Construction and Building Materials 343 (2022) 128054

Contents lists available at ScienceDirect

Construction and Building Materials

journal homepage: www.elsevier.com/locate/conbuildmat

Determining crack kinematics from imaged crack patterns

B.G. Pantoja-Rosero [a], K.R.M. dos Santos [a], R. Achanta [b], A. Rezaie [a], K. Beyer [a,*]

[a] Earthquake Engineering and Structural Dynamics Laboratory (EESD), EPFL, 1015 Lausanne, Switzerland
[b] Swiss Data Science Center (SDSC), EPFL and ETH Zurich, 1015 Lausanne, Switzerland

ARTICLE INFO

Keywords:
Crack kinematics
Crack propagation
Damage assessment
Image-based kinematics
Crack-edge registration
Crack-displacement map

ABSTRACT

Determining the relationship between the cause of damage and the subsequent structural behavior of infrastructure systems requires an accurate characterization of the propagation of cracks, which represents the evolution of the damage state. When no information about the cause of damage is available, kinematic approaches can be used to describe the motion of crack contours. Current image-based approaches to derive crack kinematics use digital image correlation (DIC) on a set of sequential images as the crack propagates. However, DIC is invasive in that the structure surfaces must be painted with random speckle patterns, limiting its use primarily to controlled experiments. In this paper, we propose a novel image-based methodology for computing crack opening in Mode I or Mode II. As an input, this method takes a binary image from a semantic segmentation of an image of a crack pattern. This binary image is used to detect the opposite edges along the crack, which are then registered using an optimization algorithm based on the Euclidean transformation model and non-linear least squares. As a final output, this method produces displacement maps in the tangential and normal directions to the crack skeleton. To demonstrate its performance, we validate our methodology first with synthetic crack patterns and then with real crack patterns. Because this methodology for determining crack openings requires only simple data (just a binary crack pattern image), it is straightforward, robust, and adaptable, thus contributing to the development of structural image-based damage assessments. The computational codes and datasets are available to the public for future research and benchmarking on https://github.com/eesd-epfl/crack_kinematics and https://doi.org/10.5281/zenodo.6632071.

1. Introduction

Structural inspections of existing buildings and infrastructure comprise localizing and classifying damage features in the structure. These engineering inspections are conducted on a regular basis, as is common for infrastructure, as well as after extraordinary events, such as earthquakes. Among the various manifestations of damage that engineers evaluate, cracks are the most common feature in quasi-brittle structures, such as concrete and masonry [1].

The likely causes of damage and the mechanical properties of the damaged structural element (e.g., stiffness, strength) can be estimated from the crack pattern and kinematics of a crack. Crack kinematics describes how two crack surfaces move in relation to each other, and provide important information about the stress state that caused the damage. A Mode I crack opens normal to the crack surface and is typically associated with normal stresses. A Mode II crack opens parallel to the crack surface and is associated with shear stresses in the plane of the element [2]. Shear stresses acting in the crack plane cause Mode III deformation as well, but in this case the deformation is out-of-plane [2]. Concrete and masonry elements that fail in shear (Mode II) have a lower deformation capacity than elements that fail in flexure (Mode I). Therefore, identifying cracks in structures and classifying their opening mode is important for preventing potential failures through timely interventions designed to minimize economic losses, future deterioration, and even loss of human life [3].

Existing image analysis methods for detecting and identifying cracks and their patterns can be classified as heuristic feature extraction, change detection, and deep learning techniques [4]. Heuristic feature extraction employs a hand-crafted filter on the image, the output of which is post-processed using value thresholds or a machine learning classifier [5–7]. Change detection compares an image of the considered structure (baseline) to later images of the damage evolution [8–10]. Finally, the current state-of-the-art here employs deep learning techniques, in which a system is trained to automatically detect cracks in images using known data [11–13].

A common output of image-based crack detection is a binary mask image, in which the foreground pixels show the positions in the original

* Corresponding author.
E-mail addresses: bryan.pantojarosero@epfl.ch (B.G. Pantoja-Rosero), ketson.santos@epfl.ch (K.R.M. dos Santos), radhakrishna.achanta@epfl.ch (R. Achanta), amir.rezaie@epfl.ch (A. Rezaie), katrin.beyer@epfl.ch (K. Beyer).

https://doi.org/10.1016/j.conbuildmat.2022.128054
Received 16 December 2021; Received in revised form 31 May 2022; Accepted 2 June 2022
Available online 17 June 2022

image corresponding to cracks [14]. Post-processing of these binary masks is necessary to learn the characteristics of the cracks and their influence on the mechanical properties of structural elements. As one of the state-of-the-art applications of this technique, Rezaie et al. [15] used digital image correlation (DIC) data from damaged stone masonry walls to generate binary masks representing cracks, which were then post-processed to extract features like crack width. Today, the crack kinematics can be obtained using methodologies based on DIC data [15–20], but to the best of our knowledge, no existing method can evaluate the kinematics of a crack using only the binary mask as input data.

In this paper, we present a novel methodology to determine crack kinematics from a binary mask based on the registration of 2D point-clouds (or point-sets) along the length of a crack. To this aim, a binary mask with crack information is post-processed to extract its opposite contour as a 2D point cloud that can be registered via Euclidean transformation and optimization. Therefore, the information about the crack kinematics is encoded by the transformation obtained in the registration step. That transformation is then used to calculate the relative crack movement between opposing crack contours in terms of normal (Model I) and tangential (Model II) displacements.

In the following, we state the problem and provide a general description of our approach. We then describe the methodology and its components, beginning with the nonlinear least squares method, which forms the core of our method, and its adaptation to our specific problem. Furthermore, we describe some aspects of crack patterns and how we considered them in our algorithm, such as kinematic variability along the crack and multi-crack and multi-branched patterns. To validate and demonstrate the efficacy and robustness of our methodology, we present experiments taken from synthetic data and real data (including a building damaged due to an earthquake). To demonstrate the stability of our method and to provide guidance on the proper selection of the hyper-parameters involved in the algorithm, we run Monte Carlo simulations. The robustness of our code is evaluated by adding noise to the crack edges, which simulates material detachment from the crack edges. Experiments are carried out using real-world data, including a damaged stone masonry wall tested under shear compression loads and a building damaged by an earthquake. Finally, we present the conclusions reached during the methodology development process as well as future developments that can further improve the damage characterization research.

2. Problem statement

There are three different modes of crack propagation differentiated by analytical solid mechanics [21], as shown in Fig. 1. In Mode I, tensile stresses acting perpendicular to the crack plane cause the crack to open. Modes II and III are caused by perpendicular shear stresses acting parallel to the crack plane, causing in-plane deformation in the former case and out-of-plane deformation in the latter. In this paper, we will develop an approach to determine the kinematics of Mode I and Mode II cracks using only a binary image of the crack as input.

From previously developed approaches, an image of a crack pattern (Fig. 2a) can be segmented to obtain the crack as a binary mask image (Fig. 2b). In this binary mask, all pixels that belong to the crack are white and all other pixels are black. We herein develop a method to determine the crack mode and its change along the crack axis using only this binary mask. To accomplish this, we detect the two edges of the crack as 2D point-clouds (Fig. 2c). We divide one crack edge into smaller sets of points and determine the Euclidean transformation (rotation and translation) required to register this set of points onto a set of points that belongs to the second crack edge. The rotation and translation required to register the two point-clouds (point-sets) are then used to decompose the crack opening into a normal and a tangential component (Mode I and Mode II). The fracture mode is determined by the dominant deformation of these two movements. Mode III is not considered herein because the images only contain 2D information, and no out-of-plane shear deformation can be captured. Future work could use stereo images to include the determination of the crack motions for Mode III.

3. Methodology

This section describes the framework used to determine the crack kinematics from a binary crack mask. In this mask, the foreground pixels (white) in this segmentation correspond to the segmented crack, while the background pixels (black) are related to the material. Note that while the examples in the text show crack patterns on speckled surfaces, our methodology works for any situation for which a binary crack map can be obtained.

3.1. Nonlinear least squares

Nonlinear least squares is an optimization algorithm for fitting a nonlinear model to an observed dataset [23]. This method aims to find the parameters $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_n]$ of a model $f(x;\boldsymbol{\alpha})$ that minimize a loss function $L(x;\boldsymbol{\alpha})$ (Eq. (1)) defined as the mean squared error (MSE) of a residual function $r_i(\boldsymbol{\alpha}) = f(x_i;\boldsymbol{\alpha}) - y_i$ at N discrete points x_i in the domain, where y_i is the observed data corresponding to x_i:

$$L(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{1}^{N} r_i^2(\boldsymbol{\alpha}). \tag{1}$$

Therefore, the optimization problem can be written as

$$\boldsymbol{\alpha}_{opt} = \arg\min_{\boldsymbol{\alpha}} \; L(\boldsymbol{\alpha}), \tag{2}$$

where $\boldsymbol{\alpha}_{opt}$ is the set of parameters minimizing $L(\boldsymbol{\alpha})$. To solve the minimization problem in Eq. (2), the Gauss–Newton method is commonly used. In this method, the optimal set of parameters $\boldsymbol{\alpha}_{opt}$ is found through variations of the elements of $\boldsymbol{\alpha}$ in the direction determined by the step $\Delta\boldsymbol{\alpha}$, obtained iteratively as shown in Eq. (3) (for iteration $n+1$). More detailed information is presented in Betts et al. [24].

$$\boldsymbol{\alpha}_{n+1} = \boldsymbol{\alpha}_n - \gamma\Delta\boldsymbol{\alpha}, \tag{3}$$

where γ is the learning rate controlling the step size, and the step $\Delta\boldsymbol{\alpha}$ is given by

$$\Delta\boldsymbol{\alpha} = \mathbf{H}[L(\boldsymbol{\alpha})]^{-1}\nabla L(\boldsymbol{\alpha}), \tag{4}$$

where $\mathbf{H}[L(\boldsymbol{\alpha})]$ is the Hessian of the loss function, and $\nabla L(\boldsymbol{\alpha})$ is its gradient.

3.2. Solving for the crack kinematics

In this section, we solve the 2D point-set registration problem by transforming one set of points that belongs to one crack edge to fit over another set of points that belongs to the other crack edge. The contours are detected from the binary mask using the marching squares method [25], and they are divided into two sets of points, each representing one edge of the crack, as shown in Fig. 2c. The pixels selected here as crack contour belong to the image background rather than the crack, meaning they are part of the set of black pixels in Fig. 2b. The two edges of a crack segment are represented by the sets of points $\mathbf{E}_0 = [e_0^{(0)}, \dots, e_N^{(0)}]$ and $\mathbf{E}_1 = [e_0^{(1)}, \dots, e_n^{(1)}]$, respectively. Further, each element e_i is a coordinate point in $\mathbb{R}^2$.

The method developed herein assumes that a crack contour possesses three degrees-of-freedom (DOF), which are associated with a vertical and a horizontal translation $\mathbf{t} = (t_x, t_y)^T$ and a rotation θ. Thus, our goal is to find the parameters $\mathbf{T} = [t_x, t_y, \theta]$ transforming the edge $\mathbf{E}_0 \Rightarrow \mathbf{E}_0'$, such that the Euclidean distance between the translated set of points $\mathbf{E}_0'$ and a set of points on the second crack edge $\mathbf{E}_1$ is minimized. $\mathbf{E}_0'$ is given by

$$\mathbf{E}_0' = \mathbf{R}(\theta)\mathbf{E}_0 + \mathbf{t}, \tag{5}$$

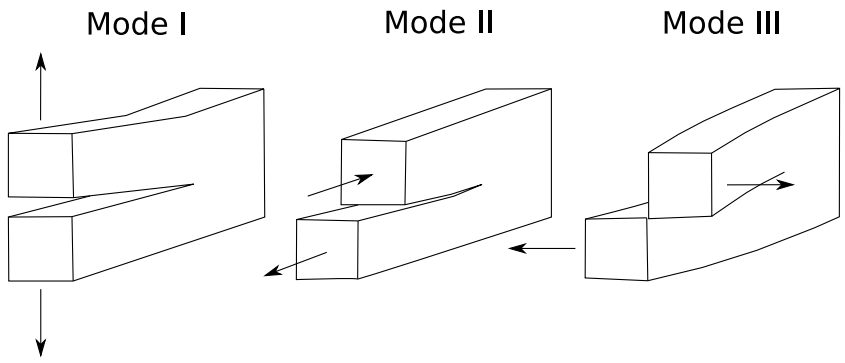


Fig. 1. Fracture propagation modes [22].

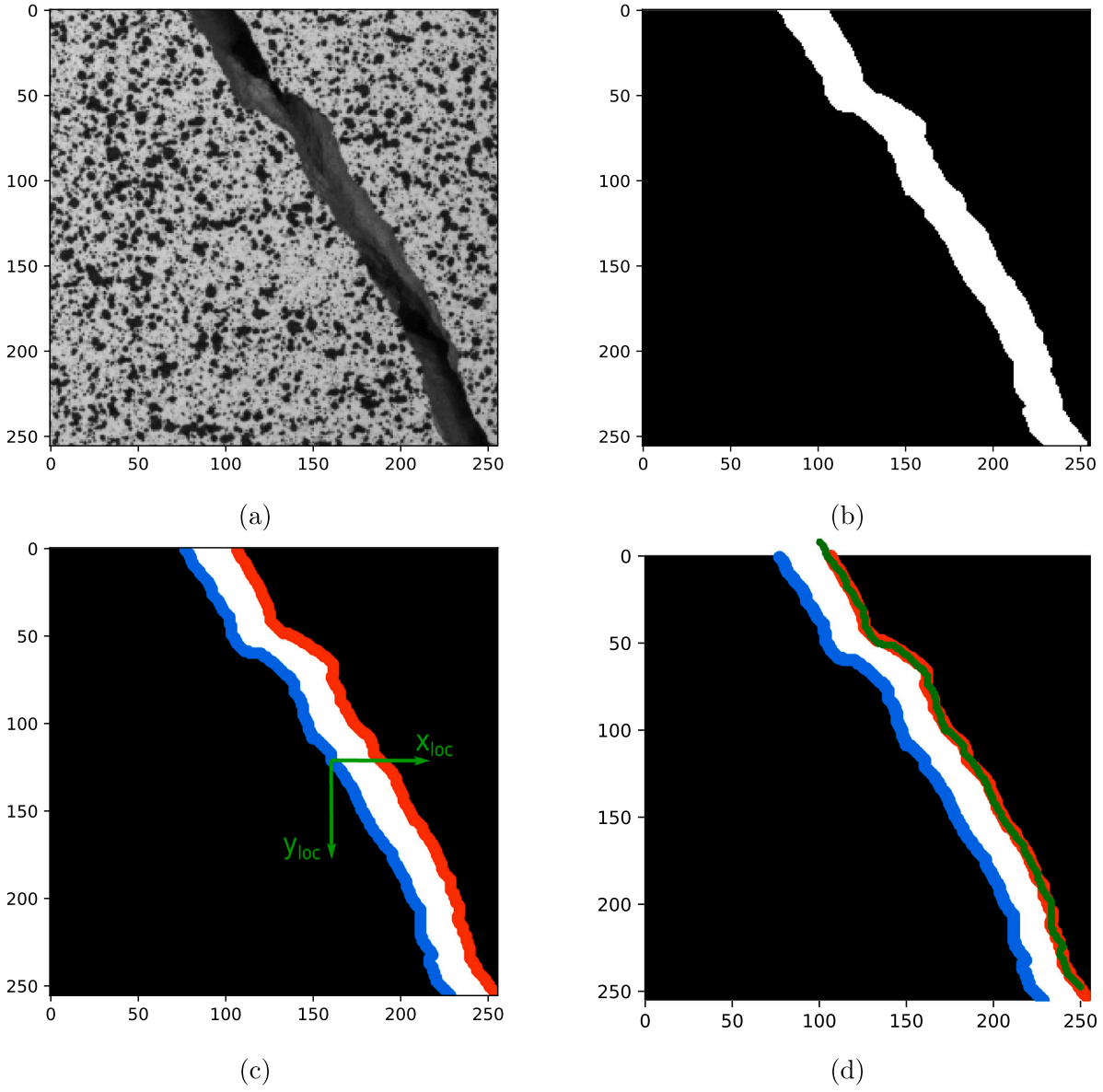


Fig. 2. Image of a single crack (256 × 256 px) considering the (a) crack pattern image, (b) detected binary mask, (c) identified crack edges in red and blue, and (d) transformed edge (blue) after applying Euclidean transformation (green). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where $\mathbf{R}(\theta)$ is the rotation matrix defined as

$$\mathbf{R}(\theta) = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}, \tag{6}$$

and $\mathbf{t}$ is the translation vector. Next, the residual function is defined as

$$r(\mathbf{T}) = (\mathbf{R}(\theta)\mathbf{E}_0 + \mathbf{t}) - \mathbf{E}_1. \tag{7}$$

