainly because the structure of interest is often scarcely represented in the image space, while at the same time exhibiting radiometric properties which are shared with other elements, such as shadows, holes, and mold. Although some solutions have been proposed^{15–17} for identifying “unusual” structures in a broad sense, the community has settled on a discriminative definition which considers the crack as “a thin, linear object, mostly composed of dark pixels.” Proposed algorithms usually perform crack detection in a two-step process.

Local analysis: In a first stage, a local analysis is performed exhaustively in the image space based on the properties of local minimum paths, or by employing alternative techniques running in a local window, such as mathematical morphology, spectral analysis, covariance estimation, or saliency evaluation.^{18–21} The objective of this first step is to associate to each image location a numerical indicator of being part of a crack. This first step does not entirely solve the detection problem by itself since areas of the analyzed surface may exhibit crack-like properties at a fine scale while not belonging to an actual crack structure.

Reconnection (also denoted as seed-growing): A second step is required in order to actually reconstruct the cracks according to the definition above by imposing a geometric coherence upon the crack membership map provided by

the first step. As its name recalls, this step is necessary not only for pruning the false positives generated by the local analysis but also for identifying crack parts which are less apparent but which extend other obvious parts by presenting at the same time an acceptable level of “atypicalness” compared to their length.

Reconnection is a subjective process, and strategies vary significantly. Some works ignore this step altogether and rely on fine-tuning the local detection^{4,22} in order to eliminate false positives. Reconnection may be performed on a binary map² provided by the local analysis, but in this case the process relies only on geometric considerations. Another solution is represented by an iterative process^{1,23} which updates the probabilities’ values of pixels being part of a crack. Finally, one may rely on computing minimal paths, this time in the whole image space, between potential crack parts; then *ad hoc* rules or a strategy minimizing a connection cost^{24,25} may be used in order to accept or reject the potential paths.

Both processing steps require a significant amount of tuning via thresholding in order to account for the specificity of cracks (size, singularity, and regularity), as well as for the compromise which is intended in terms of noise versus coarseness of detection. Unfortunately, in the specific context of UAV data acquisition, the image degradation has an added negative impact upon the processing steps, and implicitly upon the final result of the detection. In the following paragraphs, we will try to highlight more clearly the limitations of related works and also the mechanisms that degrade the performance in the UAV context.

The performance of the local analysis is actually of prime importance; therefore, this step constitutes the distinctive trait among different approaches. In the following parts of this section, we will detail two widely used strategies employed for local analysis in order to highlight the challenges raised by image degradation.

2.1 Minimum Path-Based Strategies

Free-form anisotropy (FFA) is a defect detection approach²² which is based on the assumption that if a pixel belongs to a defect, then a significant minimal path traverses that pixel, the cost being represented by pixel intensities. Therefore, for a given pixel location l and for a fixed local window, we consider a set of four constrained minimal paths traversing the location, i.e., up–down (UD), left–right (LR), upper left–lower right (ULLR), and upper right–lower left (URLL). In order to compute these minimal paths efficiently, directional subpaths originating from the central pixel are imposed each time by two opposing pairs of three-connectivity oriented neighborhoods. If we assume that a fissure corresponds to a minimal path which exhibits statistical properties which are different from those of the other three “regular” minimal paths, we dispose of a crack detection strategy which does not require any training data. In the first two images of Figs. 1(a) and 1(b), we illustrate the four minimal paths (red for UD, yellow for LR, white for ULLR, and blue for URLL) in the case of a crack pixel, and that of a normal area pixel, respectively. The blue square indicates the size of the analysis window. Note that for the crack pixel, the yellow path is able to accurately follow the crack, while two other paths follow it only partially due to the imposed

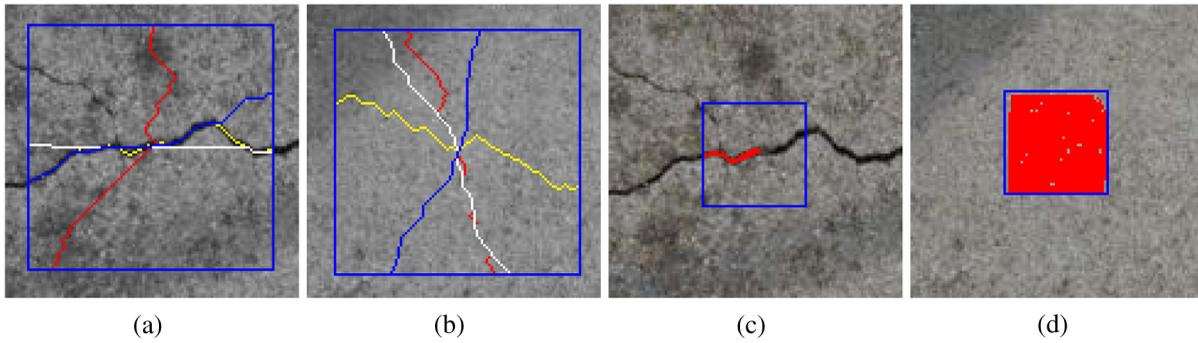


Fig. 1 Behavior of (a, b) free-form anisotropy (FFA) and (c, d) percolation local analysis methods for crack pixels and normal areas, respectively.

connectivity, and a fourth path is constrained to explore only normal areas.

Let us denote by $s_1 = (\mu_1, \sigma_1^2)$ and by $s_2 = (\mu_2, \sigma_2^2)$ the mean and variance of pixel radiometric values for the minimal cost path (among the four of them), and, respectively, for the three others combined. Avila et al.²² rely on a degree of coherence h between two sources adopted from the possibility theory. A low degree of coherence, close to 0, indicates that the first source is different from the second, thus in our situation, a low value of h implies that the pixel may very well be located on a genuine fissure.

A final step is necessary in order to get a binary assignment at pixel level related to the presence of a defect. For the input image locations $l_i \in I$, $h(l_i) \in [0, 1]$, thus a thresholding step is required. Avila et al.²² apply a hysteresis threshold which is performed using a multiclass variant of the Otsu algorithm.

Adjustments for our problem: In our experiments, we replaced the degree of coherence, since a distance between two distributions may be approached straightforwardly by computing the Bhattacharyya distance $D_B(s_1, s_2) \in [0, \infty)$, then the Bhattacharyya coefficient $BC(s_1, s_2) = e^{-D_B(s_1, s_2)}$ that we will employ for the same purposes is h .

Impact of image degradation: FFA is based on the assumption that the local crack path exhibits different statistical properties. This assumption is used for identifying cracks via thresholding the value of $h(l_i)$; the relevance of this step is directly affected by blurring phenomena which clump together the values of s_1 and s_2 .

2.2 Image Percolation

For this approach, we mainly consider the following reference.⁴ The percolation model is based on the natural phenomenon of liquid permeation. In this analogy, pixels within the analysis window are percolated, starting with the central pixel. Their progressive inclusion in the percolation set around the initial pixel is based on a comparison with a dynamic threshold; the pixel values must be lower than the threshold in order to be percolated, but the threshold value increases progressively during the iterative percolation process. In this manner, if the pixel does lie on a crack, the percolation process will build up a path along the crack toward the border of the local analysis window [Fig. 1(c)] while in the normal case the percolation will tend to evolve isotropically [Fig. 1(d)]. A circularity feature computed on the resulting percolation set indicates whether a potential

crack is present (linear structure) or not (rather circular structure).

Impact of image degradation: The percolation method and the computationally efficient optimization are based on the assumption that a noncrack pixel exhibits an isotropic diffusion at an early stage of the percolation process. However, image degradation brings the crack pixel values closer to the ones of neighboring pixels and promotes an increased level of isotropic percolation. In turn, the percolation analysis will not be able to clearly differentiate crack pixels from homogeneous ones.

2.3 Limitations and Immediate Perspectives

As we previously stated, these approaches provide a map of indicators, and the resulting map has to be thresholded and then postprocessed. Here, we are more interested in estimating the robustness of this first step of local analysis with respect to image degradation, or, stated differently, we note that if this first step fails significantly then the reconnection is futile. Recently, some works have been proposed^{25,26} which implicitly place more weight on the reconnection step; the strategy is to select a significant set of potential crack pixels as seeds at a local level, connect them in a minimal path network at the image level, then identify the cracks as a subset of the resulting network. Compared to the previous strategies, this one exhibits some major advantages. First, it relies jointly on global geometric and photometric constraints in order to identify the cracks; second, the computation of the minimal paths may be performed efficiently. However, the nontrivial aspect which remains to be dealt with is the discontinuity of cracks in certain areas. In Ref. 25, the cost function associated to a path from source s to destination d

$$C(s, d) = \frac{1}{\text{card}(s \rightarrow d)} \sum_{m=s}^d I(m), \quad (1)$$

is nonmonotonic, thus it is up to the user to tune the algorithm in order to favor short, distinctive cracks, or long ones, which lack potentially specific characteristics locally. As a result, this behavior is controlled by four thresholds, and although they are inferred automatically from image and cost function statistics, the selection process remains subjective and sensitive to changes in image characteristics. In another related work,²⁶ a local distance histogram²⁷ is used for seed detection, then crack elements are formed, linked, and finally

pruned. Again, the linking and pruning parts make empirical assumptions about what cracks to connect or to remove (validation area for potential links and length for pruning branches). Although these assumptions are adapted for standard crack appearance, they are based on thresholds and they miss defects which are nonetheless salient compared to their background.

We emphasize the fact that our detection algorithm is positioned between algorithms which perform the main processing step in a local window^{4,22} iteratively, and global detection algorithms, which rely on a costly optimization step in order to enforce constraints on the result at image level.^{26,28} Coupled with the NFA process, the dynamic programming (DP) strategy allows us to avoid producing local decision errors (i.e., selecting small salient objects or missing fine connections between large cracks), while at the same time maintaining a low computational complexity.

2.4 Main Contributions

The main contribution of our work is that we propose a solid statistical framework for coping with the analysis of variable length minimal paths which avoids proposing heuristics for costs such as the one in Eq. (1). Instead, the *a-contrario* strategy identifies robustly interesting structures, and thus allows us to be robust in the presence of an arbitrary degradation of image quality. We emphasize the following points that allow our proposed algorithm to exhibit improved robustness to data acquired with degraded quality:

1. an *a-contrario*-based algorithm for seed selection at local level,
2. an *a-contrario*-based reconnection step which identifies minimal paths by optimizing their “atypicalness” jointly in terms of length and appearance with respect to a statistical background model.