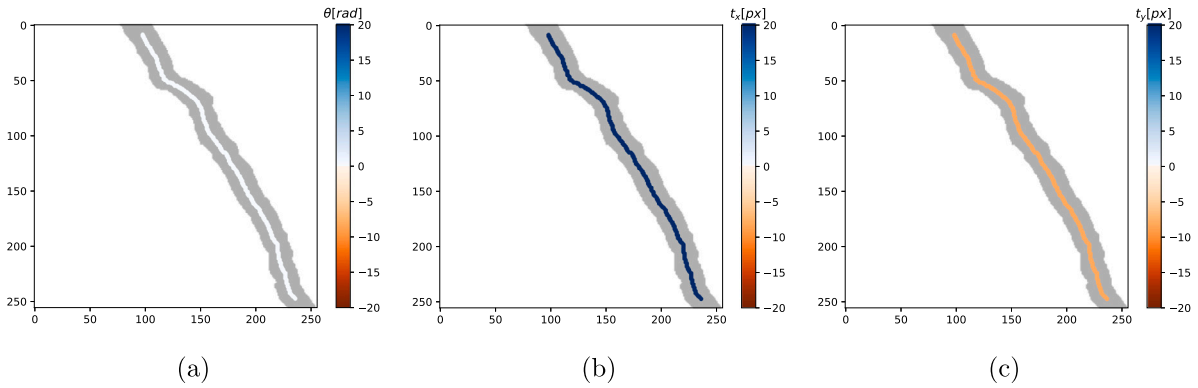


Fig. 3. Displacements maps for crack transformations using full edges: (a) rotation θ, (b) horizontal translation t_x, and (c) vertical translation t_y.

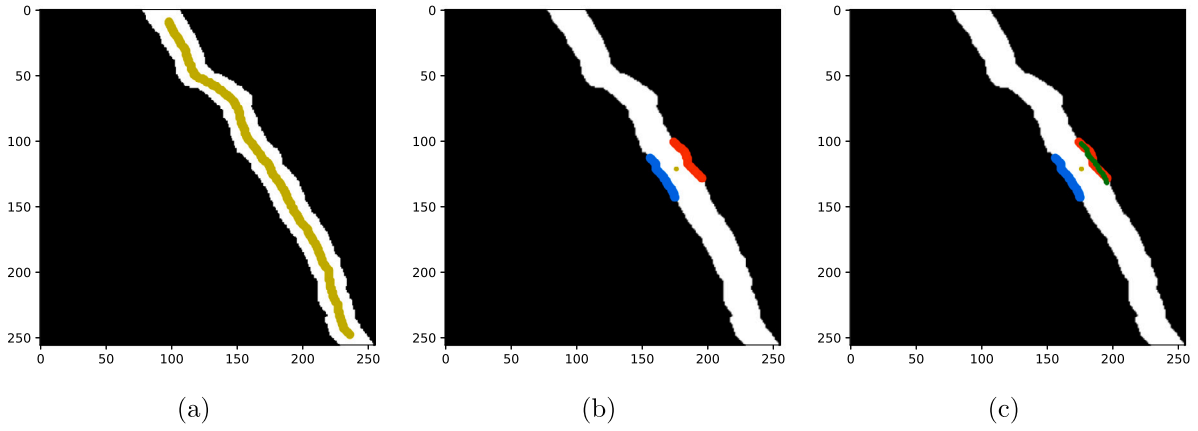


Fig. 4. Finite segment edge registration: (a) crack skeleton, (b) the finite edge elements corresponding to $\mathbf{E}_0$ (blue) and $\mathbf{E}_1$ (red) determined from a single pixel (yellow dot) of the crack skeleton, and (c) the finite edge segment of $\mathbf{E}_0$ after the optimal transformation over a finite edge segment of $\mathbf{E}_1$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Herein, the nonlinear least square method is used to find the transformation $\mathbf{T} = [t_x, t_y, \theta]$ that determines the crack kinematics. Fig. 2d shows the results after applying the transformation defined by the optimal parameters $\mathbf{T} = [t_x, t_y, \theta]$ to $\mathbf{E}_0$. Additionally, Fig. 3a–c show the numerical values of the transformation $\mathbf{T} = [t_x, t_y, \theta] = [22.25\text{ px}, -8.18\text{ px}, 6.61 \times 10^{-3}\text{ rad}]$ as displacements maps along the crack skeleton, which is obtained by the skeletonization methods developed previously [26,27]. Note that the image coordinates (global coordinates) of the edges are different from the coordinates used in the transformation (local coordinates), whose origin we place at the mean of the coordinates of the point set $\mathbf{E}_0$. To determine their local coordinates, we subtract this mean coordinate from the global coordinates of $\mathbf{E}_0$ and $\mathbf{E}_1$. Fig. 2c shows the origin of the edges' local coordinate system with respect to which the transformation is performed.

3.2.1. Variability of kinematics along the crack

As the crack edges are not rigid, the translation and rotation that registers the crack edge segment of one edge onto the other edge is not always constant along the crack axis. To account for this effect, we decompose the crack edges into edge segments composed of a reduced set of points. To define these edge segments, we find the k-neighboring points in $\mathbf{E}_0$ and $\mathbf{E}_1$ for each point $\mathbf{x}_{sk}$ of the set of crack skeleton points $\mathbf{X}_{sk}$ (Fig. 4a). Fig. 4b depicts a single pixel (yellow dot) of the skeleton in Fig. 4a and its k-neighbor points in $\mathbf{E}_0$ and $\mathbf{E}_1$ for $k = 50$. The same optimization methodology described in the preceding section is then applied to the finite edge segments (Fig. 4b,c) to find the local crack edge translation and rotation.

After repeating this procedure for each skeleton pixel, the non-constant movement of the crack is found, as shown in Fig. 5a–c. In this example, the mean values of all three DOFs are $\theta = 53.03 \times 10^{-2}$ rad, $t_x = 20.17$ px, and $t_y = -10.87$ px, which differ from the values obtained using full edges and demonstrate the localized behavior of the crack movement.

Next, we represent the local crack movement using only two DOFs, namely t'_x and t'_y. The axes x' and y' correspond to the local crack coordinate system and are parallel to the x and y global axes. The displacements corresponding to t'_x and t'_y are estimated from the mean of the difference in coordinates of the $\mathbf{E}_0$ k-neighbors before and after the transformation:

$$\mathbf{t}' = \begin{pmatrix} t'_x \\ t'_y \end{pmatrix} = \frac{1}{k} \sum_{i=1}^{k} \left[\mathbf{e}_i'^{(0)} - \mathbf{e}_i^{(0)} \right]. \tag{8}$$

Later in this paper, t'_x and t'_y will be used to find the displacements parallel and orthogonal to the crack surface, which define the crack kinematics and determine the opening mode of the crack (Mode I, Mode II, or mixed mode—Mode I and Mode II combined).

3.2.2. Edge registration over an extended region of $\mathbf{E}_1$

If the crack opening has a significant Mode II component, the k points on $\mathbf{E}_0$ that are the k-nearest neighbors to a selected point $\mathbf{x}_{sk}$ on the crack skeleton will not correspond to the k-neighbor points on $\mathbf{E}_1$. In this case, we must register k points of $\mathbf{E}_0$ on a segment of $\mathbf{E}_1$ that contains more than k points. In the following, we use a greedy approach to register the selected k points from $\mathbf{E}_0$ on all the possible groups of k consecutive points in $\mathbf{E}_1$. The transformation with the lowest error is then selected as the correct transformation. However, this procedure is computationally expensive. To overcome this limitation, we propose two solutions. In the first, we define two hyper-parameters, μ and λ,

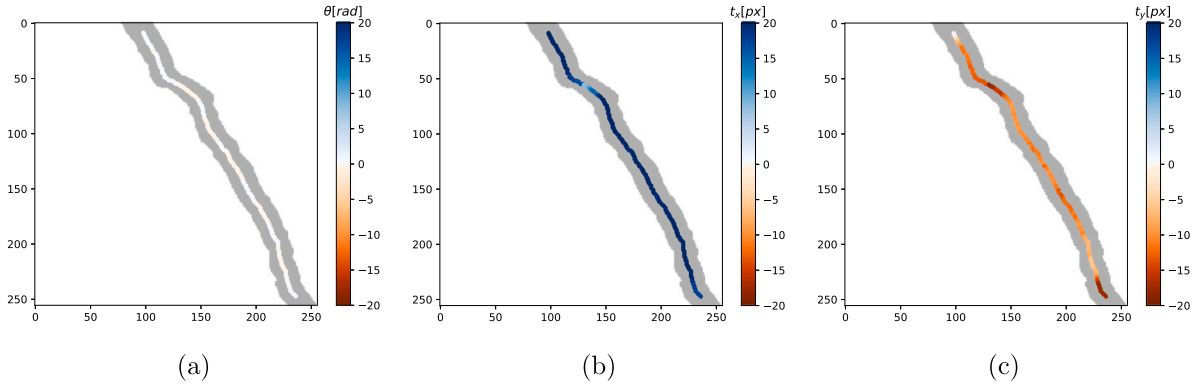


Fig. 5. Displacement maps indicating the movement of finite segments of crack edges: (a) crack rotation map θ, (b) crack horizontal translation map t_x, and (c) crack horizontal translation map t_y.

that describe the selection of groups of k points in $\mathbf{E}_1$ registered with the k points in $\mathbf{E}_0$. First, μk is the size of $\mathbf{E}_1$ in which we search for the k points of $\mathbf{E}_0$ (Fig. 6a). In this case, we select various groups of k points (M groups in total) from the μk points of $\mathbf{E}_1$, where the first group composed of the first consecutive k points (the initial point is one of the extremes of the μk points; Fig. 6b). The initial point of the second group of k consecutive points is placed at the λth position (Fig. 6c). The third group starts at the 2λth point, and so on, for the other groups until all the μk fractions of $\mathbf{E}_1$ have been covered. Thus, M transformations are obtained from which we select the one with the lowest value in the loss function. Fig. 6d shows the optimal registration of k points of $\mathbf{E}_0$ with μk points in $\mathbf{E}_1$. It can be seen here that the optimal registration did not occur with the k-nearest neighbor points in $\mathbf{E}_1$, but instead occurred after considering a group of k points located close to the end of the chosen fraction μk of $\mathbf{E}_1$.

The second solution we propose for registering the $\mathbf{E}_0$ edge segment on the extended $\mathbf{E}_1$ is the use of heuristic algorithms to find the optimal transformation among the various possibilities. We specifically employ the global optimization algorithm proposed in Shaqfa et al. [28], which is based on Pareto-like sampling. This sampling algorithm selects a number of uniformly distributed random samples from the variable domain λ (we select 5% of the discrete domain as samples) in which the loss function is evaluated. The sample that produced the lowest loss value generates a tentative smaller interval inside the variable domain where the optimal value might be placed with a defined probability (90% for our problem). The remaining chances of containing the optimal solution are assigned to the rest of the domain. The algorithm runs a defined number of iterations (five in our case), selecting one of the two intervals following the assigned probabilities, generating new random samples, evaluating the loss function in them, and selecting the best variable value.

3.2.3. Multiple branches and cracks—hyper-parameter η

In this section, we propose a methodology to assess the crack kinematics for more complex patterns. To ensure that the k-neighbors points come from two opposing crack edges, endpoints and junctions are used to segment edges of cracks with multiple branches and/or multiple isolated cracks in the same image. Fig. 7a,b,e,f show two different cracks with the described typology and their corresponding contour detection. As the developed algorithm considers each pixel in the skeleton to detect the two closest contours and take the k-neighbors points, it is necessary to divide those contours into multiple edges. To that end, we detect junctions and endpoints in the skeleton using the Mahotas hit and miss algorithm [29] with kernels of the main typologies of junctions and endpoints as shown in Fig. 7c,g. Then, the closest edges to each of these points are halved at the closest point, as shown in Fig. 7d,h.

Because the size of the segmented edges varies, the k parameter should be chosen accordingly. Instead of k, we propose using a related hyper-parameter η, which represents a fraction of the full length of a segmented edge and is related to k as follows: $k = Length(\mathbf{E}_i)/\eta$. Through this parameter, we ensure that the number of neighbors k adapts to the edge size when the pattern contains multiple branches and/or isolated cracks.

3.2.4. Crack kinematics—normal and tangential deformations for Modes I and II

To determine the opening in Mode I and Mode II, it is necessary to express the crack opening in tangential and normal movements in relation to the local crack axis. However, the information about crack kinematics contained in the horizontal displacement t'_x and vertical displacement t'_y is not meaningful from a mechanical point of view because it is expressed in the horizontal and vertical coordinate system of the image, as shown in Fig. 8a. To translate this to the local crack axis, we begin by determining the normal and tangential directions along the length of the skeleton (given by the angle β), defining a local coordinate system for each of its pixels (parallel and perpendicular axes with respect to a skeleton pixel; Fig. 8b). We use principal component analysis (PCA) in k-nearest neighbor points (ten neighbors for our experiments, $k_\beta = 10$) to determine both tangential and normal directions for each skeleton pixel. Note that when computing the normal angles for the skeleton, we force them to fall into either the first or the second Cartesian quadrants ($0 \leq \beta \leq \pi$; if angles fall in other quadrants, we add or subtract π) and assign this as the positive direction of the normal axis n. Then, the axis t is selected to be perpendicular to n following the right-hand rule and after determining the direction of the local axis (β) for each pixel, as shown in Fig. 8c,d, the normal and tangential movements (t_n, t_t) of the crack can be calculated using t'_x and t'_y as follows:

$$t_t = -t'_x \sin(\beta) + t'_y \cos(\beta), \qquad t_n = t'_x \cos(\beta) + t'_y \sin(\beta). \tag{9}$$

After computing the normal and tangential deformations of the crack along the axis of the skeleton, we obtain the displacement maps presented in Fig. 9.

3.2.5. Interpreting displacement maps

We use displacement maps to display the relative tangential and normal movement of the cracks, and for a better interpretation of the results, we propose a sign convention for their values. Here, we define a positive normal displacement when the crack opens and a positive tangential displacement when the relative movements of the crack edges rotate counterclockwise. This can be visualized in Fig. 9, where the displacements along the crack are represented with oriented vectors. For positive values of normal displacements, the oriented vectors align and point towards opposite directions (Fig. 9a). For positive values of tangential displacements, the oriented vectors placed at the opposite sides of the crack show counterclockwise rotation; negative values show clockwise rotation. (Fig. 9b).

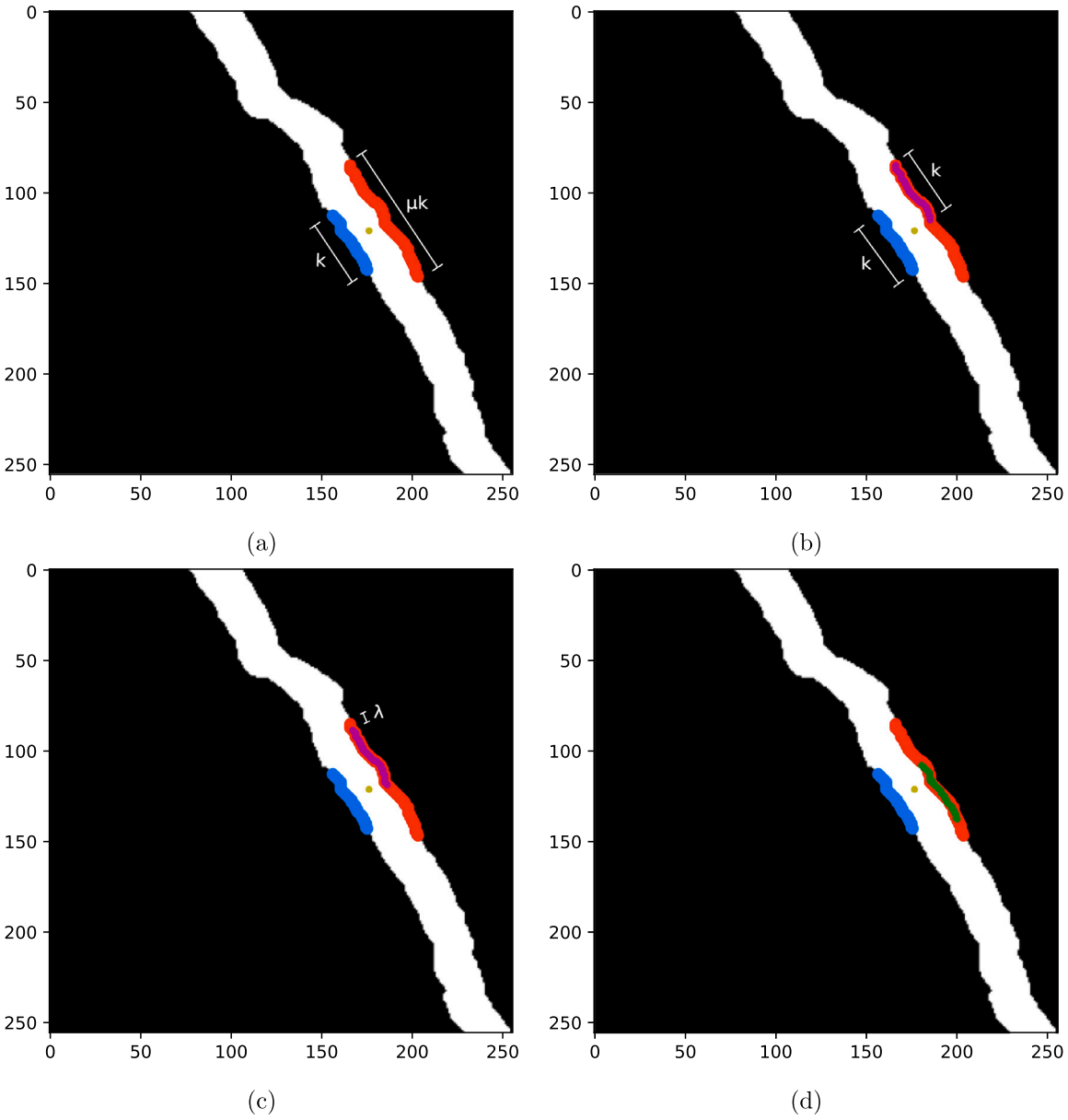


Fig. 6. Registration using a larger set of points of $\mathbf{E}_1$ with the hyper-parameters $k = 50$, $\mu = 2.0$, and $\lambda = 5$: (a) k-neighbors in $\mathbf{E}_0$ and μk-neighbors in $\mathbf{E}_1$ for a skeleton point (pixel), (b) first k group ($m = 1$) of points from $\mathbf{E}_1$ (magenta), (c) second k group ($m = 2$) of points from $\mathbf{E}_1$, and (d) k-neighbors points from $\mathbf{E}_0$ registered over a larger set of points of $\mathbf{E}_1$ (green) after applying the optimal transformation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4. Experiments

In this section, we present several examples to demonstrate the effectiveness of the developed technique. First, the algorithms are validated using two different types of synthetic examples (line-based and crack-based), in which pre-defined translations and rotations are applied to a given edge to simulate the binary mask of a crack. Following that, a Monte Carlo simulation evaluates the sensitivity of the method to the involved hyper-parameters and provides guidance on their selection. Later, by adding Gaussian noise to the edges, the robustness of the algorithm is tested on cases where there is material detachment from the crack contours. To further demonstrate the capabilities of the approach, the semantic segmentation of real cracks, as presented by Rezaie et al. [14], is used as input data. Next, we make a qualitative comparison between the DIC pipeline presented previously [16,17] and our full pipeline, which begins with semantic crack segmentation using the deep learning model published in Rezaie et al. [14]. Finally, we demonstrate the versatility of our method for various engineering applications by applying it to a building damaged by seismic loads.

4.1. Synthetic crack patterns with linear contour

The synthetic crack patterns shown in Fig. 10a–c are based on linear segments created with a pre-defined edge $\mathbf{E}_0$ and a Euclidean transformation (θ, t_x, and t_y). In this regard, we defined the vertices and created the line segments between them to create the synthetic edge $\mathbf{E}_0$. Next, a pre-defined transformation was applied to $\mathbf{E}_0$ to obtain $\mathbf{E}_1$ (observe blue and red edges in Fig. 10a–c). Columns 2 to 4 of Table 1 contain the parameters of the transformation used to create $\mathbf{E}_1$ from $\mathbf{E}_0$.

Applying the technique developed herein for estimating the crack kinematics of the synthetic data generated the results presented in columns 5 to 7 of Table 1. These columns show the prediction values

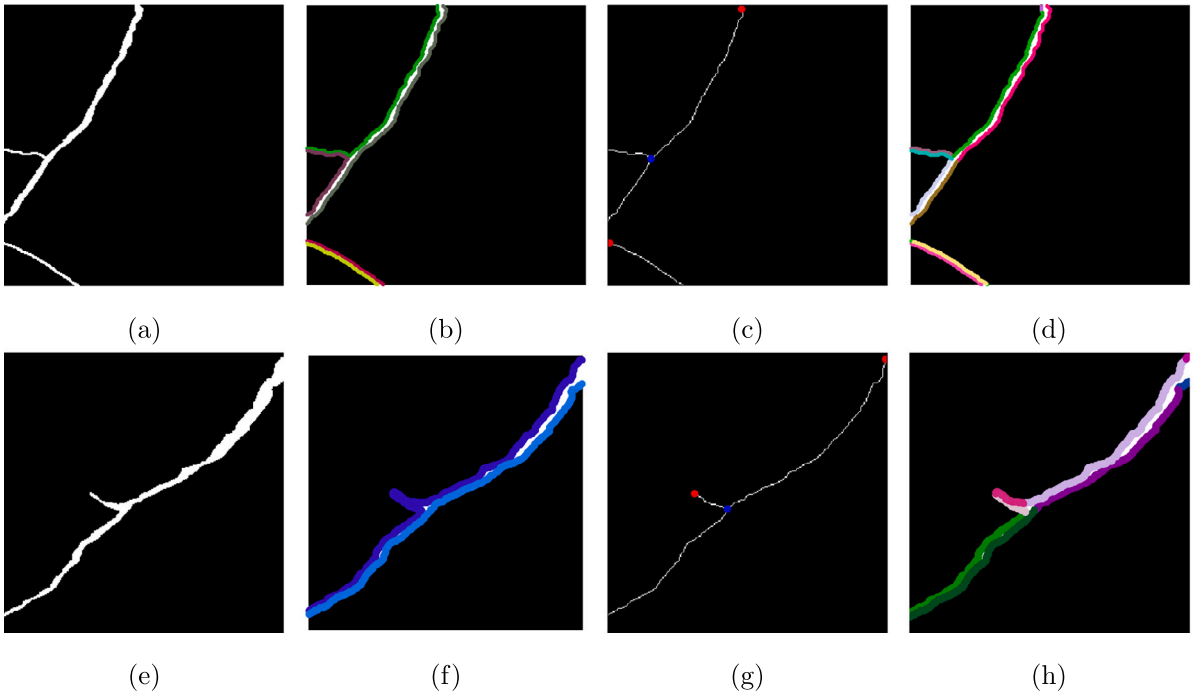


Fig. 7. Patterns with multiple branches and isolated cracks: (a) binary mask of detected crack (pattern 1), (b) contours detected using the marching squares method (pattern 1), (c) endpoints and junctions found in the skeleton (pattern 1), (d) group of edges found using detected contours together with endpoints and junctions (pattern 1), (e) binary mask of detected crack (pattern 2), (f) contours detected using the marching squares method (pattern 2), (g) endpoints and junctions found in the skeleton (pattern 2), and (h) group of edges found using detected contours together with endpoints and junctions (pattern 2).

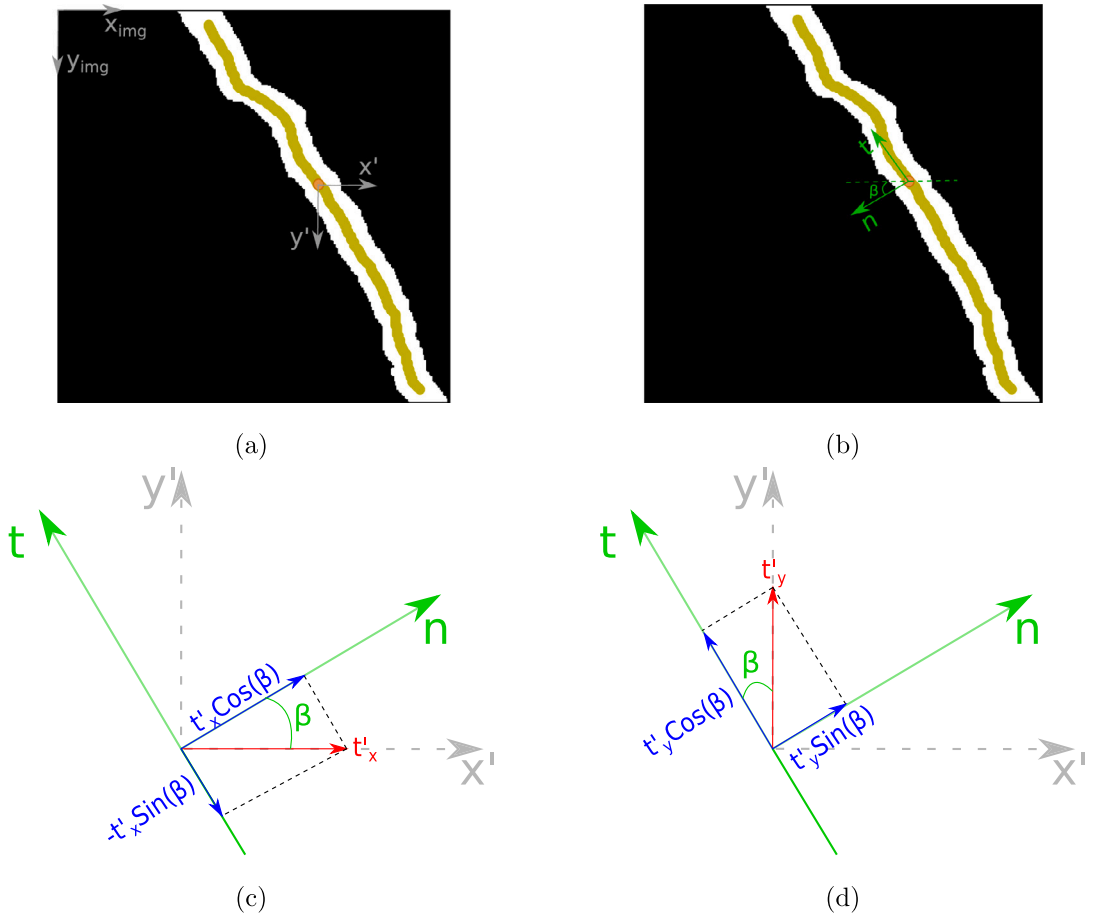


Fig. 8. Crack skeleton and directions of interest: (a) image and local coordinate systems, (b) local tangential t and normal n axes for a pixel/point of the skeleton with corresponding direction β, (c) local horizontal displacement component t'_x decomposed into tangential and normal components, and (d) local vertical displacement component t'_y decomposed into tangential and normal components.

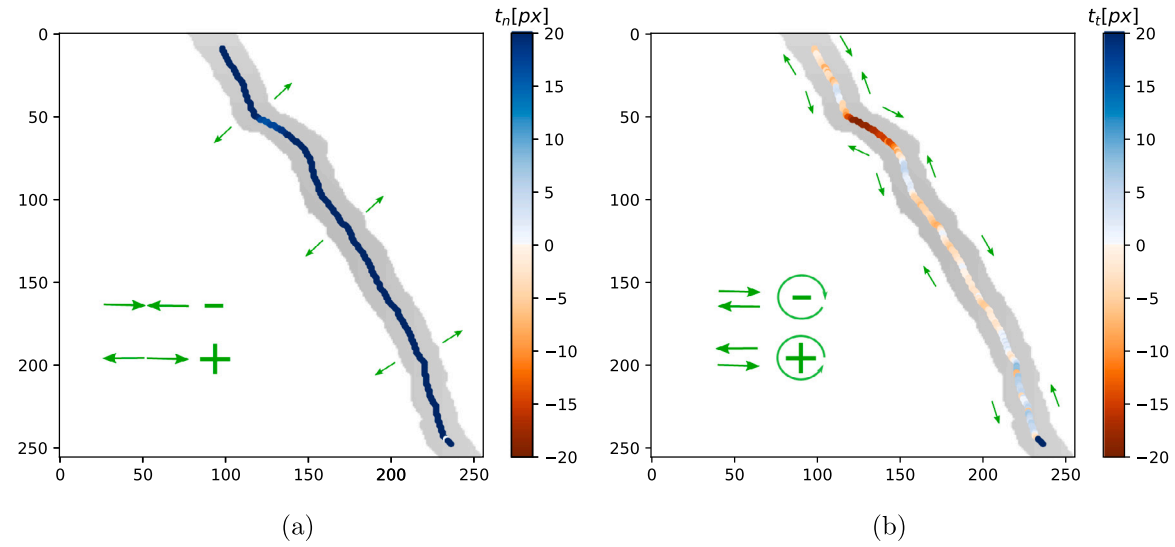


Fig. 9. Displacement maps for the tangential t and normal n directions: (a) normal displacement map t_n and (b) tangential displacement map t_t.