The current work builds up on an initial investigation²⁹ where we were interested in the basic applicability of an *a-contrario* reconnection, compared to a local (FFA) and also a complex (marked point process), global approach. In the present version, we perform quantitative comparisons, refine our main algorithm, and add preprocessing modules which allow us to test using several datasets.

A flowchart for visualizing the data flow of our algorithm is provided in Fig. 2, and the following section provides the details regarding the different modules. The three main steps identified in Fig. 2 could be substituted by other processing

with the same input and output. For instance, the preprocessing could be changed depending on the features of the materials which are being investigated. The seed selection can either be based only on pixel gray level values (as proposed) or involve neighborhood tests (e.g., local minima). However, we found it more consistent to use the same probabilistic framework all along the study.

3 Proposed Approach

3.1 Data Features

Cracks not only have a different intensity level compared to a normal surface, they are also typically found as thin lines in an image. Space scales and Hessian matrix have been first proposed^{30,31} to detect ridges, thus second-derivative filters are widely used^{32,33} to extract such linear structures. In order to compute derivatives along different orientations, several approaches have been proposed such as morphological path operators³⁴ or the steerable filters. These latter, first proposed by Freeman and Adelson,³⁵ are a class of filters in which a filter of arbitrary orientation is obtained as a linear combination of a set of basis filters. Due to the linearity of the convolution, the result of the filter at angle θ can be computed as a linear combination of basis filters. In their work, Jeong et al.³⁶ used a steerable filter to evaluate the relevance of the orientation of a segment representing a subpart of the face wrinkles. The basis filter set they used is provided by the second derivatives of the two-dimensional (2-D) centered Gaussian function. Denoting f_g , the one-dimensional (1-D) centered Gaussian function $f_g(t) = 1/\sqrt{2\pi}\sigma \exp(-t^2/2\sigma^2)$, for any pixel $s = (x, y)$ where x and y are the column and row axes, respectively, assuming independence between x and y , $f_G(s) = f_g(x) \times f_g(y)$ and

$$\mathbf{G}_2(s) \propto \begin{bmatrix} \frac{\partial^2 f_G(s)}{\partial x^2} & \frac{\partial^2 f_G(s)}{\partial x \partial y} & \frac{\partial^2 f_G(s)}{\partial y^2} \end{bmatrix}^\top, \quad (2)$$

$$\propto \begin{bmatrix} (\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2})f_G(s) & \frac{xy}{\sigma^4}f_G(s) & (\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2})f_G(s) \end{bmatrix}^\top. \quad (3)$$

This formulation is more general than that in Ref. 35 Appendix H, since σ appears as a scale parameter. In addition, normalization coefficients are not considered for the reasons which will be stated below. Thus, the interpolating function of orientation is, according to Ref. 35 Appendix H, $k(\theta) = [\cos^2\theta \quad -\sin 2\theta \quad \sin^2\theta]^\top$. In practice, once the three images, noted $\mathbf{G}_{2,1} * I$, $\mathbf{G}_{2,2} * I$, and $\mathbf{G}_{2,3} * I$, corresponding

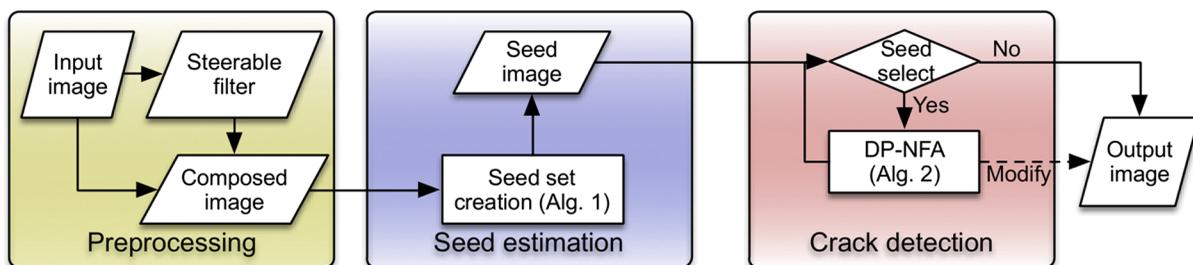


Fig. 2 A comprehensive flowchart of the different processing steps. The preprocessing part is introduced in Sec. 3.1, and Algorithms 1 and 2 are detailed in Sec. 3.5. Algorithm 3, also presented in Sec. 3.5, corresponds to the entire flowchart.

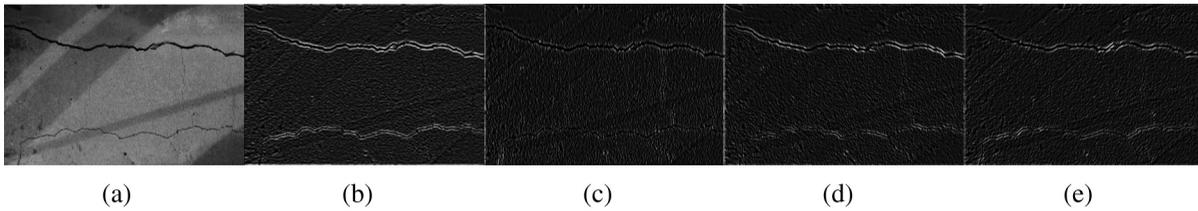


Fig. 3 Responses $\mathbf{G}_{2,\theta} * I$ of the steerable filter to (a) for four directions of $\theta: \{0, \frac{\pi}{2}, \frac{\pi}{4}, \frac{3\pi}{4}\}$ in (b) to (e), respectively.

to the filtering of an original image I by each of the three components of \mathbf{G}_2 are computed, the result of the second-derivative filter associated with any orientation θ can be obtained equivalently by filtering the image by filter $\mathbf{G}_{2,\theta}(s) = \mathbf{k}(\theta)^T \mathbf{G}_2(s)$ or as the linear combination of previous images weighted by $\mathbf{k}(\theta)$ components: $\mathbf{G}_{2,\theta} * I \propto (\mathbf{G}_{2,1} * I) \cos^2 \theta - (\mathbf{G}_{2,2} * I) \sin 2\theta + (\mathbf{G}_{2,3} * I) \sin^2 \theta$.

Figure 3 shows the result of the application of the steerable filter to a crack image [Fig. 3(a)] with $\theta \in \{0, \pi/2, \pi/4, 3\pi/4\}$. We see that the crack edges appear or not depending their direction with respect to the orientation θ .

Finally, since searching for thin, linear structures in the images boils down to searching structures with a high gradient norm in only one direction, we compute the maximum (over the investigated orientations) of the ratio between the filter response in one direction θ and filter response in the perpendicular direction and we define

$$r_I(s) = -\max_{\theta} \frac{|\mathbf{G}_{2,\theta} * I(s)|}{|\mathbf{G}_{2,\theta+\frac{\pi}{2}} * I(s)|}. \quad (4)$$

In Eq. (4), we take the opposite (or any involution reversing the ordering) of ratio maximum since further processing involves a minimization. In addition, in our case, θ values are subsampled so that $\theta \in \{0, \pi/4, \pi/2, 3\pi/4\}$.

Image r_I may be rather noisy. In addition, in the case of crack detection, the knowledge that crack pixels are dark can be used as a strong prior. Then for crack estimation, we consider the image Y_I that is a linear combination between image I and r_I image after dynamic normalization: denoting $\max_r = \max_s \{r_I(s)\}$

$$Y_I(s) = I(s) - \alpha \times \left[\frac{255}{\max_r} r_I(s) \right]. \quad (5)$$

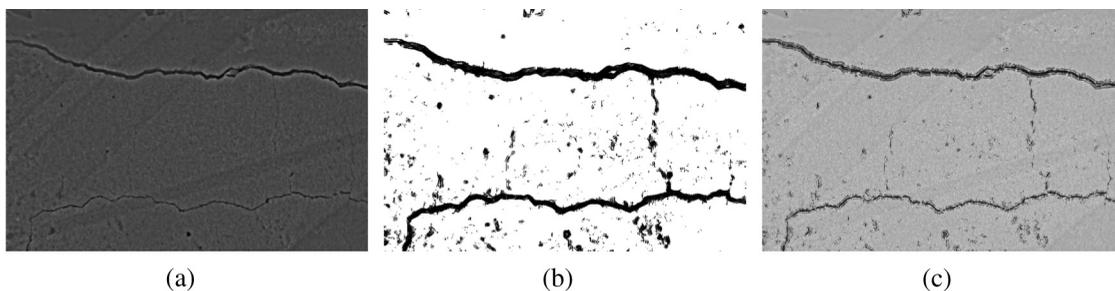


Fig. 4 Derivation of the image Y_I in (c), used for crack detection as a linear combination of background subtraction in (a) with $\tilde{I} = I - f_{G(\sigma)} * \phi(I)$ [I being shown in Fig. 3(a)], and ratio image r_I [Eq. (5)] shown in (b), here with $\alpha = 0.5$.

Finally, in order to be robust to shadow, I is not the original image of the output of a preprocessing step to remove large scale nonstationary effects. In this work, this preprocessing step is simply a background subtraction where the background is the result of the Gaussian filtering of an original image with parameter $\sigma = 5$ performed after a morphological opening³⁷ of the image (that allows for the removal of the cracks as thin dark structures and, therefore, for a better estimation of the background). Let \tilde{I} denote the I image after this background subtraction step. Figure 4 shows the background subtraction result $\tilde{I} = I - f_{G(\sigma)} * \phi(I)$, where $\phi(\cdot)$ denotes the morphological opening operator, the r_I image, and the obtained Y_I image. We see that the cracks appear as connected sets of dark value pixels.

3.2 Probabilistic Measure

In this section, we denote the data image in a generic way by X . In our application, X will refer to Y_I as defined by Eq. (5), but the proposed algorithms may be applied to other entry data. We only assume that X has only positive values: $\forall s, X(s) \geq 0$.