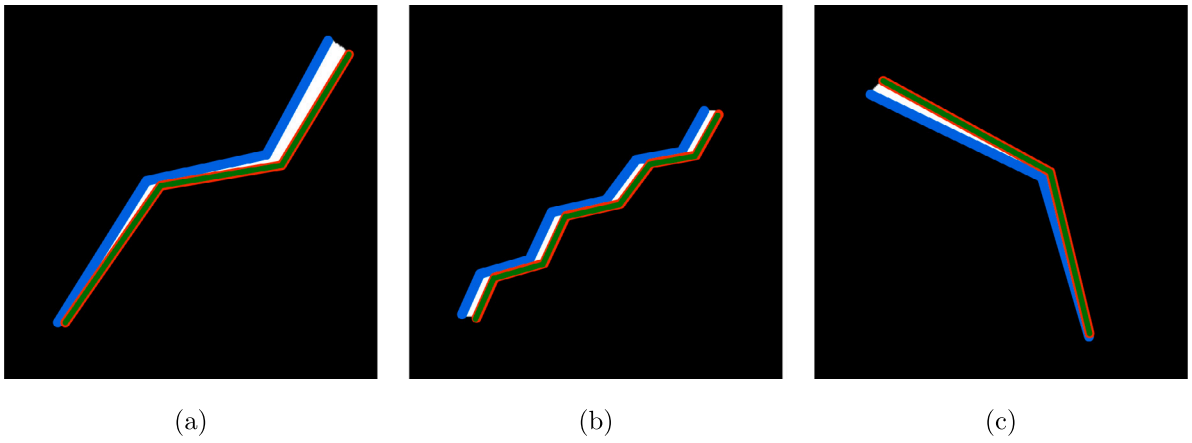


Fig. 10. Synthetic binary masks for a line-based crack. Edge $\mathbf{E}_0$ (blue), edge $\mathbf{E}_1$ (red) and registered edge $\mathbf{E}_0$ over edge $\mathbf{E}_1$ (green) for: (a) pattern 1, (b) pattern 2, and (c) pattern 3. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Pre-defined and predicted Euclidean transformation parameters for synthetic crack patterns with linear contour using full edges approach.

	Pre-define parameters			Predicted parameters			Absolute difference		
	θ [rad]	t_x [px]	t_y [px]	θ [rad]	t_x [px]	t_y [px]	$\Delta\theta$ [rad]	Δt_x [px]	Δt_y [px]
Pattern 1	$4.91e^{-2}$	10.00	5.00	$4.91e^{-2}$	9.84	4.77	0.00	0.16	0.23
Pattern 2	$0.00e^{-2}$	10.00	2.50	$-0.05e^{-2}$	9.30	3.08	$0.05e^{-2}$	0.70	0.58
Pattern 3	$4.91e^{-2}$	5.00	−5.00	$4.91e^{-2}$	5.35	−4.77	0.00	0.35	0.23

for rotation and displacements. In this example, the straight nature of the cracks renders the use of finite segments of $\mathbf{E}_0$ and $\mathbf{E}_1$ irrelevant, as fitting $\mathbf{E}_0$ on $\mathbf{E}_1$ cannot guarantee correct placement (e.g., a small linear segment of $\mathbf{E}_0$ will be identical to numerous small linear segments of the $\mathbf{E}_1$). Thus, we used the full edges of opposite sides as $\mathbf{E}_0$ and $\mathbf{E}_1$ for the optimization of this set of synthetic data. The green edges in Fig. 10a–c show the transformation of $\mathbf{E}_0$ overlying $\mathbf{E}_1$ for the optimal parameters obtained from the optimization process. These figures also demonstrate the proper functioning of the present technique because the two edges match all along the crack length. This is confirmed by comparing the obtained and target values through the small absolute difference values presented in columns 8 to 10 of Table 1.

4.2. Synthetic real-contour crack patterns

In this example, the edge $\mathbf{E}_0$ is transformed with pre-defined parameters θ, t_x and t_y to create the edge $\mathbf{E}_1$, and consequently the crack pattern. The difference in this case is that the edge $\mathbf{E}_0$ in each pattern is retrieved from the binary mask representing the semantic segmentation of a real crack (Fig. 11a–c). Fig. 11d–f show the binary patterns created after applying the transformation with the parameters presented in the columns 2 to 4 from the Table 2 to the edge $\mathbf{E}_0$, which is presented together with the edge $\mathbf{E}_1$.

Here, we initially ran our algorithms using the full edges, as for the line-based patterns, and then we used the general approach described in the methodology section to calculate the kinematics of each skeleton pixel using finite segments of $\mathbf{E}_0$ and $\mathbf{E}_1$. The qualitative and

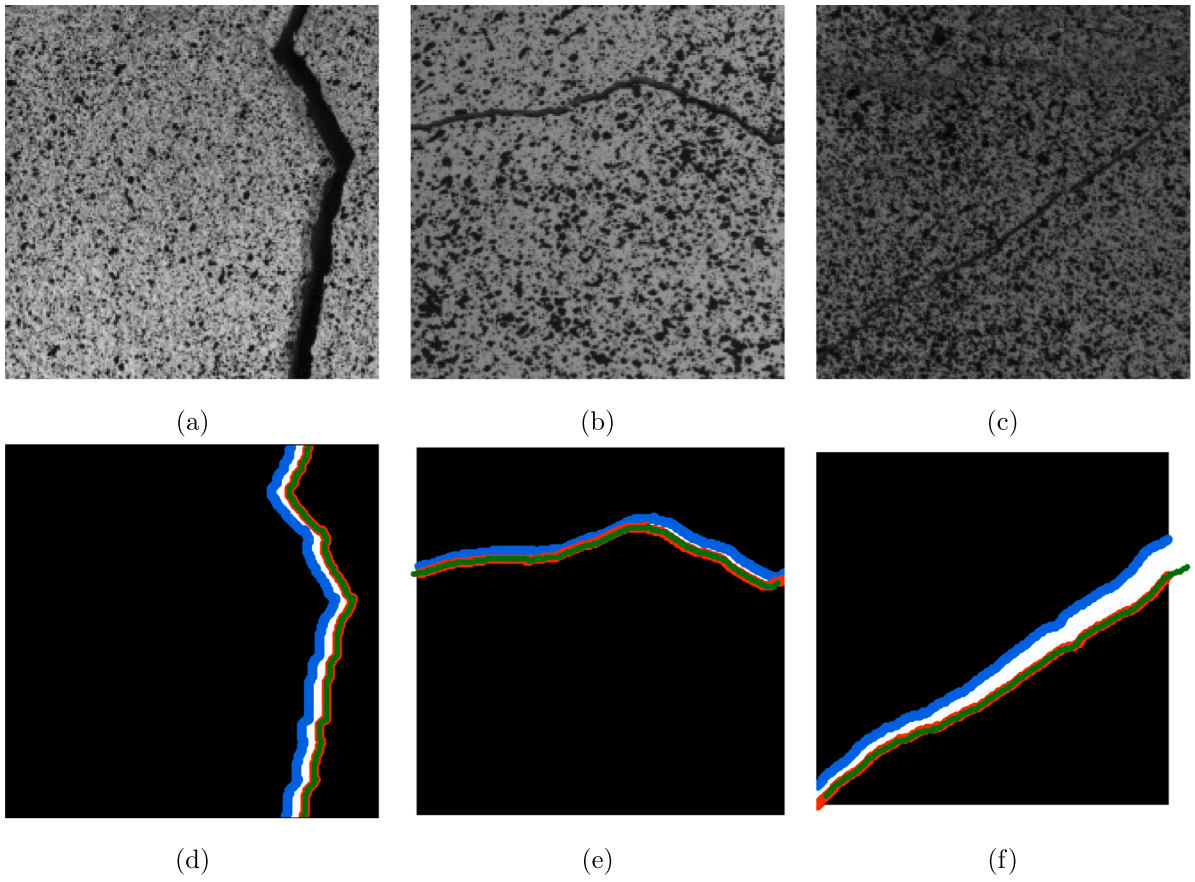


Fig. 11. Synthetic real-contour crack patterns. Crack pattern image: (a) pattern 1, (b) pattern 2, and (c) pattern 3. Edge $\mathbf{E}_0$ (blue), edge $\mathbf{E}_1$ (red) and registered edge $\mathbf{E}_0$ over edge $\mathbf{E}_1$ (green) for synthetic real-contour crack: (d) pattern 1, (e) pattern 2, and (f) pattern 3. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

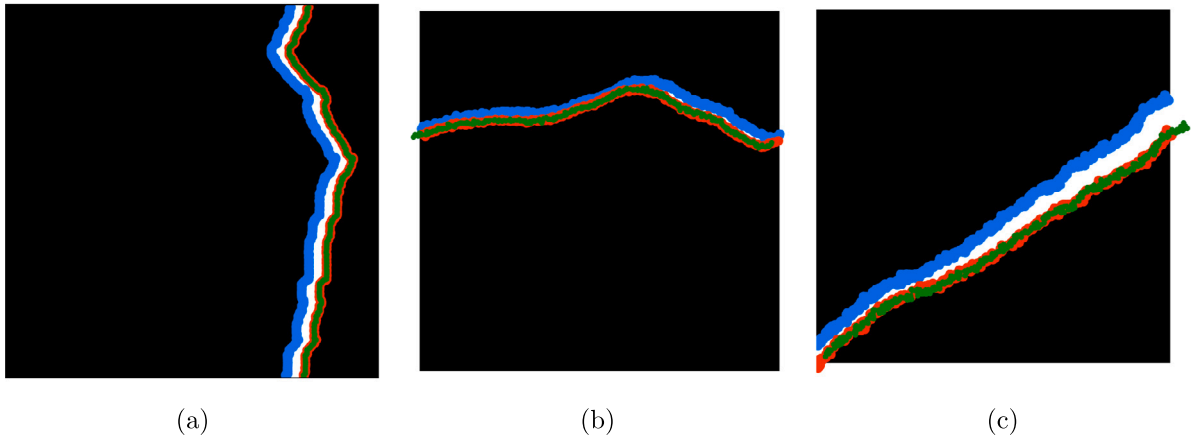


Fig. 12. Simulating loss of contour material by adding Gaussian noise to crack edges $\mathbf{E}_0$ (blue) and $\mathbf{E}_1$ (red). The registered noisy edge $\mathbf{E}_0$ (green) for: (a) pattern 1 with 10% noise, (b) pattern 2 with 50% noise, and (c) pattern 3 with 100% noise. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

quantitative results for full edges, presented in Fig. 11d–f and in the columns 5 to 7 from the Table 2, validate the performance of our method. Fig. 11d–f show the match between $\mathbf{E}_0$ and $\mathbf{E}_1$ when the optimal transformation is applied with the parameters presented in the columns 5 to 7 from the Table 2. Moreover, when that table of optimal transformation parameters is compared to the pre-defined (columns 2 to 4 of Table 2), the differences in parameter values are noticeably small, which confirms the qualitative observation.

Next, we performed the registration by considering finite edge segments in this synthetic data; the means of the absolute values of the three obtained DOFs are presented in the columns 8 to 10 from the Table 2. The results here slightly differ from those obtained using full edges, which quantitatively shows that the presented method is performing properly. This can be visualized properly in the Table 3 where the absolute difference for the predictions using full and finite segment edges is presented.

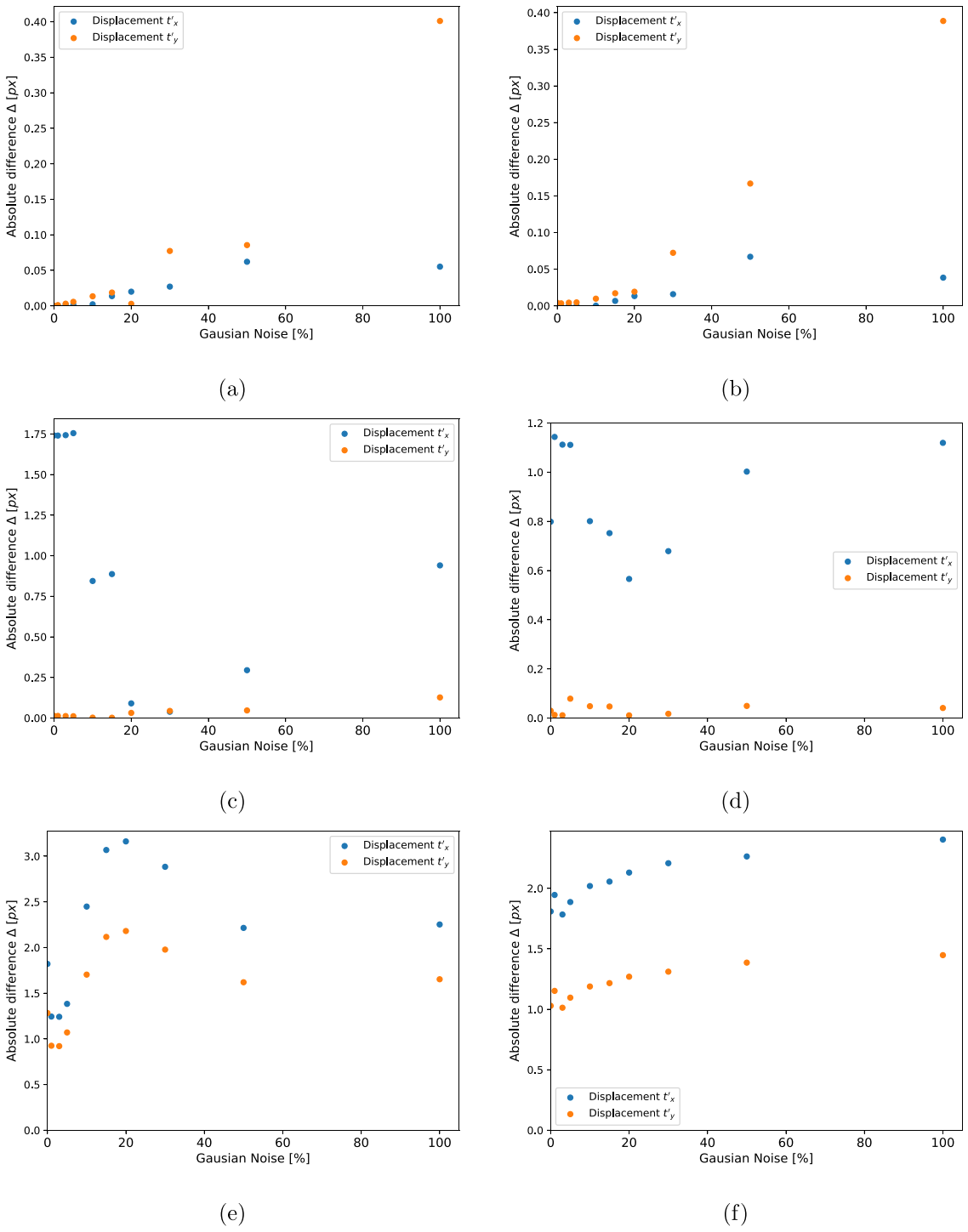


Fig. 13. Absolute difference in pixels between ground truth and predicted values for transformation parameters using two DOFs (t'_x and t'_y). Using full edge approach for: (a) pattern 1, (c) pattern 2, and (e) pattern 3. Using finite segment edge approach for: (b) pattern 1, (d) pattern 2, and (f) pattern 3.