In order to evaluate the likelihood that a subset of pixels (either connected or not as we will see further) corresponds to a crack, we need a measurement. In this study, we focus on the number of false alarms (NFA)³⁸ that quantify the number of cases where a detection may occur only “by chance.” In order to model the noise rather than modeling the researched structures (as stated in Sec. 1), we assume a “naive” model representing the noise (here, the term “naive” emphasizes the fact that this noise model is simple). Based on this naive model, we can compute the probability of any observation including observation of structured data. For instance, Desolneux et al.³⁸ evaluate the probability that random uniform distribution of 2-D points leads to an alignment “by chance.” However, since any observation with a non-null

probability may occur when increasing the number of random draws, the significance of an observation should be evaluated taking into account the number of draws or possibilities of occurrence. Thus works³⁹⁻⁴¹ defined the NFA as the expectation of the number of occurrences of a structured observation under the assumed naive model and proposed to compute it as the product between the probability value (under the naive model) and a normalization term equivalent to the number of tests.

In this study, we intend to detect crack structures. The naive model is defined to be contradicted in the presence of a crack. It represents an image without any crack but that is not necessarily a realistic representation of the background (which depends on background material, e.g., asphalt for roads and concrete for walls). Indeed, the naive model should be simple and involve as few parameters as possible, otherwise its interest relative to the definition of the structure model will be low. Following^{42,43} works, we use a Gaussian model. In order to handle only one parameter and since the naive model does not have to be realistic, we simply take a mean equal to 0 and variance equal to image X second moment:

Definition 1 (naive model H_0). The image X is a random field of $|X|$ independently centered Gaussian variables $\mathcal{N}(0, \sigma^2)$ with $|X|$ the cardinality of X and $\sigma^2 = \mathbb{E}[X^2(s)]$.

Relative to the H_0 model, the presence of a crack will be characterized by the presence, in the image, of a subset D of pixels presenting surprisingly 0-close values (dark pixels). If this subset presents a low cardinality, according to H_0 , the “surprise” is also low, but when the cardinality of D increases, the probability of its observation decreases under H_0 hypothesis (even faster than σ^2 is large). Then an actual crack characterized by the presence of a significant number of dark pixels will be detected as highly improbable. To compute this probability, we consider the sum of the square values (SSV) of X on a subdomain D (included in the set of pixels): $v_D = \sum_{s \in D} [X(s)]^2$. The probability of observing v_D by chance is denoted $\mathbb{P}_{H_0}(v_D)$ and the NFA associated with the observation of a subdomain D having v_D SSV is defined by

$$\text{NFA}(D) = \eta(D) \mathbb{P}_{H_0}(v_D, \sigma), \quad (6)$$

where $\eta(D)$ is the “number of tests” that is chosen equal to $|\mathcal{D}| \binom{|X|}{|D|}$ as in Ref. 43 [$\binom{a}{b}$ denoting the binomial coefficient]. It satisfies the constraint⁴¹

$$\sum_{D \subseteq \mathcal{D}} \frac{1}{\eta(D)} \leq 1. \quad (7)$$

Then NFA defined by Eq. (8) depends on parameters $(v_D, |D|, \sigma, |X|)$: D is the researched subdomain, v_D directly derives from D and the image X , $|X|$, and σ only depends on the image X (respectively, the number of pixels of X and the second-order moment)

$$\text{NFA}(v_D, |D|, \sigma, |X|) = \eta(D) \frac{1}{\Gamma\left(\frac{|D|}{2}\right)} \int_0^{v_D/2\sigma^2} e^{-t} t^{\frac{|D|}{2}-1} dt, \quad (8)$$

where Γ is the Gamma function so that NFA values can be computed using the incomplete Gamma function $\mathbb{P}(a, x) = 1/\Gamma(a) \int_0^x e^{-t} t^{a-1} dt$ with $a = |D|/2$ and $x = v_D/2\sigma^2$.

3.3 Number of False Alarms Minimization

Equation (8) provides a criterion to evaluate the significance of a subset of pixels, denoted as D , in terms of possible cracks. Then among all subdomains D , we search the “best candidate” to be a crack or a set of cracks that is, according to the previous probabilistic measure, the one that minimizes $\text{NFA}(v_D, |D|, \sigma, |X|)$. In this study, the NFA criterion minimization is performed twice considering different subsets on subdomains D :

- In the first case, we aim at detecting some seeds of the crack(s) without considering any spatial constraint so that D may be any subset of pixels. Let us denote \hat{D}_s as the result of this first minimization.
- In the second case, we aim at reconstructing the crack (s) from the seed(s) taking into account the assumption of connectivity of the crack, so that D corresponds to connected subsets of pixels derived from seeds. Let us denote \hat{D}_{cc} as the result of this second minimization.

The minimization of Eq. (8) should be performed considering any D belonging to the set \mathcal{S}_D of the considered subdomains, which is generally intractable. For instance, in the first minimization \hat{D}_s is chosen among $2^{|X|}$ subdomains. However, fortunately, we do not have to consider all these subdomains in the minimization process since, given the cardinality of D , the NFA minimum value is achieved by the subdomain with the lowest SSV v_D . Hence, the minimization is performed in two steps:

1. For each cardinality $|D|$, find the minimal v_D values. In the absence of constraints on \mathcal{S}_D , a v_D minimum value is achieved by the $|D|$ minimal values of X . The output of this first step is thus a function $v_X: \mathbb{N} \rightarrow \mathbb{R}$.
2. Find the minimal NFA value varying $|D|$: $\text{NFA}_{\min} = \min_{n \in \{1, \dots, |X|\}} \text{NFA}[v_X(n), n, \sigma, |X|]$.

In order to illustrate NFA minimization, let us anticipate the results. Figure 5 shows the NFA curves and crack detection images in the case of the example shown in Fig. 3(a). The v_X function is plotted in Fig. 5(a) which also shows the NFA function $\text{NFA}: n \mapsto \text{NFA}[v_X(n), n, \sigma, |X|]$ versus n (representing $|D|$). We see that the transformation from v_D values to NFA values allows for the emergence of a minimum clearly indicating the cardinality of \hat{D}_s (6750 in the presented case). Knowing $|\hat{D}_s|$, \hat{D}_s of the pixels belonging to the seeds of the crack(s) itself is derived as the subset of $|\hat{D}_s|$ image lowest values. The latter are shown in Fig. 5(b), where we point out in red the first selected seed s_0 (visible in the upper-right part of the image). It was empirically chosen as the darkest pixel of the largest connected component of the image of the seeds.

Then from pixel s_0 , some connected paths are “pre-selected” (using DP as explained further) and along these paths, the SSV are computed. Figure 5(c) shows the “SSV” curves representing, for each D cardinality, the lowest SSV and NFA values. The “SSV” curves correspond to the lower

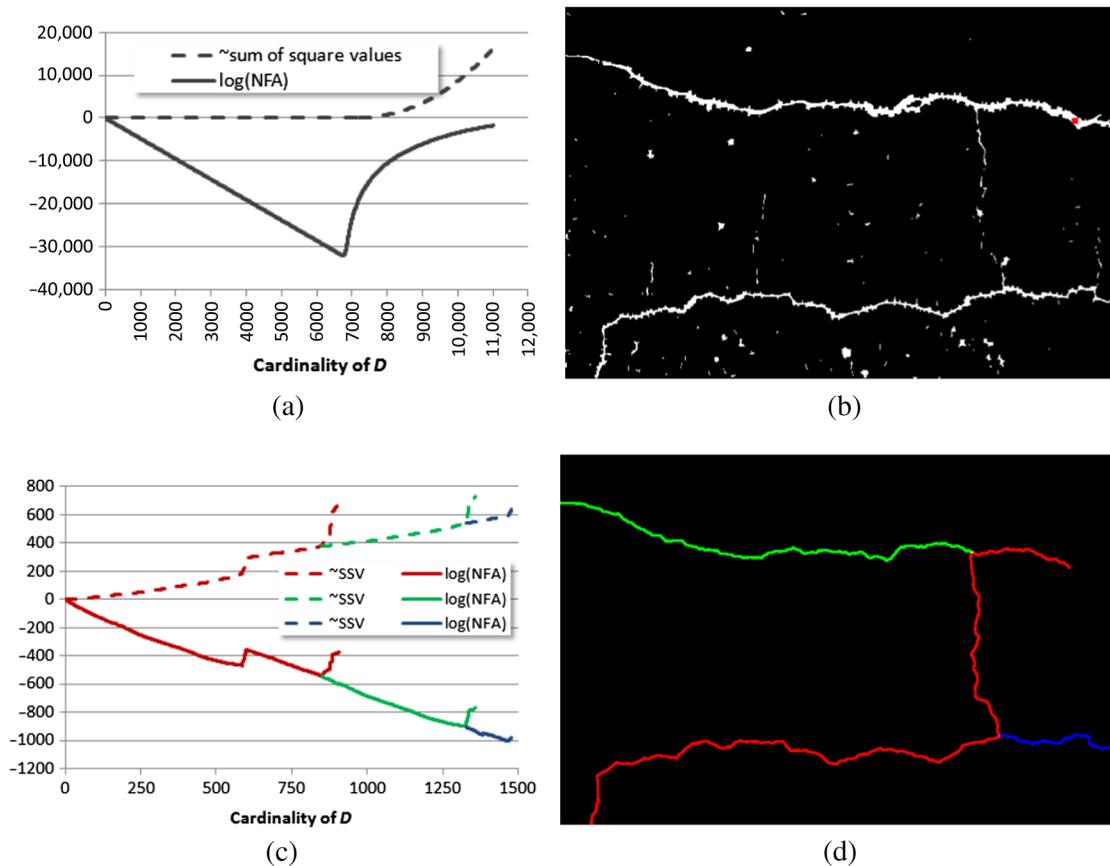


Fig. 5 Number of false alarms (NFA) curves and NFA minimization result in terms of subsets of pixels: (a, b) crack seed selection and (c, d) crack reconstruction from seed in red in Fig. 5(b) (seed located in the upper right branch).

envelope curves of the SSV and NFA functions computed on each path. Finally, the three colors correspond to estimated subparts of the cracks. Specifically, starting from s_0 seed [cf. Fig. 5(b)], the subset of connected pixels achieving the minimum NFA value has its cardinality equal to 850, as achieved with a path bifurcating from the upper crack to the lower one. However, until about 550, the path achieving the minimal NFA value was the one following the upper crack. Once this most significant path of the cracks is identified, other ramifications or branches are tested, e.g., the green one and the blue one in Figs. 5(c) and 5(d). As previously, the detection of the minimum NFA value allows us to consider only the subpart of each path corresponding to the cracks.

3.4 Path Selection

Let us now explain the mechanism of path preselection. We call a “path” a set of connected pixels of width 1 pixel. The basic idea is to define a crack in terms of a path. Indeed, even if a crack may be wider, its shape is mainly determined by its skeleton since the actual width of the crack could be estimated using postprocessing (e.g., adding some connected pixels basically selected from their gray level relative to the crack skeleton pixel values).

As explained in the previous section, NFA criterion is used to select the path that is the most likely to represent the crack or a subpart of the crack(s). Now, to keep the algorithm tractable, we do not compute NFA criterion on every subset of connected pixels, but on a subset of paths that is derived using DP. DP is a well-known technique^{44–46} that

allows us to find among the paths starting from a given graph node (pixel seed in our case) and ending at a given node (that may be any reachable pixel of the image in our case) the path with the minimal cost (SSV here). Note that such a path selection is heuristic since, theoretically, even if $v_{D_1} \geq v_{D_2}$, if $|D_1| \geq |D_2|$, $NFA(D_1)$ could be lower than $NFA(D_2)$. However, in practice, we do not observe such a situation and in order to guarantee the tractability of the crack detection, we compute the paths of interest using DP. Then for any pair of different paths arriving at a given pixel, only the path exhibiting the lowest v_D value is kept. In the following, these paths are denoted as “minimal” and their set is noted \mathcal{S}_p or $\mathcal{S}_{p(s)}$ when the seed s is specified.

Then NFA criterion is used first to decide the optimal path, noted \hat{p} , within \mathcal{S}_p and, second, to shorten \hat{p} if necessary by relaxing its first and last pixels. For any possible path length (i.e., between 1 and the maximal path length in \mathcal{S}_p), the minimal NFA value is achieved by the minimal z_D^2 value, hence the representation of the lower envelope in Fig. 5(c). The optimal path \hat{D} is those minimizing NFA versus the path length. Its last pixel is the pixel whose index along the path equals $|\hat{D}|$ minimizing NFA versus the path length. Finally, its first pixel is obtained by following the same path, but in reverse from the previously estimated last pixel. Once more this represents a simplified algorithm (heuristic) relative to the one that would test the significance of any subpart of \hat{D} , relaxing simultaneously the first and last pixels. However, this simplification is necessary to keep the algorithm tractable and the results prove it is efficient. Figure 6 shows

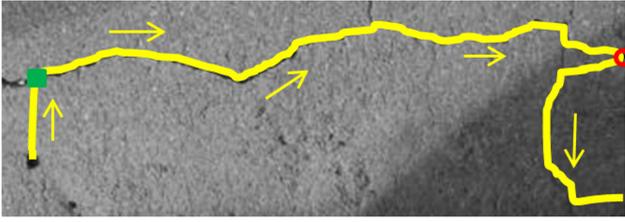


Fig. 6 Illustration of the need to estimate the first (green square) and the last (red circle) pixels of a selected path (in yellow) when the seed does not belong to the crack and the assumed maximal length of the path exceeds the actual size of the crack [Fig. 3(a)].

it in the case of the bottom left subpart of the image presented in Fig. 3(a). The considered seed (before the first arrow at the left of the image) is a defect in the (concrete) wall that does not actually belong to a crack, but due to its very low gray level values, may be confused with one when not considering the connectivity constraints. The identified path \hat{D} starts from this seed and then rejoins the actual path that it follows until its end. Finally note that in order to ensure the actual crack is a subset of the paths estimated by DP, all the computed paths in \mathcal{S}_p have a given length that is chosen to be very large. However, this implies the necessity to estimate the pixels of \hat{D} ending and beginning at the actual crack.

3.5 Algorithms

Algorithm 1 presents the estimation of the seeds. As stated previously, it is based on the ordering of the (square) values of image X that allow for the derivation of the minimal SSV

Algorithm 1 Estimation of the set of the seeds.

Data: (Positive) gray level image X ; parameter σ_0 ;

Result: Set of the seeds \hat{D}_s ;

Allocate a vector $v[]$ of dimension $|X|$;

Initialize $v[]$ elements with the square values of X ;

Order $v[]$ by increasing values ($\forall (i, j), i < j \Rightarrow v[i] \leq v[j]$);

Initialize the scalars NFA_{\min} to $+\infty$ and N to 0;

for $i = 1$ to $|X|$ **do**

$v[i] \leftarrow v[i] + v[i - 1]$;

Compute NFA_i using Eq. (8) and parameters $(v[i], i + 1, \sigma_0, |X|)$;

if $NFA_i < NFA_{\min}$ **then**

$NFA_{\min} \leftarrow NFA_i$;

$N \leftarrow i + 1$;

end

end

\hat{D}_s is the set of the N pixels with the lowest values in X .

v_D values (function v_X). Then only $|X|$ NFA values have to be computed, among which the minimum is searched.

Algorithm 2 presents the reconstruction of a crack (or a subpart of the set of cracks) from a given seed. Specifically, it considers the set of path \mathcal{S}_p derived by DP. SSV are computed along some subparts of these paths so that we introduce the following notations. For any path $p \in \mathcal{S}_p$, $|p|$ denotes its cardinality, i.e., number of pixels, and $\forall i \in \{0 \dots |p| - 1\}$ $p[i]$ denotes the i 'th pixel from its origin. Given the image X , we compute the value of a path denoted $v(p)$, as the SSV of its pixels $p[i]$ in X : $v(p) = \sum_{s \in p} [X(s)]^2 = \sum_{i \in \{0 \dots |p| - 1\}} [X(p[i])]^2$. In Algorithm 2, we also consider the value of a connected subpart of a path that is derived by summing the square values of the pixels of this subpart: denoting $v(p[i \dots j])$ as the value of path p between its i 'th pixel and its j 'th pixel: $v(p[i \dots j]) = \sum_{k \in \{i \dots j\}} [X(p[k])]^2$.

As explained previously, Algorithm 2 is heuristic. It involves three main steps: first, find the "optimal" path, noted as \hat{p} , among \mathcal{S}_p not considering NFA values but SSV from the origin pixel (given seed). The second step focuses on \hat{p} and, assuming the beginning pixel is the considered seed, finds the ending pixel, denoted $p[j_{\text{end}}]$, based on NFA criterion; third, given \hat{p} and $p[j_{\text{end}}]$, find the beginning pixel, denoted $p[j_{\text{begin}}]$, based on the NFA criterion. Additional notations are p_{index} to store the index of \hat{p} in \mathcal{S}_p and \mathcal{C} that contains the pixels already detected as belonging to the crack. Indeed, the three steps are repeated N_{branch} times in order to retrieve the different branches of a crack if there are several branches, or simply if the considered seed is not at an extremity of the crack. In order to actually consider different branches of the crack, at the end of each iteration, all the paths that are close to the already reconstructed part of the crack (stored in \mathcal{C}) are removed from the set of the paths to consider \mathcal{S}_p . In addition, v_0 and N parameters are updated. They represent the SSV and the cardinality of the already reconstructed parts of the crack, respectively. They are required to take a decision globally on the crack: since the previously detected part of the crack is stored in \mathcal{C} , only a path that allows the further decrease of the NFA measure is considered as a supplementary part of the actual crack(s). Then v_0 and N are both inputs and outputs of Algorithm 2, allowing its call several times during the whole crack set reconstruction (e.g., considering different seeds, see Algorithm 3).

Figures 5(c) and 5(d) show the first three iterations of the "repeat" loop in Algorithm 2: at the end of the first iteration, the minimal NFA value is around -543 and $\mathcal{C} = 850$ [red pixels in Fig. 5(d)]; at the end of second iteration, $NFA_{\min} \approx -901$ and $\mathcal{C} = 1326$ [green pixels in Fig. 5(d)]; at the end of the third iteration, $NFA_{\min} \approx -1005$ and $\mathcal{C} = 1467$ [blue pixels in Fig. 5(d)]; supplementary iterations do not allow us to further decrease NFA_{\min} so that \mathcal{C} does not change.

Algorithm 3 presents the proposed approach. First, we compute the ratio image $r_{\bar{\gamma}}$, and then we get the data image $Y_{\bar{\gamma}}$ used for crack detection. The set of seeds is obtained by Algorithm 1. Then for each considered seed s in an iterative way, the set of paths $\mathcal{S}_{p(s)}$ is derived by DP as the set of "minimal" paths (cf. Sec. 3.4). Knowing $\mathcal{S}_{p(s)}$, Algorithm 2 is then called to derive the (sub)part of the crack reachable from s . Finally, this value is added to the crack and, before reiteration, the set of the seeds is

Algorithm 2 Crack reconstruction from one seed.

Data: Gray level image X , set of paths \mathcal{S}_p starting from s ; initial values v_0 and N [SSV and cardinality of crack(s) already detected], parameters: σ_0 , N_{branch} ;

Result: Binary image B of the crack passing through seed s , updated value v_0 ;

$M \leftarrow \max_{p \in \mathcal{S}_p} |p|$;

Allocate vectors $v[]$ and $p_{\text{index}}[]$ with dimension equal to M ;

Initialize $v[i]$ elements and scalar NFA_{min} to $+\infty$;

Initialize indices j_{begin} and j_{end} to -1 ; Initialize \mathcal{C} (set of pixels) to \emptyset ;

repeat

for $p \in \mathcal{S}_p$ **do**

for $i \in \{1 \dots |p|\}$ **do**

if $v(p[0 \dots i]) < v[i]$ **then**

$v[i] \leftarrow v(p[0 \dots i])$;

$p_{\text{index}}[i] \leftarrow p$;

end

end

end

for $i \in \{1 \dots n\}$ **do**

$NFA \leftarrow NFA(v_0 + v[i], N + i + 1, \sigma_0, |X|)$ defined by Eq. (8);

if $NFA < NFA_{\text{min}}$ **then**

$NFA_{\text{min}} \leftarrow NFA$;

$\hat{p} \leftarrow p_{\text{index}}[i]$;

$j_{\text{end}} \leftarrow i$;

end

end

for $i \in \{1 \dots n\}$ **do**

$NFA \leftarrow NFA(v_0 + v(\hat{p}[i \dots j_{\text{end}}]), N + j_{\text{end}} + 1 - i, \sigma_0, |X|)$ defined by Eq. (8);

if $NFA < NFA_{\text{min}}$ **then**

$NFA_{\text{min}} \leftarrow NFA$;

$j_{\text{begin}} \leftarrow i$;

end

end

Add the pixels of path \hat{p} between indices j_{begin} and j_{end} to \mathcal{C} ;

$v_0 \leftarrow v_0 + v(\hat{p}[j_{\text{begin}} \dots j_{\text{end}}])$; $N \leftarrow N + j_{\text{end}} - j_{\text{begin}} + 1$;

Remove from \mathcal{S}_p the paths “too close” to \mathcal{C} ;

until N_{branch} times;

Derive B where the pixels of \mathcal{C} have value 1 and other pixels have value 0.

updated, removing any seed already put in the current crack estimation. In this study, the morphological operators used in algorithms are the classical ones,³⁷ but one could replace them by their more efficient implementation.⁴⁷

Figure 7 shows the iterations of Algorithm 3: the preprocessed image used for crack detection and estimation, namely image $Y_{\tilde{\gamma}}$, the binary image of the pixels likely to belong to the crack (in the absence of any connectivity constraint) among which the seeds are selected, and three successive iterations of the “while” loop in Algorithm 3.