Table 2
Pre-defined and predicted Euclidean transformation parameters for examples with synthetic real-contour crack patterns using full and finite edge segment approach. Finite segment edge approach uses $\eta = 1.5$, $\mu = 1.5$, $\lambda = 4$. .

	Pre-defined parameters			Predicted parameters full edges			Predicted parameters finite segment edges		
	θ [rad]	t_x [px]	t_y [px]	θ [rad]	t_x [px]	t_y [px]	θ [rad]	t_x [px]	t_y [px]
Pattern 1	0.00	10.00	0.00	$-7.54e^{-12}$	10.00	$2.68e^{-7}$	$5.17e^{-5}$	10.00	$4.08e^{-3}$
Pattern 2	0.00	−5.00	5.00	$5.07e^{-3}$	−3.26	5.01	$5.35e^{-3}$	4.20	5.03
Pattern 3	$4.91e^{-2}$	10.00	10.00	$4.85e^{-2}$	8.18	11.28	$5.13e^{-2}$	8.19	11.03

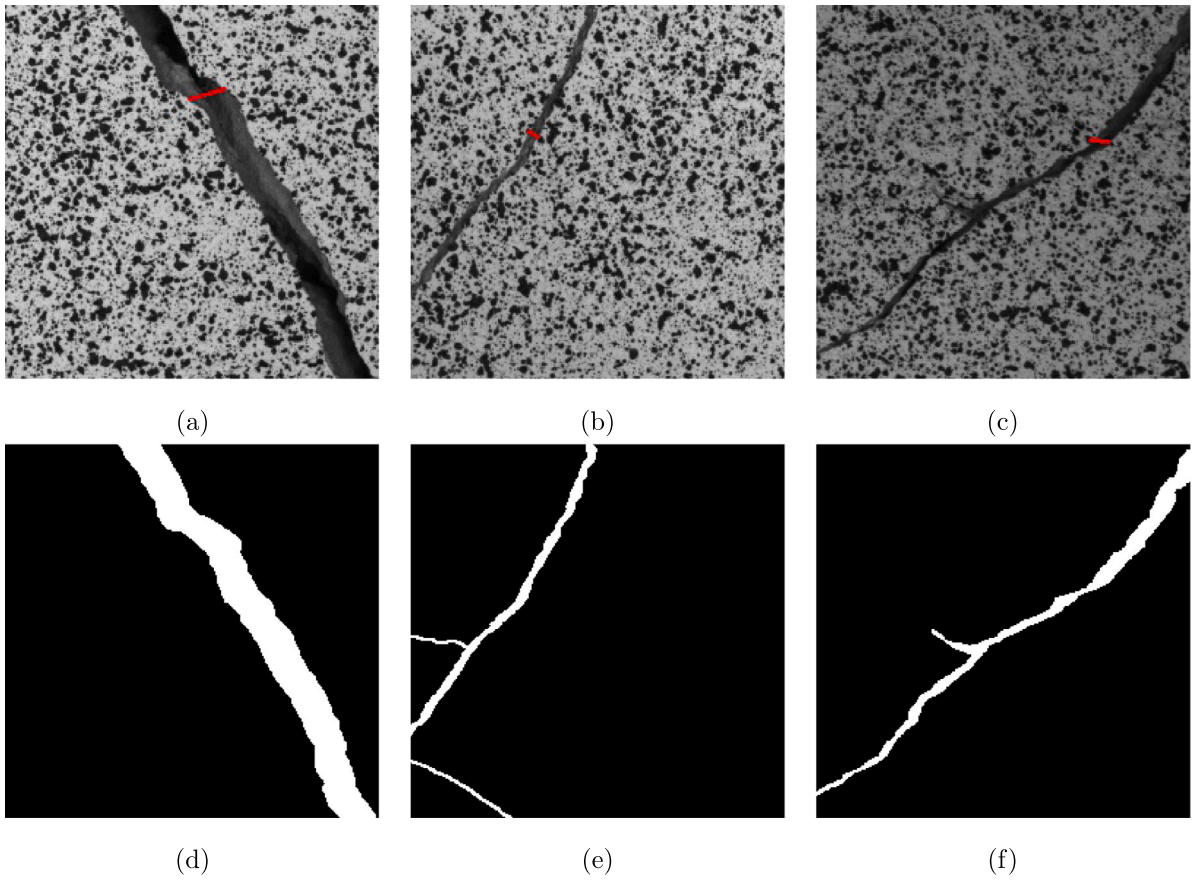


Fig. 14. Real crack patterns. Crack pattern image: (a) pattern 1, (b) pattern 2, and (c) pattern 3. Binary mask of real crack: (d) pattern 1, (e) pattern 2, and (f) pattern 3.

Table 3
Absolute difference for predictions of Euclidean transformation parameters for synthetic real-contour crack patterns using full and finite segment edges approach.

	Full edges			Finite segment edges		
	$\Delta\theta$ [rad]	Δt_x [px]	Δt_y [px]	$\Delta\theta$ [rad]	Δt_x [px]	Δt_y [px]
Pattern 1	$-7.54e^{-12}$	0.00	$2.68e^{-7}$	$5.17e^{-5}$	0.00	$4.08e^{-3}$
Pattern 2	$5.07e^{-3}$	1.74	0.01	$5.35e^{-3}$	0.80	0.03
Pattern 3	$0.06e^{-2}$	1.82	1.28	$0.22e^{-2}$	1.81	1.03

4.3. Robustness against noise

As the crack is exposed to weathering, cyclic loading or abrasion within Model II cracks, some material loss may occur along the two crack edges. In our algorithm, this will result in point sets of the segmented edges $\mathbf{E}_0$ and $\mathbf{E}_1$ that are not 100% similar. To simulate this and demonstrate the robustness of our method in such situations, we took the three synthetic crack patterns with real crack contours presented in Section 4.2 and added different levels of random Gaussian noise ($\mathcal{N}(\mu_{noise}, \sigma^2_{noise})$) to the coordinates of both edges: $\mathbf{x}_{noise} = \mathbf{x} + \epsilon\mathbf{r}(\sigma^2_{noise}, \mu_{noise})$ with $\mathbf{r} \sim \mathcal{N}(\mu_{noise} = 0, \sigma^2_{noise} = 1)$ and ϵ the fraction of noise added. Fig. 12 shows some of the noisy edges overlaid over the original binary crack pattern and the result after registration of edge $\mathbf{E}_0$ over edge $\mathbf{E}_1$ which qualitatively validates the robustness against noise.

We ran these experiments by adding random Gaussian noise of 1%, 3%, 5%, 10%, 15%, 20%, 30%, 50%, and 100% to the three crack patterns. The results in terms of the two DOFs (t'_x and t'_y) using our method with full and finite segment edge approaches are presented in Fig. 13. Comparing those quantitative results against the ground truth values showed that the noise addition did not substantially affect the outcome, especially when the finite segment edge approach was applied. It should be noted that, while the absolute difference values appear to be increased, such values are acceptable from a practical standpoint. As demonstrated in the following examples with real damaged conditions, those percentages are equivalent to fractions of mm, which are irrelevant because standard physical measurements with crack gauges can lead to similar errors. Furthermore, due to the simple required input and the lack of the need for sophisticated set-ups to capture image information, the algorithm accuracy is valuable in real-world applications, such as damage inspections.

4.4. Real crack patterns

In this section, the performance of the developed method is assessed using real crack patterns as presented in Fig. 14a–c retrieved from Rezaie et al. [30]. The binary masks representing the crack segmentation are presented in Fig. 14d–f. As stated in Section 3, though the patterns presented in this paper have speckled surfaces, we can obtain similar results in any material by relying solely on the binary representation of the crack.

The results obtained for the following hyper-parameters $\eta = 1.5$, $\mu = 1.5$, and $\lambda = 4$ are presented in Fig. 15. The detected crack contours were divided into edges, shown in Fig. 15a–c, and these were used for selecting edges $\mathbf{E}_0$ and $\mathbf{E}_1$ to compute the optimal transformations. The normal and tangential displacement maps are presented in Figs. 15d–f and 15g–i, respectively. Here, the predicted opening and shear-sliding movements agree with the pathology of the crack, even when considering multiple branches and isolated cracks.

Expectedly, some irregularities in the crack segments were observed near the image borders, as some points of edge $\mathbf{E}_0$ can be missing in corresponding edge $\mathbf{E}_1$, and vice-versa. Moreover, irregularities were also observed near crack junctions, which was also expected because the algorithms occasionally confuse the selections of edges $\mathbf{E}_0$ and

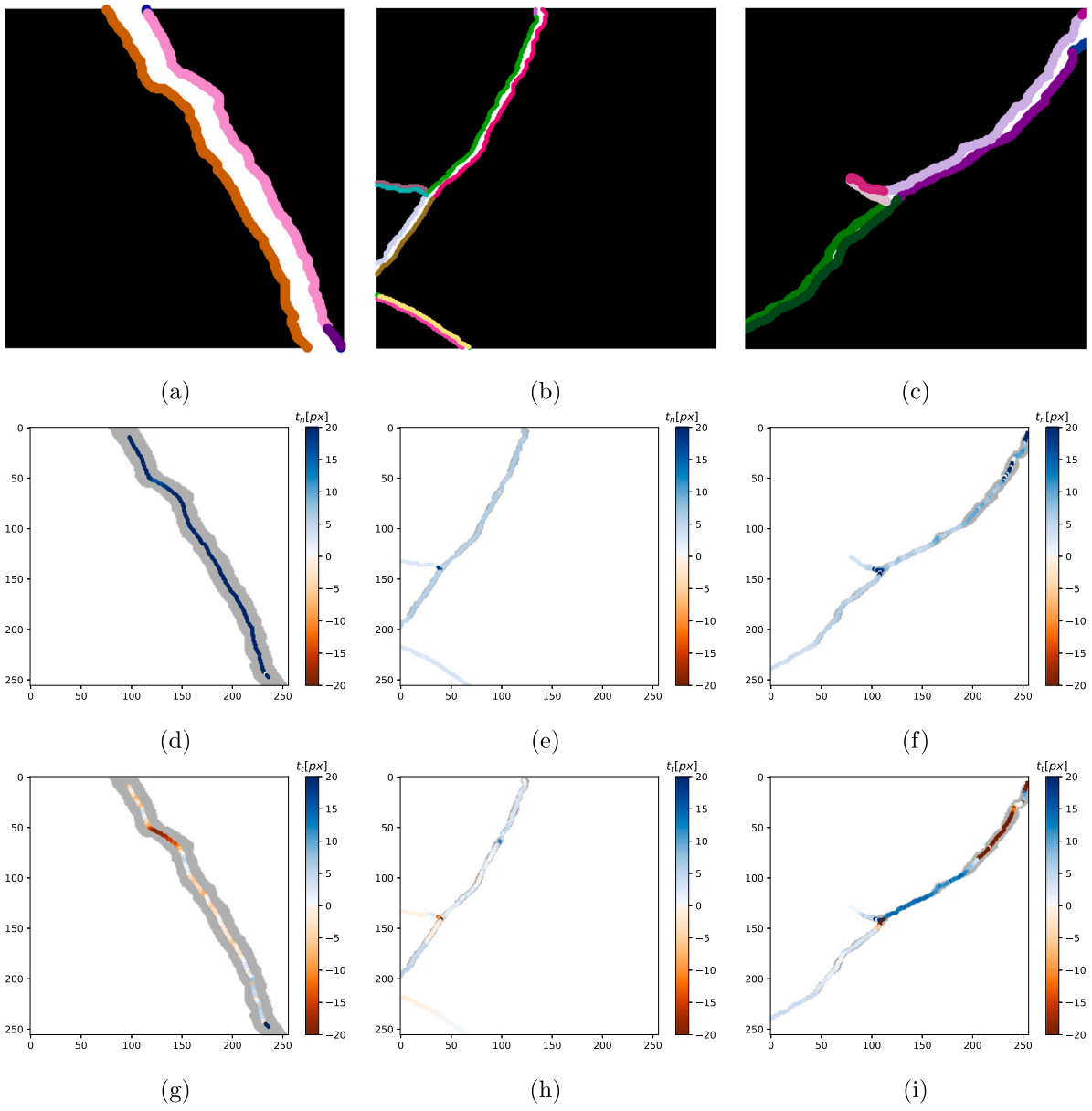


Fig. 15. Results for real crack patterns using $\eta = 1.5$, $\mu = 1.5$, and $\lambda = 4$. Group of edges found using detected contours, endpoints, and junctions for: (a) pattern 1, (b) pattern 2, and (c) pattern 3. Normal displacement maps (t_n) for: (d) pattern 1, (e) pattern 2, and (f) pattern 3. Tangential displacement map (t_t) for: (g) pattern 1, (h) pattern 2, and (i) pattern 3.

$\mathbf{E}_1$ among the available options. Furthermore, the present technique divides contours into multiple edges based on junctions and endpoints, which can influence the regularity of the results due to ambiguities in the distances from the skeleton to the edges. Regardless, the mechanically more relevant information is found between junctions where the algorithm performs well.

The means of the absolute value of normal and tangential movements ($[t_n, t_t]$) for each of the three crack patterns are: pattern 1: $[22.57, 7.74]$, pattern 2: $[4.99, 2.26]$ and pattern 3: $[9.29, 16.04]$. From the magnitude, it is clear that the crack represented by pattern 1 propagates in Mode I (crack opening). Conversely, the other two cracks are governed by propagation in combined Modes I and II.

To quantitatively validate the results, we manually select two opposite edge points (red line end-points in Fig. 4a–c) and measure the kinematics of the crack by decomposing displacements in the direction of the correspondent skeleton point. The measured values and results obtained by our algorithm in pixels and mm are shown in Table 4. Note that the image resolution for the used dataset is 0.43 mm/px, according to Rezaie et al. [30]. The small absolute differences in the table allow us to evaluate the proper performance, accuracy, and robustness of the methodology presented herein under real-world conditions.

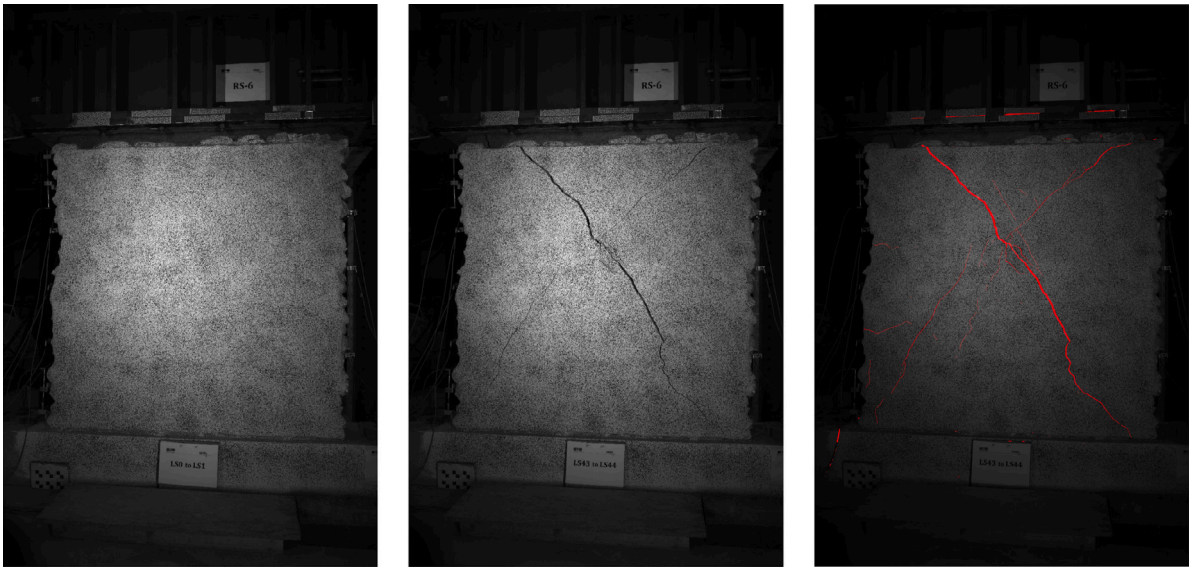
4.5. Comparison with DIC pipeline

We present qualitative and quantitative comparisons of two pipelines for predicting crack kinematics in damaged structures. The first is based on the principal strain maps obtained using DIC, while the second is based on binary semantic segmentation of the crack, i.e., the new method presented in this paper. Despite the fact that the required input data varies significantly between approaches, we extract comparable parts from crack patterns to benchmark our methodology quantitatively. The structure that was analyzed is a plastered stone masonry wall tested in an experimental campaign published in Rezaie et al. [15] and shown in Fig. 16.