Algorithm 3 Main crack detection algorithm.

Data: Gray level image I ; parameter(s): σ , steerable filter parameters;

Result: Binary image of the crack(s);

Perform background subtraction on I : $\tilde{I} = I - f_{G(\sigma)} * \phi(I)$;

Compute the image $r_{\tilde{\gamma}}$ according to Eqs. (2) and (4);

Compute the image $Y_{\tilde{\gamma}}$ according to Eq. (5);

Compute the binary image $I_{\hat{D}_s}$ of seeds from the set of the seeds \hat{D}_s estimated by Algorithm 1 with inputs $Y_{\tilde{\gamma}}$ and $\mathbb{E}[Y_{\tilde{\gamma}}^2]$;

Perform $I_{\hat{D}_s}$ dilation;

Compute the set of connected components S_{cc} of $I_{\hat{D}_s}$;

Initialize the set of crack pixels \mathcal{C} to \emptyset and the SSV v_0 to 0;

while S_{cc} **do**

Choose the next seed s (e.g., darkest pixel of largest remaining connected component in S_{cc} exhibiting an isoperimetric ratio greater than a threshold value);

Compute the set $\mathcal{S}_{p(s)}$ of minimal paths on $Y_{\tilde{\gamma}}$ image using DP from seed s ;

From Algorithm 2 with inputs $Y_{\tilde{\gamma}}$, $\mathcal{S}_{p(s)}$, v_0 , $|\mathcal{C}|$, $\mathbb{E}[Y_{\tilde{\gamma}}^2]$, and N_{branch} , compute the binary image B_s of the crack passing through s and the new value of v_0 ;

Update \mathcal{C} by adding the pixels with value 1 in B_s ;

Update S_{cc} by removing elements or subpart(s) of elements in the geodesic reconstruction of $I_{\hat{D}_s}$ from B_s ;

end

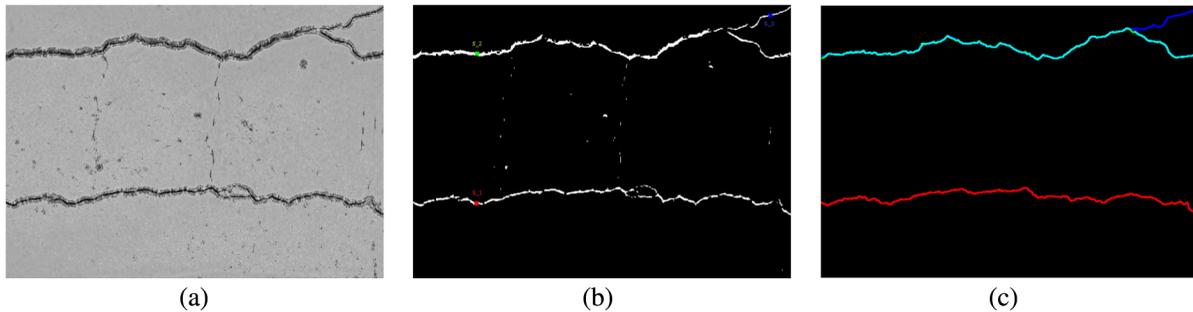


Fig. 7 Example of intermediate results provided by Algorithm 3: (a) gray level image Y_j , (b) I_{D_s} dilation with the different seeds (highlighted in color) selected iteratively, (c) subparts of the crack reconstructed iteratively: first iteration: red subpart obtained from red seed s_1 (bottom left), second iteration: cyan subpart obtained from green seed s_2 (top left), third iteration: both cyan and blue subparts obtained from blue seed s_3 (top right).

We note that after the first iteration, the bottom crack is retrieved; after the second iteration, most of the top crack is retrieved, but a branch is missing. Indeed, the position of the used seed [in green in Fig. 7(b)] is such that, after reconstructing the main part of the top crack, the missing branch appears less significant. However, the third iteration that starts from a seed located on this missing branch, finally allows the complete reconstruction of the top crack.

The parameters of the proposed approach are as follows:

- Parameters involved in the preprocessing step (derivation of \tilde{I} and Y_j), namely radius of the structuring element disk used for the morphological opening, standard deviation of the Gaussian filter for background estimation, standard deviation of the steerable filter [Eq. (2)], number of tested directions for ratio maximization [Eq. (4)], and weighting coefficient α in Eq. (5). The first three parameters are linked to the image resolution relative to the crack width. In our case, we set them to default values 2.0, 5.0, and 1.6, respectively, without tuning them to optimize our results. (We feel that these parameters are not sensitive.) In the same way, the number of tested directions is set to default value 4 (directions are, thus $\pm\pi/8$) that is sufficiently precise for the considered scale of the steerable filter. Finally, parameter α weighs the importance of the second-order derivative term relative to the basic image term. In our study, we set it to 0.5 as a compromise.
- Parameters involved in the crack detection from the image at the output of the preprocessing step, namely the number N_{branch} of branches tested in the crack reconstruction from one seed (Algorithm 2), the structuring element used for dilation of the seed image, and the isoperimetric ratio threshold used in the seed selection (Algorithm 3). Note that in Algorithms 1 and 2, all parameters but N_{branch} are automatically estimated. Between values 3 and 6, N_{branch} exhibits very limited sensitivity due to the possibility of considering several seeds. By default, the structuring element being used is the same as in the preprocessing step. Eventually, the threshold of the isoperimetric ratio aims to increase the probability of selecting a seed belonging to the actual crack(s) in order to save computation time. It is based on the same hypothesis used throughout the paper that cracks are longitudinal structures and basic tests show that a value around 4 provides good results.

However, we emphasize that none of these parameters (N_{branch} , structuring element, and isoperimetric ratio threshold) are very sensitive according to our tests.

4 Experimental Results

4.1 Experimental Data

For our experiments, we considered a crack dataset proposed by Fujita and Hamamoto¹ that allows for quantitative performance analysis due to ground truth data. Specific objects such as shadows, mold, and holes are systematically present beside cracks, and this allows us to highlight the reliability of the geometric assumptions that we implicitly consider in our approach. In order to degrade the image quality, we have blurred the images from the dataset using an algorithm for generating sudden motion blur⁴⁸ which allows us to incorporate highly realistic camera shake and uniform motion blur, and to control the amount of degradation applied. Various intensities of complex blur are introduced by varying the simulated exposure time T , which takes values from $T = 0$ (no blur added) to $T = 1$ s (see Ref. 48 for further details).

4.2 Qualitative Validation

Figure 8 shows the results obtained in the case of the original data (no blurring applied). For better examination, a subpart of the images is shown using a color composition as follows: the input image is in the blue channel, the crack estimation (binary result) is in the red channel, and the ground truth reference (binary result) is in the green channel. Thus, yellow pixels correspond to true positives (TP), pink pixels correspond to false positives (FP), and light green pixels correspond to false negatives (FN).

In this study, we compare our algorithm with methods using fundamentally different approaches: percolation, local anisotropy, and finally, FoSA which is based on a minimal path search at image level. For the two first methods, we provide both qualitative and quantitative results in Figs. 10 and 11, whereas for FoSA, we use the same data as Li et al.²⁶ (Fig. 12), which allows for a qualitative comparison with the results they report.

In Fig. 8, the results of three methods for crack estimation are presented in the three different columns. Regarding the proposed approach (second column), first of all we note the high number of TP, showing the efficiency of the proposed approach. However, two kinds of errors are present, depending on the considered image:

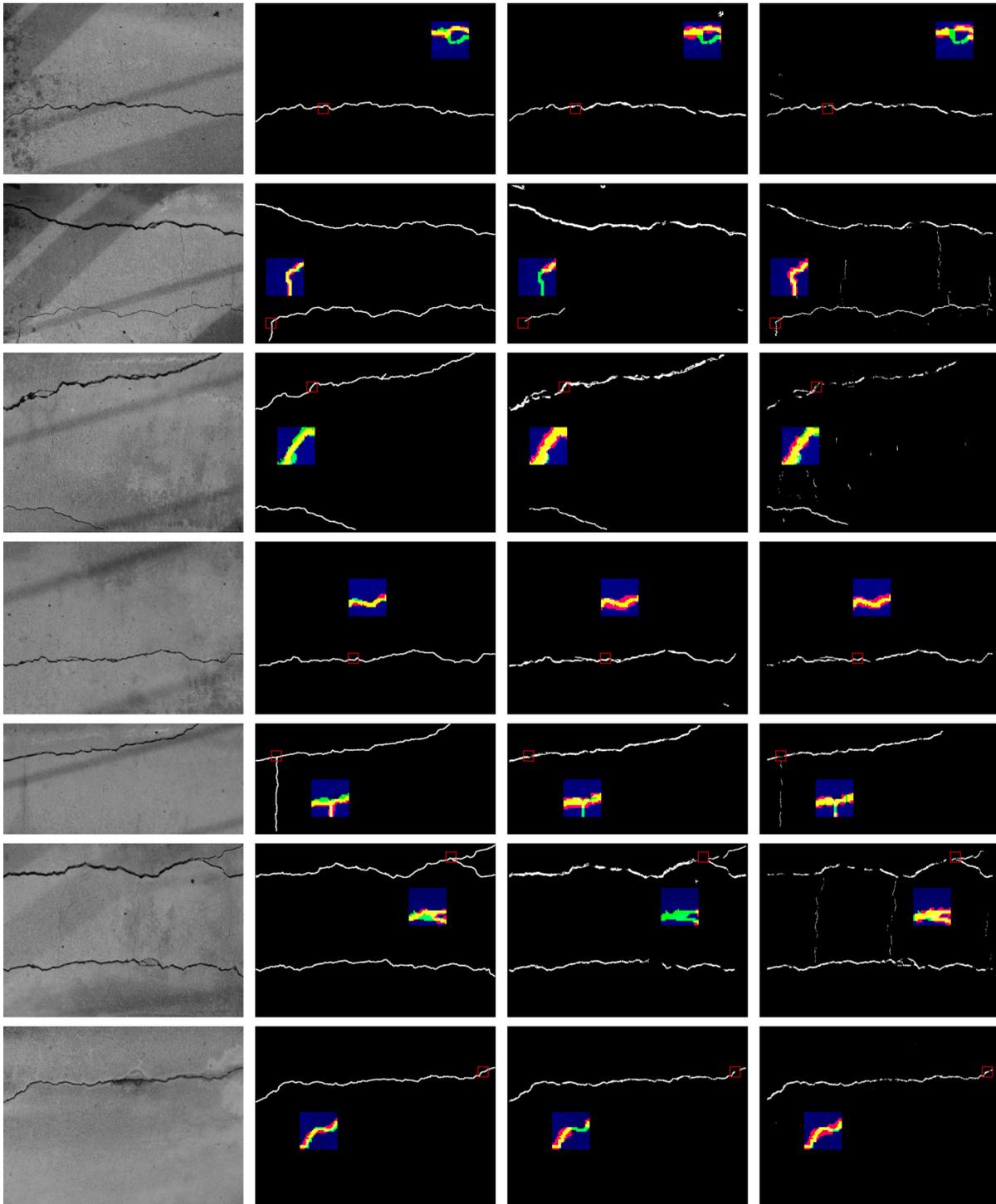


Fig. 8 Examples of the results obtained on the original images: the original images (first column), the proposed approach using NFA criterion (second column), the approach using FFA measure²² (third column), and the percolation approach⁴ (fourth column). Red squares highlight some areas which were magnified for a detailed visualization (the color code is specified in the text).

- The width of the crack may be not correct since with the proposed approach the estimated crack is always one-pixel width (or 3-pixels width for better visualization) independent of the actual crack width. In addition, slight differences may be observed between the achieved result and ground truth in terms of the location of the crack.
- Thin cracks are not always detected: e.g., the images in the first column and second and sixth lines present

some thin cracks which are missed, whereas in the case of the image on the fifth line, the thin crack is completely recovered.

Concerning the performance of the FFA and percolation methods, we note that both methods output good results provided we tune the most important parameters for the specific type of images used (in this case $d = 11$ for

FFA, and $N = 21$, $M = 41$, $w = 1$ for the percolation). One significant difference at this point is that FFA more often misses the finer structures (see, e.g., the second, fifth, or sixth rows).

Figure 9 is similar to Fig. 8 except that the input images are the blurred images with the highest considered intensity $T = 1$ s. Concerning the proposed approach using NFA

criterion (second column), main comments relative to the results are:

- Errors in the crack location are more important due to the decrease of image quality entailed by the blur effect.
- The number of undetected thin parts of cracks has increased due to the low-pass filter effect associated

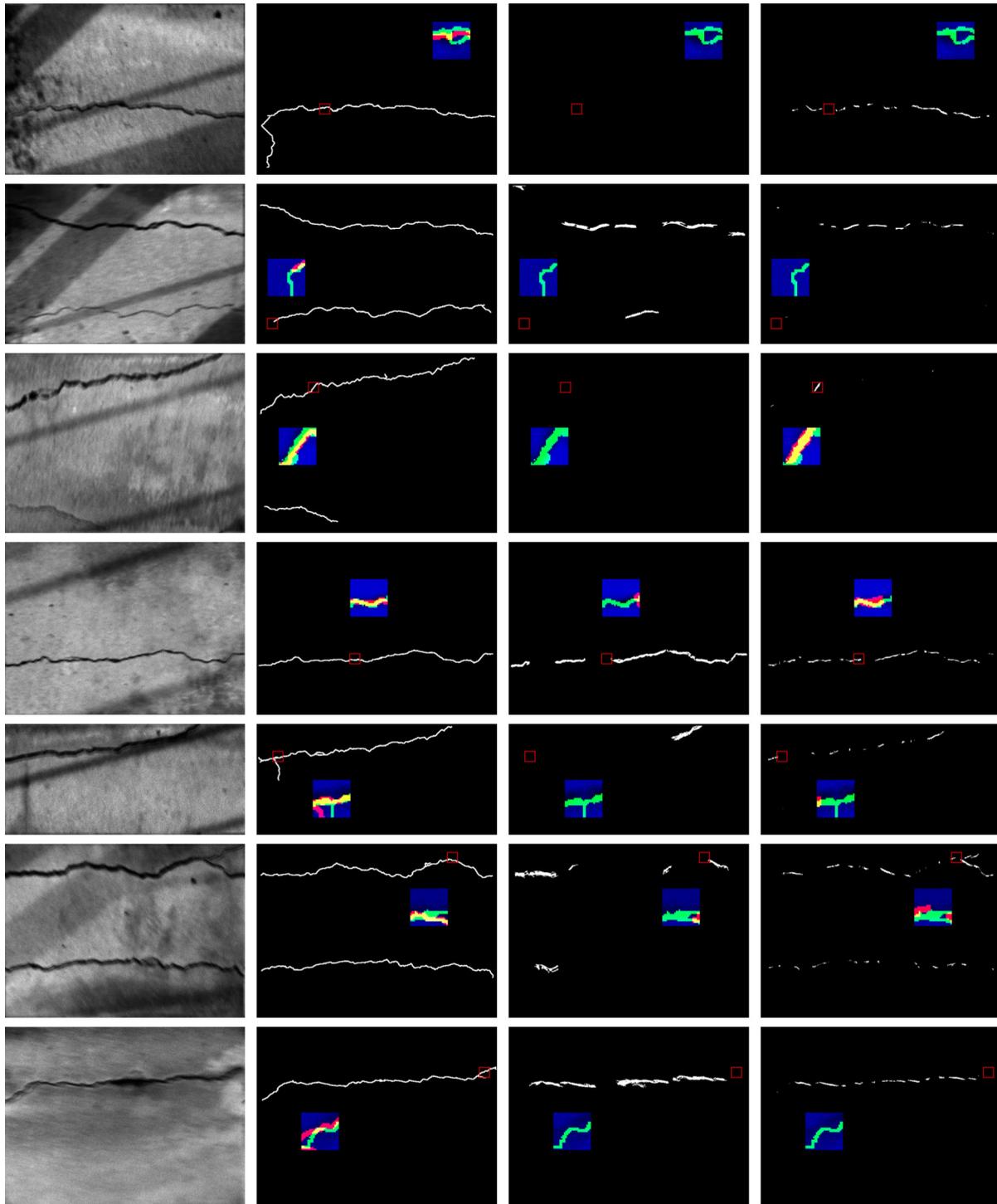


Fig. 9 Examples of the results obtained on the blurred images: the proposed approach using NFA criterion (first column), the approach using FFA measure²² (second column), and the percolation approach⁴ (third column). Red squares highlight some areas which were magnified for a detailed visualization (the color code is specified in the text).

with the blur effect; some fine details are indeed indistinguishable for a human at this point as a result of the blurring.

- One artifact appears in one case inducing FP.

However, we note that the main cracks are still detected correctly by our proposed method, while the number of substructures still being detected by FFA and percolation plummets.

4.3 Quantitative Validation

In order to quantify the performance of the proposed approach, we compute the classic indices “Precision” and “Recall.” They are defined based on the numbers FP, TP, and FN

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (9)$$

$$\text{Recall} = \frac{TP}{TP + FN}. \quad (10)$$

The precision aims at measuring the probability of good decisions among the estimated detections (positives), whereas the recall aims at measuring the probability of detection among the actual objects for detection (either detected or missed). In our case, the positives are the pixels belonging to the reconstructed cracks and they are true or false depending on the pixel label in the ground truth (cf. Sec. 4.1).

In order to distinguish between errors (FP or FN) due to a slight mislocation of the crack (cf. Fig. 8) and a nondetection of some parts of the cracks (or much more rarely, some

detection of defects other than cracks), we compute Precision and Recall at different scales. Specifically, TP and FP numbers are computed comparing a crack detection result with incremental dilations³⁷ of the GT image (so that positives are counted as FP only if, according to GT, there is no crack pixel within a radius equal to the scale parameter) and the FN number is computed comparing the GT image with incremental dilations of the result image (so that FN occurs only when GT crack pixels have no detection around them within a radius equal to the scale parameter).