The first pipeline, presented previously in two papers by Gehri et al. [16,17], post-processes the principal strain maps between two

Table 4
Comparison of measured normal and tangential displacements (t_n and t_t) with results using finite segment edges approach.

	Measured		Algorithm		Absolute difference	
	t_n [px] ([mm])	t_t [px] ([mm])	t_n [px] ([mm])	t_t [px] ([mm])	Δt_n [px] ([mm])	Δt_t [px] ([mm])
Pattern 1	20.30 (8.73)	15.05 (6.47)	19.59 (8.42)	15.34 (6.60)	0.71 (0.31)	0.29 (0.13)
Pattern 2	6.28 (2.70)	0.90 (0.39)	6.07 (2.61)	0.88 (0.38)	0.21 (0.09)	0.02 (0.01)
Pattern 3	9.79 (4.21)	8.44 (3.63)	9.59 (4.12)	8.32 (3.58)	0.20 (0.09)	0.12 (0.05)



(a) (b) (c)

Fig. 16. Damaged stone masonry wall used to predict crack kinematics: (a) wall without damage (time instant t_0), (b) wall with damage after applying the load (time instant t_1) and (c) segmented crack overlaid over damaged wall image using deep learning segmentation.

time instants (t_0 and t_1) that were developed using DIC, which detects cracks by establishing a strain threshold and then deduces their kinematics. Fig. 17 shows the results that we obtained using a two-sided threshold for the principal strain 1 mm/m $\leq T\epsilon_1 \leq$ 4 mm/m, a subset size $ss = 29$, and filter size $f = 15$ (refer to [16,17] for a deeper understanding of the parameters).

The second pipeline uses the approach put forward in this paper, beginning with the use of a deep learning model to semantically segment the cracks as a binary mask developed by Rezaie et al. [14] (Figs. 16c and 18a). This binary mask is then used by our algorithm to generate a crack kinematics map. In Fig. 18, we present the results obtained using this pipeline. Aside from analyzing the entire wall, patches can be extracted from the full prediction binary image and then analyzed by the algorithms for more detailed and comparable results. Figs. 18a and 19, show the patches of sizes 256 × 256 px and 512 × 512 px that we selected and Figs. 20 and 21 present their crack kinematic results using both methodology pipelines in terms of t_t/t_n and t_n respectively.

When comparing the results obtained with the two approaches, the first aspect to notice is the significant difference in the number of detected cracks (Figs. 17c and 18a). The DIC approach relies on the strain map produced by the most recent load increment and the thresholds applied to it; as a result, the number of detected cracks is strongly dependent on the threshold values applied [15]. Our approach, on the other hand, only requires the use of a segmentation technique of the structure's current damaged state for the required input. Despite the clear differences in the detected cracks, the values of the relation t_t/t_n are close between methodologies for most commonly detected cracks, as shown in Figs. 17d and 18b for the full wall, or in the Figs. 20 and 21 for extracted image patches with detected crack patterns that are similar. For a better quantitative assessment of our method, we select a skeleton crack point for each extracted patch, as shown in the Fig. 19a–c, and their kinematic displacements in mm are calculated and presented in the Table 5 (considering image resolution of 0.43 mm/px [14]). Both methodologies produced similar numerical results, according to the table. The distinctions are justified because the two methodologies require different input information, which is then post-processed according to different criteria during execution. For instance, to detect cracks using DIC, it is necessary to consider the previous loads steps and accumulate the cracks detected for each step. As our purpose is not to further develop that methodology [16,17], we restricted our example to the use of only two load steps.

Because improving the accuracy of the prediction using DIC-based algorithms might be laborious in its requirement for more complete data, our approach has an advantage because the algorithm only requires the use of a segmentation technique of the current damaged status of the structure for the required input. In addition, the results differ because the two methods compute the kinematics using different reference points from the two crack sides. While our algorithm selects points that are exactly over the segmented crack edges, the DIC method selects reference points that are measured outside of the crack contours and where displacement values are available [16,17]. As a result, when the kinematic is computed using the DIC method, the crack width may be overestimated, as shown in Table 5 where the width values are generally larger in comparison to our results. Furthermore, when comparing DIC-based methodology to the technique hereby developed, it should be noted that DIC requires significant preparations, such as applying speckle patterns to the specimen and calibrating the cameras. The current technique, on the other hand, is designed to work in the field with conventional cameras and in harsh environmental conditions. The work presented in this paper demonstrates that the current technique is adaptable and that the level of error is acceptable.

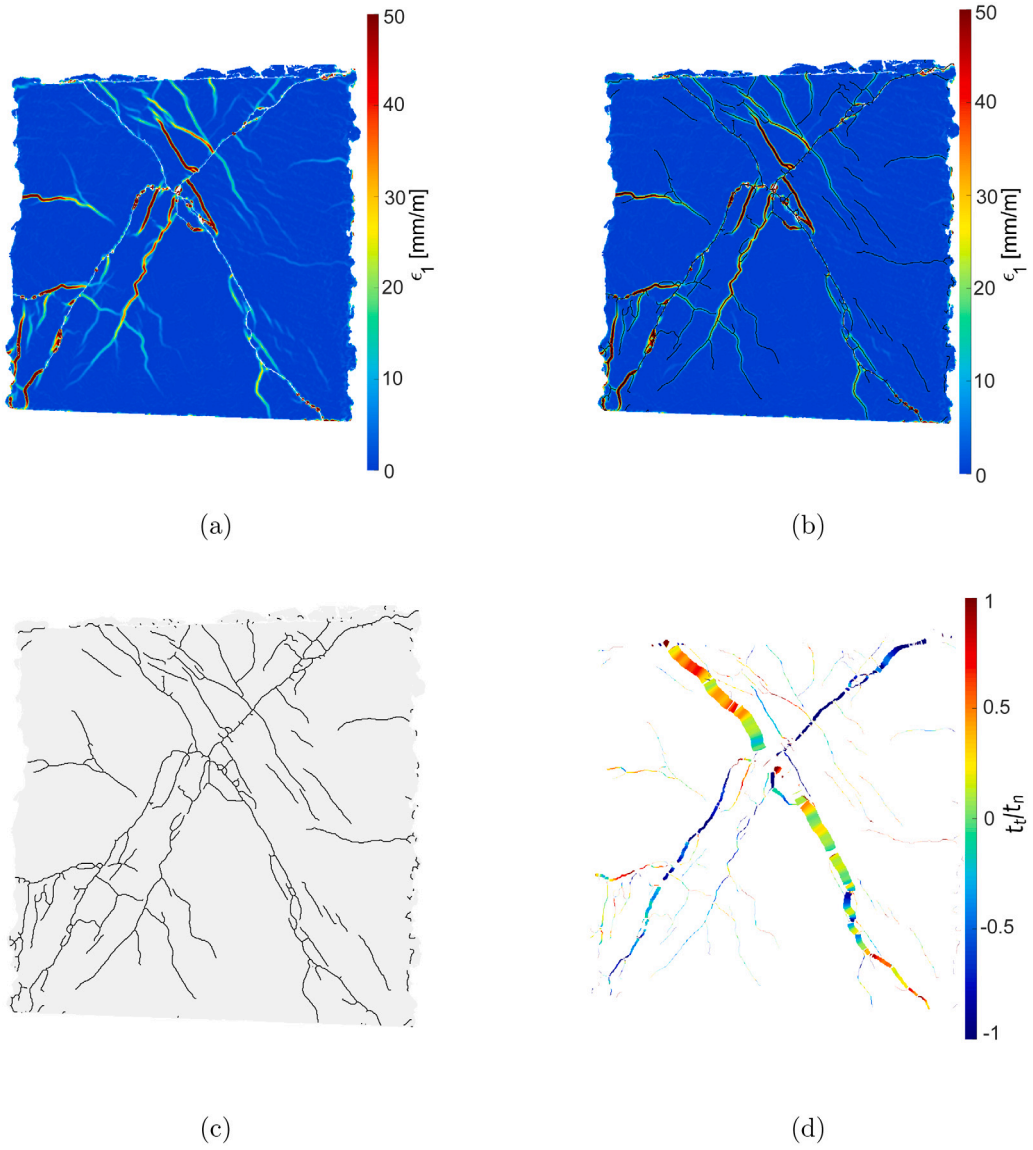


Fig. 17. Comparing pipelines for predicting the crack kinematics of a damaged stone masonry wall. DIC pipeline reported in previously published works [16,17]: (a) principal strain map ϵ_1, (b) detected crack using thresholds of the strain map, (c) detected crack skeleton, and (d) crack kinematics, relation t_t/t_n.

Table 5
Comparison of normal and tangential displacements (t_n and t_t) computed by our binary-mask-based and DIC-based approaches.

	DIC-based [16,17]		Ours		Absolute difference	
	t_n [mm]	t_t [mm]	t_n [mm]	t_t [mm]	Δt_n [mm]	Δt_t [mm]
Pattern 1	1.07	1.02	1.08	1.88	0.01	0.82
Pattern 2	10.65	4.74	9.13	4.97	1.52	0.23
Pattern 3	11.40	0.87	9.96	0.82	1.44	0.05

4.6. Damaged beam

As another validation example, we use our methodology on a cracked beam with dimensions 135 mm×15 mm×8 mm presented in the Fig. 22. The two visible cracks were caused by vertical loads applied in two load steps at midspan, moving the location of the supports between the two load steps. The results obtained using the pipeline proposed in this paper are displayed in Fig. 23.

To quantitatively assess our method with this example, we used a metric gauge to measure the crack width (normal displacement) at various points in both crack patterns, as shown in the Fig. 24a,c and compared these values to our results. We limited our validation to the Mode I crack opening because manual measurement of tangential crack displacements is not feasible. Fig. 24b,d show the patches of 256 × 256 px extracted from the original image crack patterns centered on the selected measured points (red points) and from which the normal displacement t_n (crack width) is extracted from the results (Fig. 22b,c depict the location of patches in the crack patterns).

Table 6 shows the measured width values as well as the extracted results from our method for the same locations. To compute the results in mm of our methodology, we previously determined the image resolution for the image crack patterns to be 0.0419 mm/px for pattern 1 and 0.0425 mm/px for pattern 2. When the reference measurement of the gauge for the different crack points is compared to the results of our method, we can observe appropriate performance, accuracy, and robustness. Among the error factors to consider are the subjectivity of measuring crack properties with a gauge and crack edge segmentation, the latter of which can be improved if better methodologies are used. Despite this, the results are satisfactory and validate our methodology.

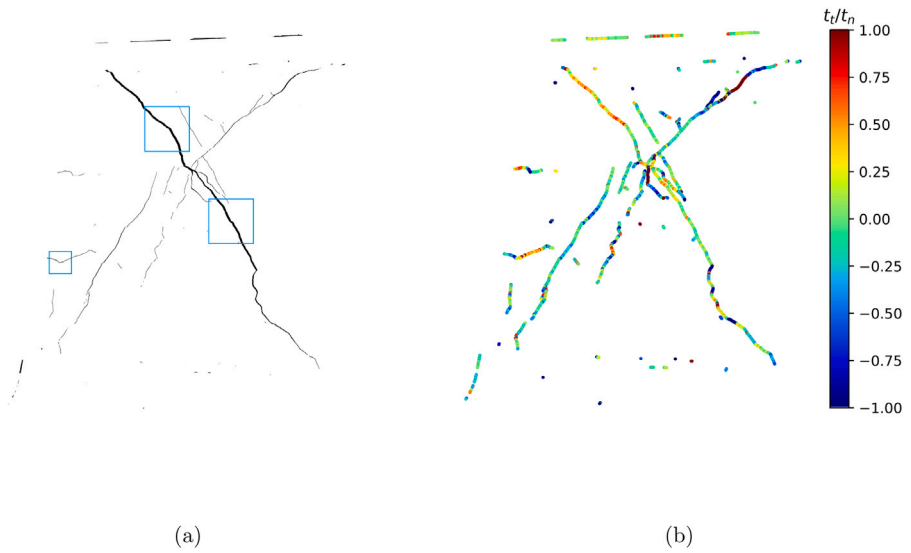


Fig. 18. Comparing pipelines for predicting the crack kinematics of a damaged stone masonry wall. Binary-mask-based pipeline—crack kinematics using our approach with $\eta = 1.0$, $\mu = 1.5$, $\lambda = 4$: (a) Binary mask output from deep learning prediction and location of extracted patches sized 256 × 256 px, and (b) crack kinematics—relation t_t/t_n.

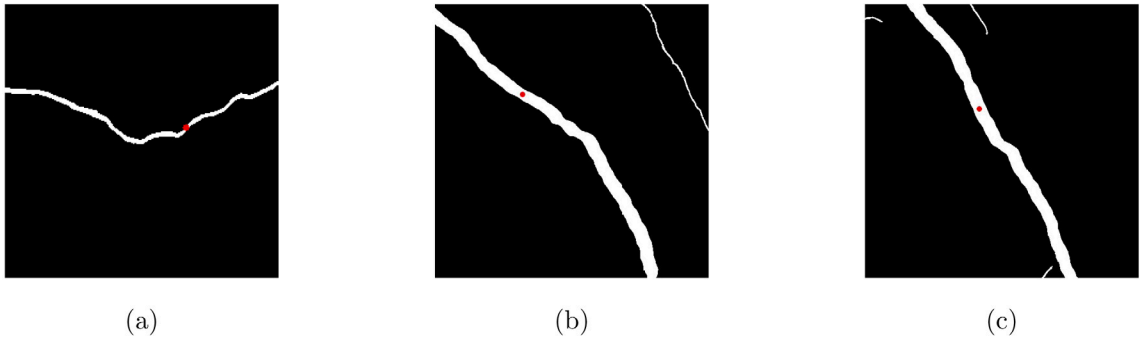


Fig. 19. Three extracted patches of 256 × 256 px and 512 × 512 px analyzed using both methods. Binary mask patches with selected skeleton points (red) where kinematic was exported to compare methodologies for: (a) 256 × 256 px pattern 1, (b) 512 × 512 px pattern 2, and (c) 512 × 512 px pattern 3. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6
Beam experiment: Comparison of normal displacements (t_n—crack width) computed by our binarymask-based approach to physically measured displacements with metric gauge for the points shown in Fig. 24b,d.