Figure 10 shows the performance indicators versus scale for the proposed approach and for the other two methods previously depicted, either considering the original data or considering the blurred data. The plotted curves correspond to the Precision versus the Recall with the points on the curves obtained by varying the scale. Since the performance indices increase with the scale, the curve extremity point with lower coordinates was obtained at scale 0, whereas the other extremity point, showing the highest coordinates, was obtained for scale 9. We clearly note that the location error is less than 3 pixels, whatever the considered method, in the case of the original data since the Precision and Recall indicators increase noticeably when the scale increases between 0 (no dilation of the cracks before image comparison) and 2. The fact that the Recall does not reach about 100% even at large scales shows that there remain some parts of some very thin cracks that are not detected (cf. Fig. 8). However, significant cracks are all detected. In addition, using a criterion that minimizes the NFA, the number of FPs is very low so that Precision reaches 100%. In the case of the blurred images, using the proposed method, the Precision also exceeds 97%, but for larger scales because

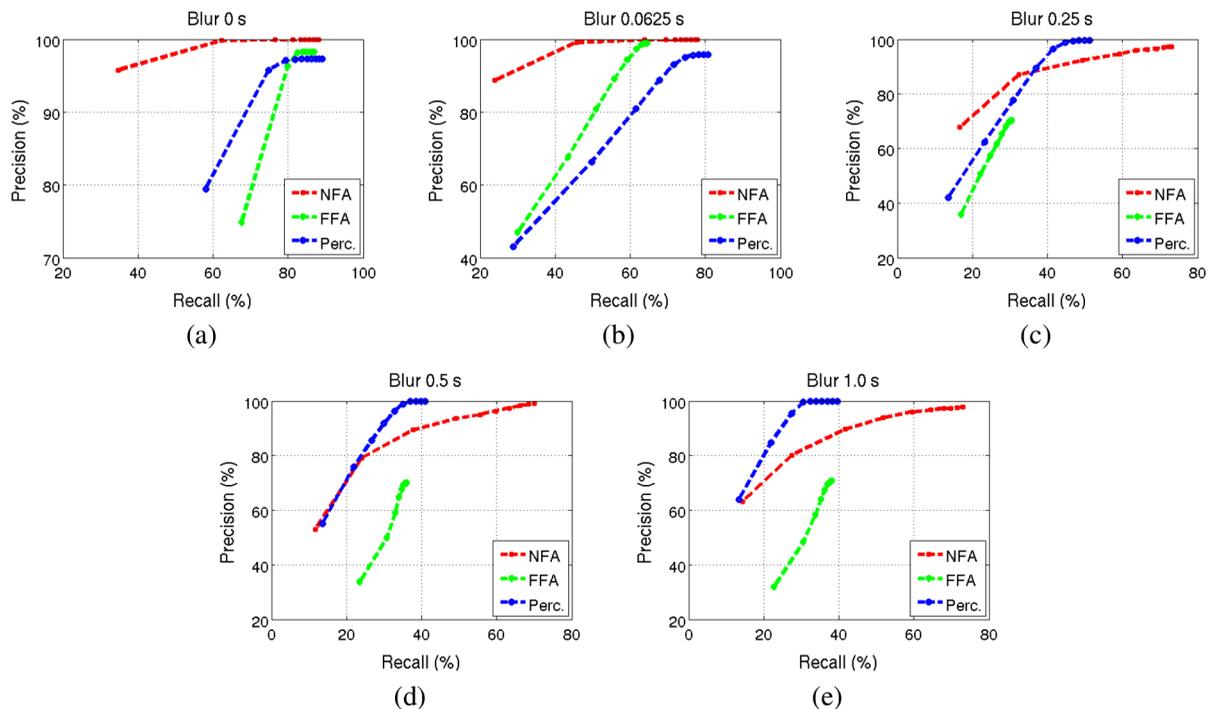


Fig. 10 Validation of the proposed approach (NFA) and comparison with alternatives (“percolation” and “FFA”) in terms of the Precision and Recall versus the scale of the analysis. The simulated exposure time T (in s) varies among the illustrations above, (a) case of original images ($T = 0$), (b) blur with $T = 0.0625$, (c) blur with $T = 0.25$, (d) blur with $T = 0.5$, and (e) blur with $T = 1$.

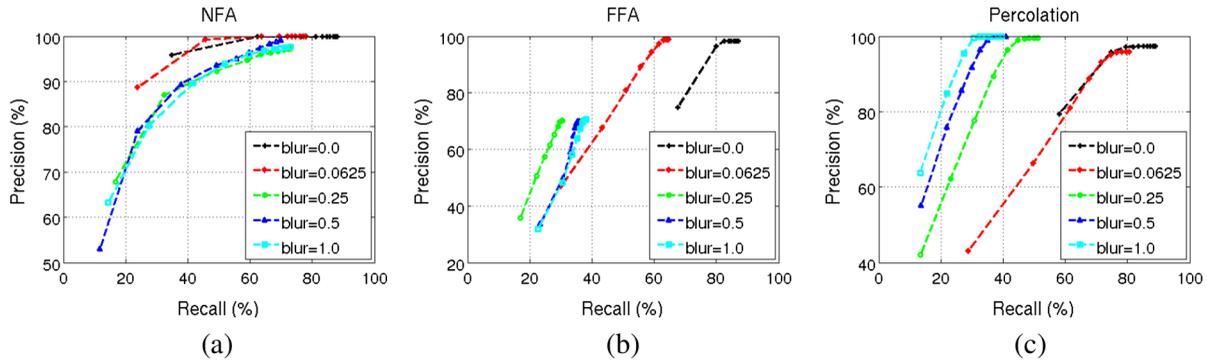


Fig. 11 Validation of the proposed approach and comparison with alternatives in terms of the Precision and Recall versus the scale of the analysis, and for varying levels of blur: (a) performance of the proposed method (NFA), (b) performance of “FFA,” and (c) performance of “percolation.”

of the intrinsic spatial imprecision introduced by the blur effect. In addition, the thin cracks are not detected at all now so that the Recall curve values are lower than before. In particular, for the largest blur ($T = 1$), the achieved value for large scales is around 73% instead of close to 88%. This means that about 15% of mostly the thinnest cracks are now missed. Looking at the performance obtained using the alternative methods, we note that their robustness to the blur effect is much lower: if for the lowest value of blur, they still reach an equivalent performance to the proposed method for large scales, in the case of more significant blur, the achieved values of Recall lower than about 50% clearly indicate that important parts of the cracks are missed. In addition, in some cases, the achieved values of Precision are also bounded (to about 70%), indicating a high rate of erroneous detections (other than cracks).

Alternatively, Fig. 11 comparatively shows the performances of the three methods we consider depending on the level of simulated exposure time, which relates directly to the image quality degradation. The evolution of the results for each method clearly shows clearly that while FFA and percolation are highly sensitive to blur with varying levels of performance loss, the proposed method exhibits robustness and a graceful degradation of performance. We also note that if the performance indices globally decrease with the level of blur, they are not strictly a monotonic function of the blur level. This shows us that the interpretation of blur may be complex: on the one hand, blur degrades the crack contrast and on the other hand, it acts as a low-pass filter for the image noise and small defaults other than crack. According to Fig. 11, it also seems that blur acts as a threshold effect since very similar performances are achieved for

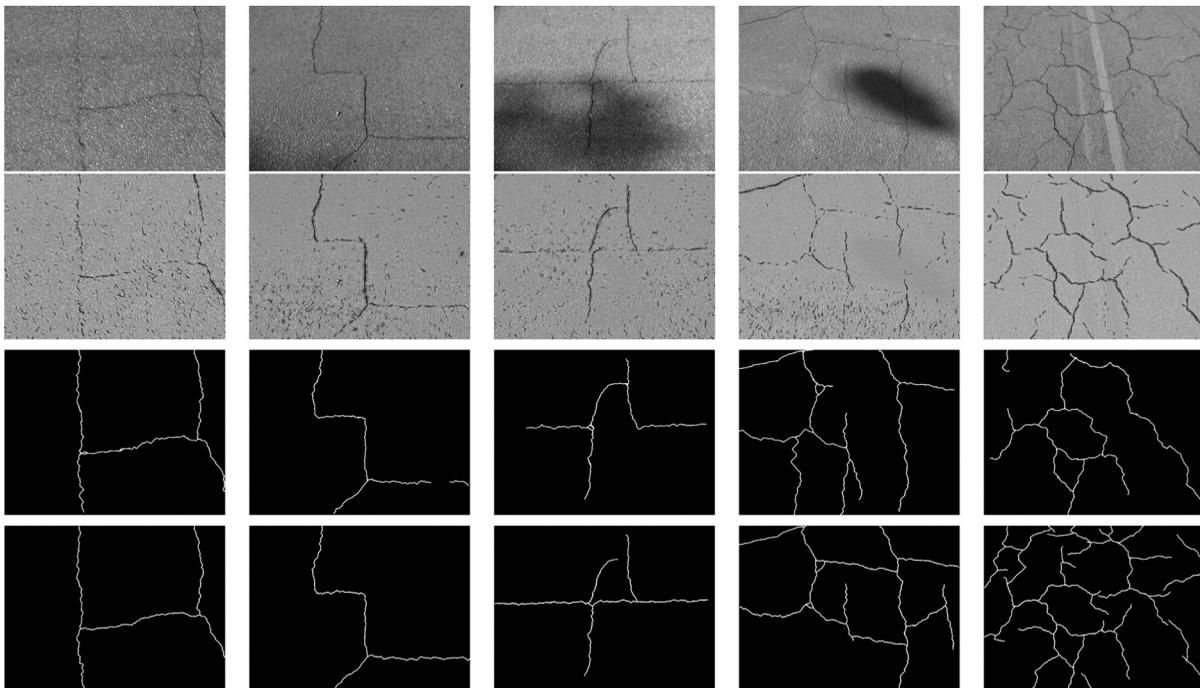


Fig. 12 Examples of the results obtained on the crack dataset³ with the original images shown on the first line, the composite image shown on the second line, the proposed approach results shown on the third line, and the ground truth images shown on the fourth line.

the first two curves ($T = 0$ s and $T = 0.0625$ s) on one hand, and for the three others ($T = 0.25$ s, $T = 0.5$ s, and $T = 1.0$ s) on the other hand.

4.4 Limits of the Proposed Approach

In order to further test our approach, we consider the dataset proposed in Ref. 3 that features road images with a significant, nonuniform texture. Indeed, second-order derivative filters may respond to such textures and thus penalize the detection of the crack itself. Figure 12 shows some results obtained on the images presented in Ref. 3 selected for comparison, plus an image much more complex (fourth column) exhibiting numerous cracks with variable orientations. The second, third, and fourth images are affected by shadow, but we see that it does not impact the crack detection. We also note that, when there are some errors, they are mainly FN (rather than FP) which is once more consistent with the fact that the used criterion is the minimization of the NFA.

Along the rows, the images are more or less ordered by decreasing performance: the crack is perfectly detected on the first image despite its very low contrast, a small subpart of the crack is missing on the second and third images despite the presence of a shadow, on the fourth image some crack subparts are missing and there are some FPs, and finally on the last image, only about half of the cracks are detected. These last two results allow for a deeper analysis. First, the textures on the image, mainly located on the bottom part on the image, probably due to the geometry of the acquisition, determine a response of the steerable filter similar to the crack response, and thus can generate some FPs when they are sufficiently close and dense. Second, when there are many cracks, our method will only focus on the most significant ones since adding some less significant cracks will not decrease the NFA value. This last point is an intrinsic limit of the proposed approach and was also the reason for some thin cracks not being detected in Fig. 8. However, when not “masked” by more obvious cracks, thin cracks can be detected as pointed out by the first example of this figure.