	Pattern 1			Pattern 2		
	Gauge t_n [mm]	Ours t_n [mm]	Δt_n [mm]	Gauge t_n [mm]	Ours t_n [mm]	Δt_n [mm]
Point 1	0.40	0.41	0.01	1.80	1.78	0.02
Point 2	1.40	1.23	0.17	4.00	3.67	0.33
Point 3	1.60	1.48	0.12	0.60	1.21	0.61
Point 4	2.00	1.50	0.50	1.80	2.28	0.48
Point 5	2.00	1.43	0.57	4.00	3.91	0.09

4.7. Post-earthquake damage assessment

To conclude the experimental work, we used another example of a building that was damaged during the 2020 Petrinjia earthquake [31, 32]. To identify the cracks, we used a deep learning model that builds on the work presented by Rezaie et al. [14], as shown in Fig. 25. We ran our code on the binary mask that represents the cracks over the building (Fig. 26a). The kinematics results for the crack pattern are presented in Fig. 26b–d.

We extracted and analyzed a 512 × 512 px patch that corresponds to the pier shown in Figs. 25b and 26a. In Fig. 27, we can see a region with a predominant crack propagating in Mode II with shear displacements from the crack kinematic results. This makes mechanical sense because shear loads generated during earthquake acceleration are expected to cause the diagonal cracks observed in that building pier.

The results obtained from the use of complex patterns and real-world structure cases (DIC wall and damaged building) demonstrate the applicability and adaptability of our methodology to real-world situations. As a result, we believe that the simplicity and robustness of our approach make it an useful contribution for damage assessment when crack characterizing is required.

5. Conclusions

In this paper, we proposed a novel methodology for determining the propagation of cracks in structural elements caused by unknown forces (crack kinematics). This procedure takes as inputs binary images obtained by semantically segmenting a crack from an image of a crack pattern. To the best of our knowledge, this is the first method to solve

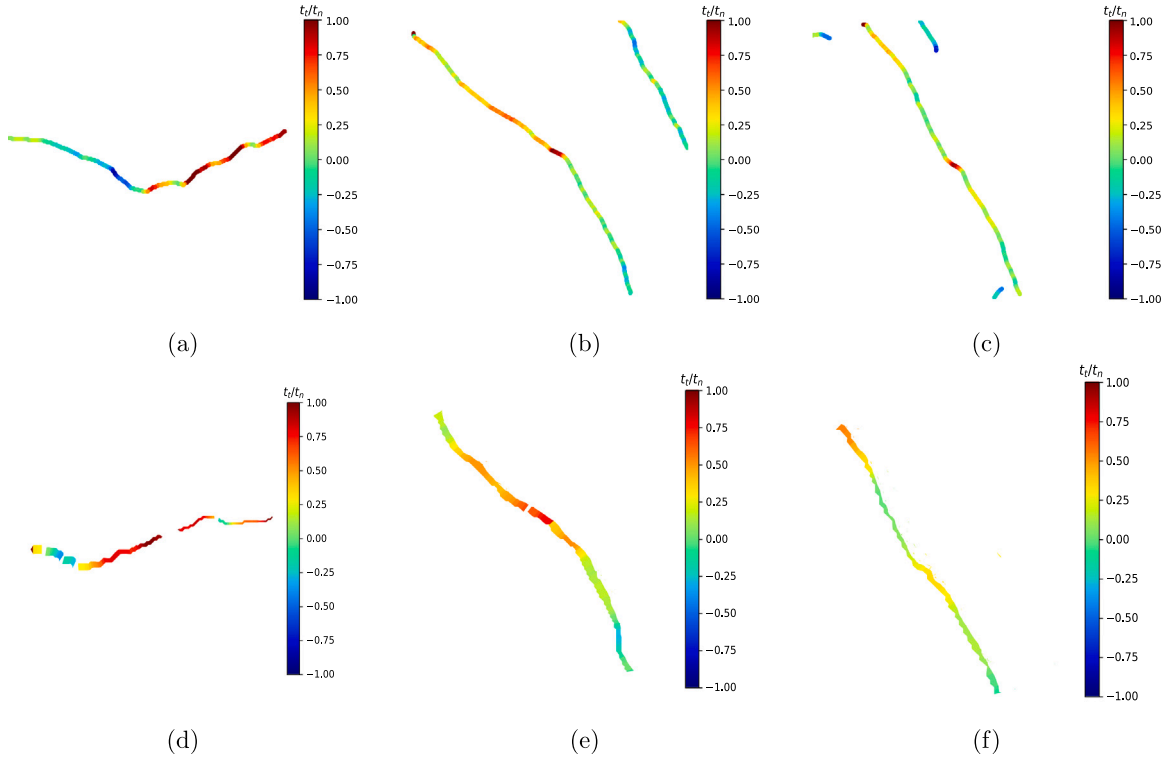


Fig. 20. Comparing pipelines for predicting the crack kinematics of a damaged stone masonry wall. Binary-mask-based and DIC pipelines—crack kinematic for three extracted patches of 256 × 256 px and 512 × 512 px analyzed using both methods: (a–c) crack kinematic—relation t_t/t_n using our binary-mask-based pipeline for patterns 1–3, and (d–f) crack kinematic—relation t_t/t_n using DIC pipeline patterns 1–3 [16,17].

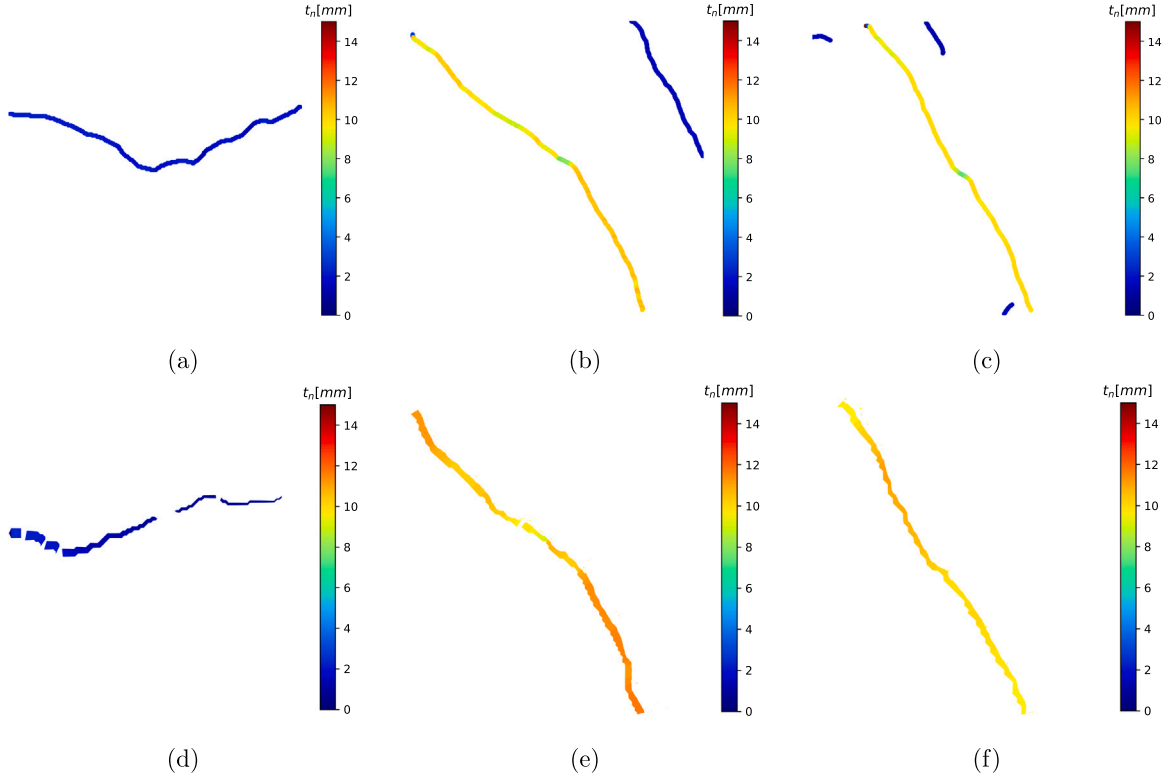


Fig. 21. Comparing pipelines for predicting the crack kinematics of a damaged stone masonry wall. Binary-mask-based and DIC pipelines—normal displacement for three extracted patches of 256 × 256 px and 512 × 512 px analyzed using both methods: (a–c) normal displacement t_n [mm] using our binary-mask-based pipeline for patterns 1–3, and (d–f) normal displacement t_n [mm] using DIC pipeline for patterns 1–3 [16,17].

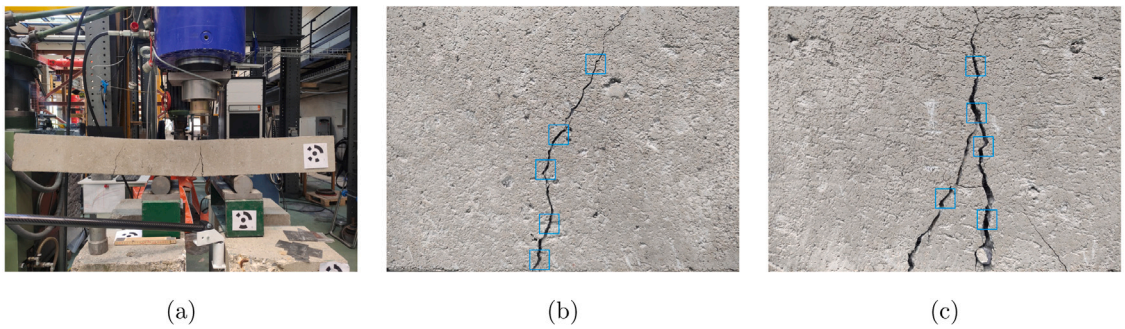


Fig. 22. Crack kinematics for damaged beam—beam crack patterns: (a) beam set up, beam crack pattern with position of extracted patches of 256 × 256 px to measure the crack for (b) Pattern 1 and (c) Pattern 2.

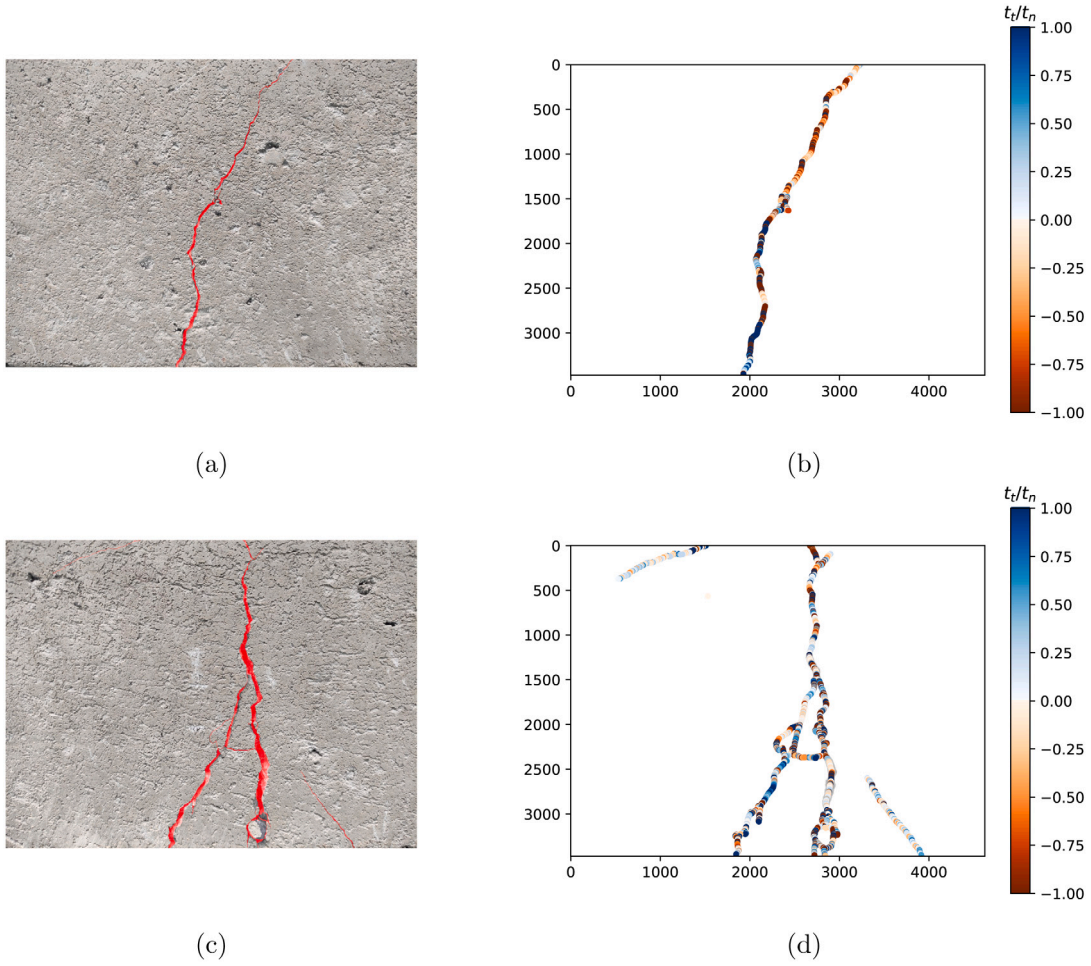


Fig. 23. Crack kinematics for damaged beam—Binary-mask-based pipeline segmenting crack: (a) segmented crack overlaid over beam crack pattern 1, (b) crack kinematics—relation t_t/t_n for crack pattern 1, (c) segmented crack overlaid over beam crack pattern 2 and (d) crack kinematics—relation t_t/t_n for crack pattern 1.

the crack kinematics determination problem based on single images, significantly advancing the state-of-the-art in the field.

The first step of our developed technique consists of detecting the contours of a crack pattern as a 2D set of points that are divided into opposite edges along the crack length. Next, the edges are registered using Euclidean transformations encoding both normal and tangential displacements, which are used to determine whether the cracks propagated in Mode I or Mode II. Additional normal features are included in our loss function to improve the results of the registration problem. We validated our method with synthetic data based on lines and real crack contours, where pre-defined rotation and translation were used to generate crack patterns, obtaining an absolute error of less than a pixel. Monte Carlo simulations were used to demonstrate the stability of our method as well as to guide the selection of hyper-parameters. Then, we put our methodology to the test with challenging crack patterns that contained multiple cracks and branches, including two real structural applications (damaged wall during laboratory testing, damaged beam and damaged building due to an earthquake event). Extra validation of the performance, accuracy and robustness of our methodology is made comparing results with a DIC-based method and manual measurements with metric gauge. The results of both validation and test experiments showed that the developed approach has the potential to deduce the crack kinematics simply from spatial features (2D edges coordinates) extracted from a binary input image.