5 Conclusion

In this paper, we first introduced the topic of NDE in the context of aerial inspection. We briefly reviewed two widely used detection strategies, namely FFA and image percolation, and then argued that currently available algorithms did not include specific mechanisms that would allow them to cope with image quality degradation. Furthermore, we proposed a solution based on *a-contrario* modeling, which alleviates the need to define multiple thresholds in order to detect crack segments and reconnects them in varying conditions of image and structure (road/concrete) degradation. The experiments showed that compared to existing approaches, our method remained effective in the presence of complex blur, and was able to retrieve the structures of interest, except perhaps the finer details which were fully lost due to the blurring process. In perspective, we intend to explore adding higher level constraints on the alignment among the detected crack parts, while at the same time maintaining a low computational cost.

Future work will also focus on coupling this strategy with image restoration algorithms in order to further minimize

the impact of the blur on the result, and on implementing the algorithm efficiently on embedded architectures.

References

1. Y. Fujita and Y. Hamamoto, “A robust automatic crack detection method from noisy concrete surfaces,” *Mach. Vision Appl.* **22**(2), 245–254 (2011).
2. H. Oliveira and P. L. Correia, “Automatic road crack detection and characterization,” *IEEE Trans. Intell. Transp. Syst.* **14**(1), 155–168 (2013).
3. Q. Zou et al., “Cracktree: automatic crack detection from pavement images,” *Pattern Recognit. Lett.* **33**, 227–238 (2012).
4. T. Yamaguchi and S. Hashimoto, “Fast crack detection method for large-size concrete surface images using percolation-based image processing,” *Mach. Vision Appl.* **21**, 797–809 (2010).
5. M. R. Jahanshahi et al., “An innovative methodology for detection and quantification of cracks through incorporation of depth perception,” *Mach. Vision Appl.* **24**(2), 227–241 (2013).
6. C. Zhang and A. Elaksher, “An unmanned aerial vehicle-based imaging system for 3D measurement of unpaved road surface distresses,” *Comput. Aided Civ. Infrastruct. Eng.* **27**(2), 118–129 (2012).
7. I. Sa, S. Hrbar, and P. Corke, “Outdoor flight testing of a pole inspection UAV incorporating high-speed vision,” in *Field and Service Robotics*, L. Mejias, P. Corke, and J. Roberts, Eds., Springer Tracts in Advanced Robotics, Vol. 105, pp. 107–121, Springer (2015).
8. N. Metni and T. Hamel, “A UAV for bridge inspection: visual servoing control law with orientation limits,” *Autom. Constr.* **17**(1), 3–10 (2007).
9. R. Adhikari, O. Moselhi, and A. Bagchi, “Image-based retrieval of concrete crack properties for bridge inspection,” *Autom. Constr.* **39**, 180–194 (2014).
10. C. Eschmann et al., “Unmanned aircraft systems for remote building inspection and monitoring,” in *Proc. of the Sixth European Workshop on Structural Health Monitoring* (2012).
11. S. Varadharajan et al., “Vision for road inspection,” in *IEEE Winter Conf. on Applications of Computer Vision, Steamboat Springs, CO, USA, 24–26 March*, pp. 115–122 (2014).
12. M. Gavilán et al., “Adaptive road crack detection system by pavement classification,” *Sensors* **11**(10), 9628 (2011).
13. J.-K. Oh et al., “Bridge inspection robot system with machine vision,” *Autom. Constr.* **18**(7), 929–941 (2009).
14. S. Chambon and J.-M. Moliard, “Automatic road pavement assessment with image processing: review and comparison,” *Int. J. Geophys.* **2011**, 1–20 (2011).
15. S. K. Sinha and P. W. Fieguth, “Morphological segmentation and classification of underground pipe images,” *Mach. Vision Appl.* **17**(1), 21–31 (2006).
16. D. Aiger and H. Talbot, “The phase only transform for unsupervised surface defect detection,” in *2010 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 295–302, IEEE (2010).
17. J. Choi and C. Kim, “Unsupervised detection of surface defects: a two-step approach,” in *2012 19th IEEE Int. Conf. on Image Processing (ICIP)*, pp. 1037–1040 (2012).
18. H. Oliveira and P. Correia, “Crackit an image processing toolbox for crack detection and characterization,” in *2014 IEEE Int. Conf. on Image Processing (ICIP)*, pp. 798–802 (2014).
19. N. Batool and R. Chellappa, “Fast detection of facial wrinkles based on Gabor features using image morphology and geometric constraints,” *Pattern Recognit.* **48**(3), 642–658 (2015).
20. D.-M. Tsai et al., “A fast regularity measure for surface defect detection,” *Mach. Vision Appl.* **23**(5), 869–886 (2012).
21. W. Xu et al., “Pavement crack detection based on saliency and statistical features,” in *2013 20th IEEE Int. Conf. on Image Processing (ICIP)*, pp. 4093–4097 (2013).
22. M. Avila et al., “2D image based road pavement crack detection by calculating minimal path and dynamic programming,” in *IEEE Int. Conf. on Image Processing (ICIP)* (2014).
23. S. Chambon et al., “Road crack extraction with adapted filtering and Markov model-based segmentation: introduction and validation,” in *Int. Joint Conf. on Computer Vision Theory and Applications (VISAPP)* (2010).
24. T. S. Nguyen, “Fine structure extraction in textured images: application on automatic road crack detection,” Thesis, Université d’Orléans (2010).
25. R. Amhaz et al., “A new minimal path selection algorithm for automatic crack detection on pavement images,” in *2014 IEEE Int. Conf. on Image Processing (ICIP)*, pp. 788–792 (2014).
26. Q. Li et al., “Fosa: F* seed-growing approach for crack-line detection from pavement images,” *Image Vision Comput.* **29**(12), 861–872 (2011).
27. Q. Li and X. Liu, “Novel approach to pavement image segmentation based on neighboring difference histogram method,” in *Congress on Image and Signal Processing (CISP ’08)*, Vol. 2, pp. 792–796 (2008).
28. Q. Zou et al., “Cracktree: automatic crack detection from pavement images,” *Pattern Recognit. Lett.* **33**(3), 227–238 (2012).

29. E. Aldea and S. Le Hégarat, "Robust crack detection strategies for aerial inspection," *Proc. SPIE* **9534**, 953413 (2015).
30. D. Eberly et al., "Ridges for image analysis," *J. Math. Imaging Vision* **4**(4), 353–373 (1994).
31. T. Lindeberg, "Scale-space theory: a basic tool for analyzing structures at different scales," *J. Appl. Stat.* **21**(1–2), 225–270 (1994).
32. N. Karssemeyer and G. M. te Brake, "Detection of stellate distortions in mammograms," *IEEE Trans. Med. Imaging* **15**(5), 611–619 (1996).
33. N. Batool and R. Chellappa, "Modeling and detection of wrinkles in aging human faces using marked point processes," in *Computer Vision—ECCV 2012, Workshops and Demonstrations*, pp. 178–188, Springer (2012).
34. H. Heijmans, M. Buckley, and H. Talbot, "Path openings and closings," *J. Math. Imaging Vision* **22**(2–3), 107–119 (2005).
35. W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Trans. Pattern Anal. Mach. Intell.* **13**, 891–906 (1991).
36. S.-G. Jeong, Y. Tarabalka, and J. Zerubia, "Marked point process model for facial wrinkle detection," in *IEEE ICIP-Int. Conf. on Image Processing*, pp. 1391–1394 (2014).
37. J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, Orlando, Florida (1983).
38. A. Desolneux, L. Moisan, and J.-M. Morel, "Meaningful alignments," *Int. J. Comput. Vision* **40**, 7–23 (2000).
39. A. Desolneux, L. Moisan, and J.-M. Morel, "A grouping principle and four applications," *IEEE Trans. Pattern Anal. Mach. Intell.* **25**, 508–513 (2003).
40. A. Desolneux, L. Moisan, and J.-M. Morel, "Gestalt theory and computer vision," in *Seeing, Thinking and Knowing*, A. Carsetti, Ed., Theory and Decision Library A, Vol. **38**, pp. 71–101, Springer, Netherlands (2004).
41. A. Desolneux, L. Moisan, and J.-M. Morel, "From gestalt theory to image analysis: a probabilistic approach," in *Imaging, Vision, and Graphics, Interdisciplinary Applied Mathematics*, Vol. 34, pp. 1–276, Springer (2007).
42. M. Ammar et al., "An a-contrario approach for object detection in video sequence," *Int. J. Pure Appl. Math.* **89**(2), 173–201 (2013).
43. A. Robin, L. Moisan, and S. Le Hégarat-Masclé, "An a-contrario approach for subpixel change detection in satellite imagery," *IEEE Trans. Pattern Anal. Mach. Intell.* **32**, 1977–1993 (2010).
44. M. Sniedovich, "Dijkstra's algorithm revisited: the dynamic programming connexion," *Control Cybern.* **35**(3), 599 (2006).
45. M. Sniedovich, *Dynamic Programming: Foundations and Principles*, CRC Press (2010).
46. O. Chimi-Chiadjeu et al., "Automatic clod detection and boundary estimation from digital elevation model images using different approaches," *Catena* **118**, 73–83 (2014).
47. L. Najman and H. Talbot, Eds., *Mathematical Morphology: From Theory to Applications*, ISTE-Wiley, London, UK (2010).
48. G. Boracchi and A. Foi, "Modeling the performance of image restoration from motion blur," *IEEE Trans. Image Process.* **21**, 3502–3517 (2012).

Emanuel Aldea is an assistant professor at the Paris-Sud University. He received his BS and MS degrees in physics from Ecole Polytechnique in 2005 and from Paris 6 University in 2006, respectively, and his PhD in image processing from Télécom ParisTech in 2009. His current research interests include image processing and robotic vision for autonomous systems.

Sylvie Le Hégarat-Masclé received his PhD in signal and image processing from Télécom ParisTech in 1996 and the HDR degree from Versailles University in 2006. She is currently a professor of image processing at the Paris-Sud University. Her research activities focus on statistical pattern recognition (gestalt and structure detection), image analysis (classification, change detection), and data fusion using belief function theory. Application interests are for remote sensing and video scene analysis and understanding.