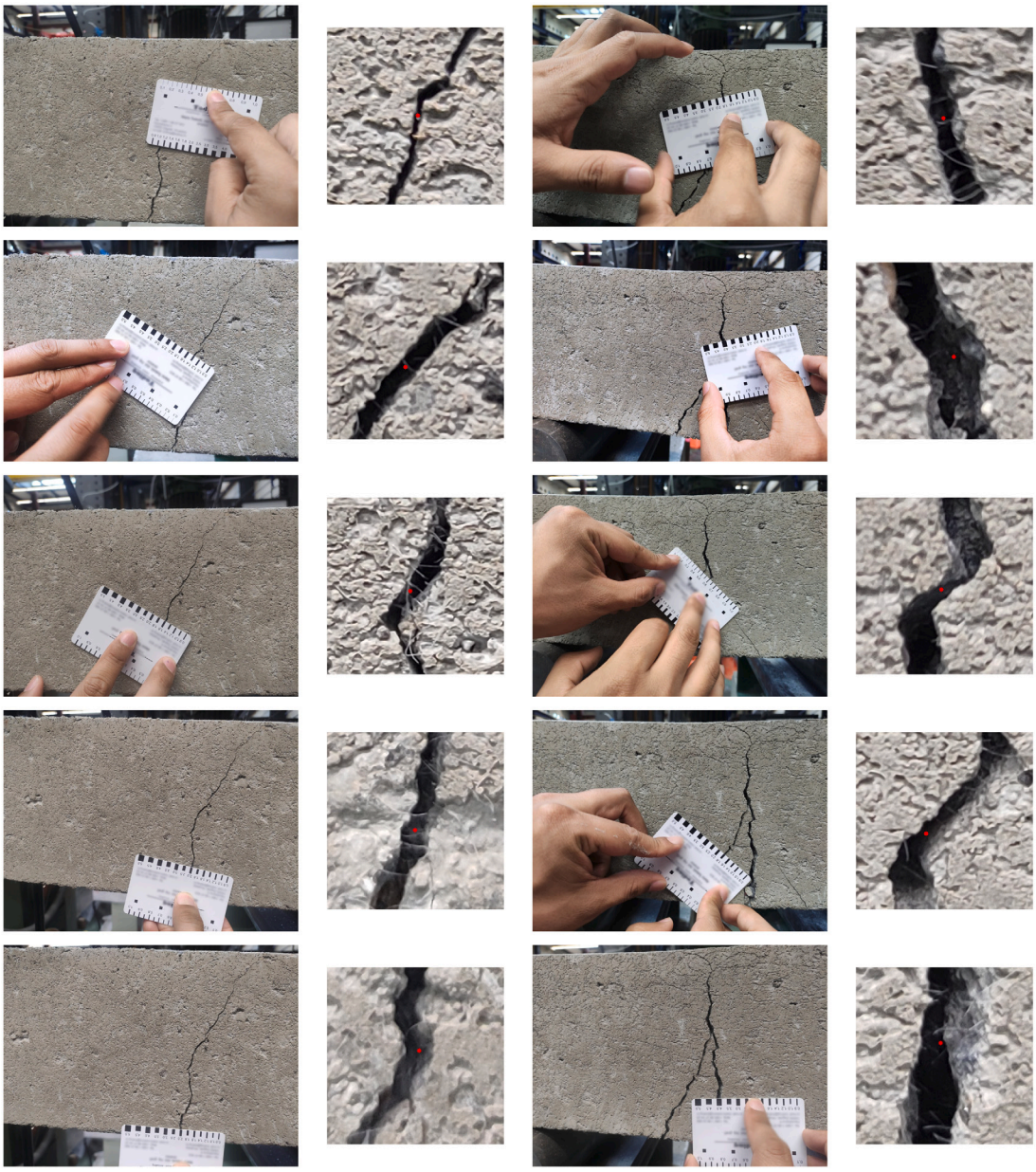


Fig. 24. Crack kinematics for damaged beam—comparison with measured crack widths: (a) measured crack widths for Pattern 1, (b) extracted patches of 256 × 256 px around measured crack points for Pattern 1, (c) measured crack widths for Pattern 2 and (d) extracted patches of 256 × 256 px around measured crack points for Pattern 2.

Although we demonstrated the effectiveness of our approach, there are some aspects that require further investigation. First, we identified the need to investigate alternative approaches to automating the calibration of the hyper-parameters, namely k (or η), mu, and $lambda$, in the selection of the transformed edges. This calibration should assign different values based on the features of the branches (e.g., size, curvature) that make up the crack pattern. Additionally, further improvements to the algorithm can be made using features other than 2D coordinates, such us features generated from gradient-based descriptors. Furthermore, we believe that combining the binary image with the original crack pattern image will be beneficial in future developments. Finally, including information about depth of the surface, either using depth instrumentation (e.g., RGBD cameras, lidar) or stereo configuration (two cameras), would allow the inclusion of Mode III crack propagation to our methodology.

We believe that our work will have a significant impact in the field of image-based damage assessment because of its simplicity, robustness, and adaptability and that it can extract valuable information, such as crack kinematics, from simple data. Moreover, it will allow for more research in a variety of applications, such as the prediction of mechanical properties of damaged structural elements or studies to understand why damage occurred.

CRediT authorship contribution statement

B.G. Pantoja-Rosero: Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **K.R.M. dos Santos:** Conceptualization, Validation, Writing – original draft, Writing – review & editing. **R. Achanta:** Conceptualization, Methodology, Resources, Supervision. **A.**

Rezaie: Conceptualization, Methodology, Resources. **K. Beyer:** Conceptualization, Methodology, Investigation, Resources, Writing – original draft, Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This project is partially funded by the Swiss Data Science Center under grant C18-04 ("Towards an automated post-earthquake damage assessment").

Appendix A. Pseudo-code of the technique

The pseudo-code is summarized in Algorithm 1. The input parameters are k, μ, and λ, and the output contains the crack kinematics in

Fig. 25. Crack kinematics for post-earthquake damage assessment—Binary-mask-based pipeline segmenting crack with deep learning model: (a) damaged building image due seismic activity and (b) segmented crack overlaid over damaged building image with 512 × 512 px patch location.

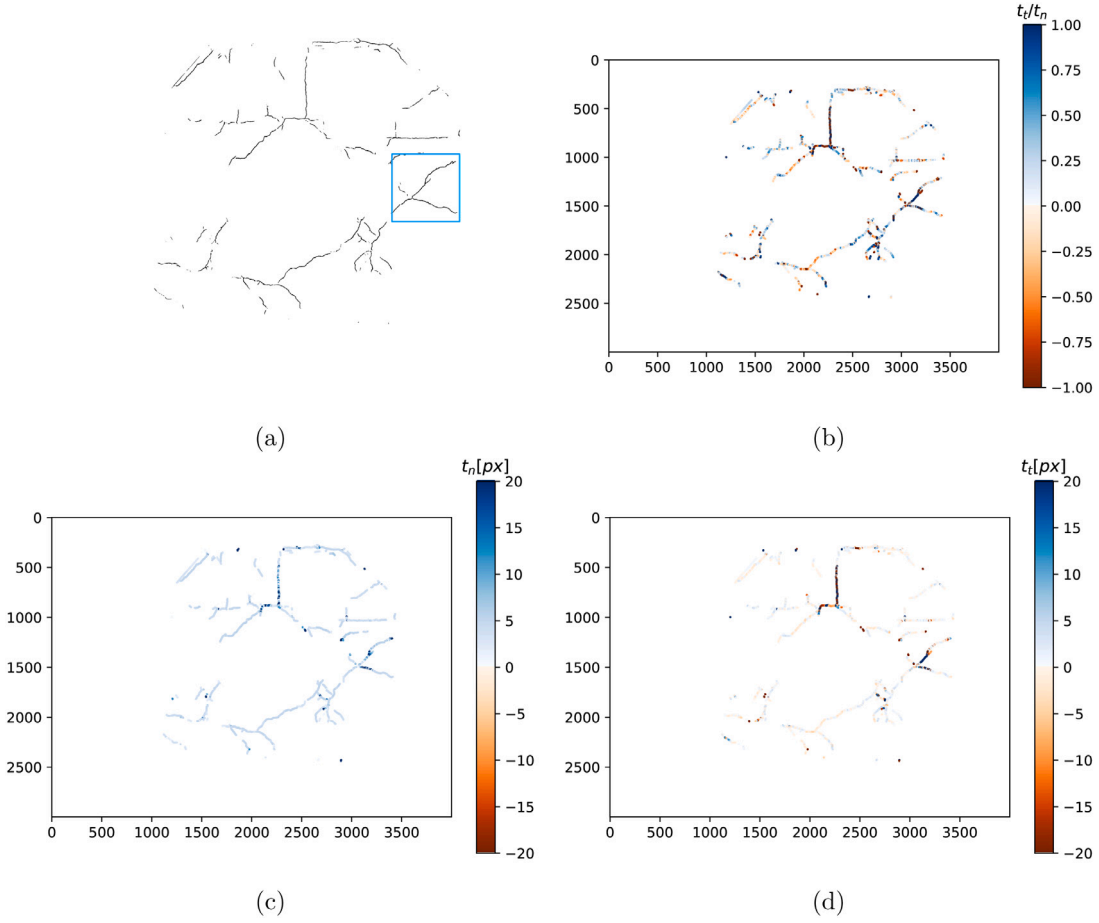


Fig. 26. Crack kinematics for post-earthquake damage assessment—crack kinematics using our approach with $\eta = 1.0$, $\mu = 1.5$, $\lambda = 4$: (a) binary mask output from deep learning prediction with 512 × 512 px patch location, (b) crack kinematics—relation t_t/t_n, (c) normal displacement map (t_n), and (d) tangential displacement map (t_t).

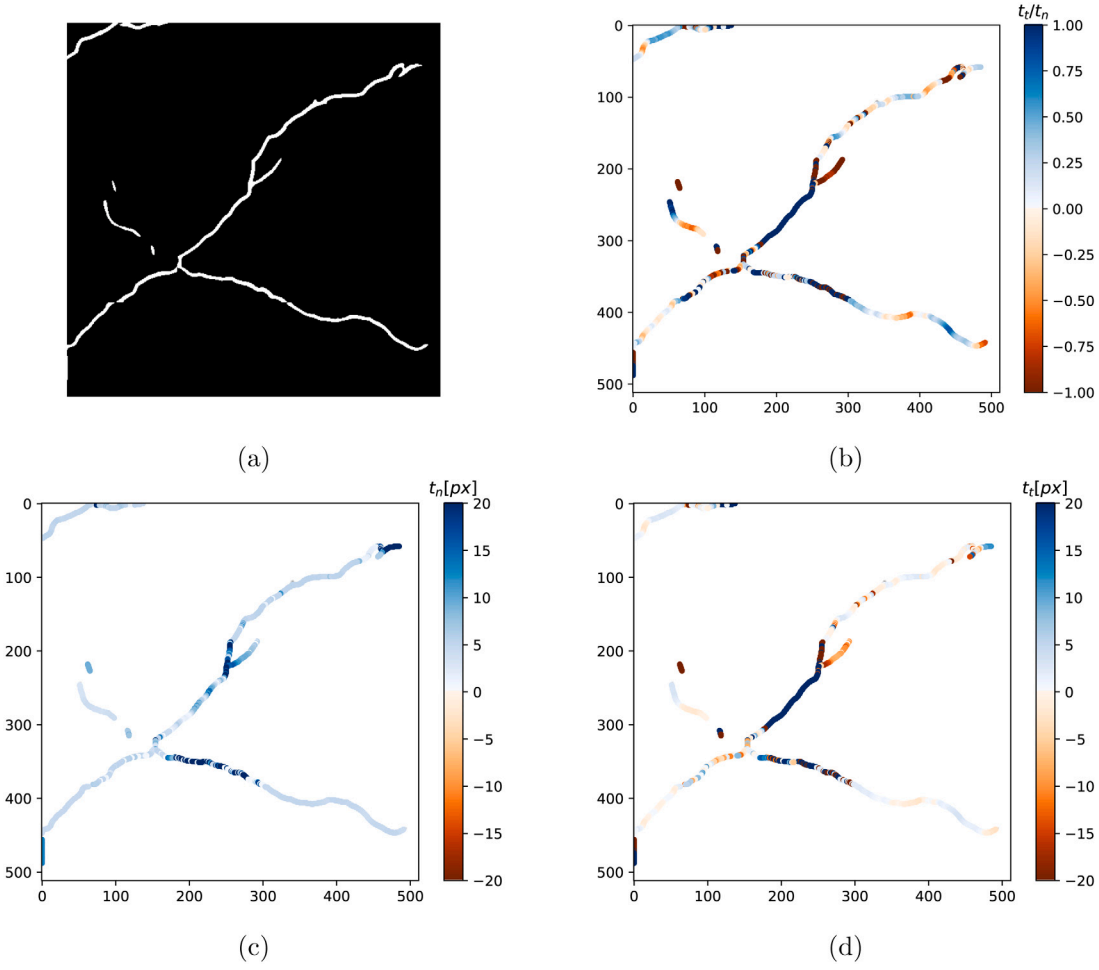


Fig. 27. Crack kinematics for post-earthquake damage assessment—crack kinematic for extracted patch 512 × 512 px: (a) binary mask, (b) crack kinematics—relation t_t/t_n, (c) normal displacement map (t_n), and (d) tangential displacement map (t_t).

terms of rotation and displacement maps. Note that the parameter k can be replaced by η, and the parameter λ can be replaced by the Pareto-like optimization algorithm [28].

Algorithm 1 Binary Image Crack Kinematics (BICK) algorithm

Input: k, μ, λ
Output: Crack Kinematic
  Crack Kinematics ← {}
  Crack Mask ← *read*(crack image path)
  Crack Contours ← *find_contour*(Crack Mask)
  Skeleton ← *skeletonize*(Crack Mask)
  Skeleton End Points ← *get_end_points*(Skeleton)
  Skeleton Junction Points ← *get_junction_points*(Skeleton)
  Crack Edges ← *find_crack_edges*(Crack Contours, Skeleton End Points, Skeleton Junction Points)
  for x, y in Skeleton **do**
    $\mathbf{E}_0$ ← *get_edge_0*(k, Crack Edges)
    $\mathbf{E}_1(1), \ldots, \mathbf{E}_1(M)$ ← *get_M_edges_1*(k, μ, λ, Crack Edges)
    for i in $\{1, \ldots, M\}$ **do**
      $\theta_i, t_{x_i}, t_{y_i}$, loss_i ← *get_euclidean_transformation*($\mathbf{E}_0$, $\mathbf{E}_1(i)$)
    end for
    θ, t_x, t_y ← *get_best_transformation*($\theta_i, t_{x_i}, t_{y_i}$,$\text{loss}_i$)
    t'_x, t'_y ← *get_2DOF_movement*(θ, t_x, t_y)
    t_n, t_t ← *get_2DOF_normal_and_tangential*(θ, t'_x, t'_y)
    Crack Kinematics ← *update_status*($\theta, t_x, t_y, \theta, t'_x, t'_y, t_n, t_t$)
  end for

Appendix B. Validation and selection of hyper-parameters—Monte Carlo simulation

To assess the robustness of our herein-developed technique, synthetic crack-like datasets were created as in Section 4.2 that generated the transformed edge by considering random translations and rotations. The experiments presented in this section consider both full and segmented edges and are composed of 50 binary masks of a crack pattern. Furthermore, the error measure considered in this analysis refers to the absolute error of the mean value of the translations $\mathbf{t}'$ (Eq. (8)), which is given by

$$\epsilon = \left|0.5(\delta t'_x + \delta t'_y)\right|, \qquad \begin{pmatrix}\delta t'_x \\ \delta t'_y\end{pmatrix} = \frac{1}{k}\sum_{i=1}^{k}\left[\mathbf{t}'_{gt} - \mathbf{t}'\right]. \tag{B.1}$$

The first analysis performed here used full edges, for which we obtained an average absolute error of 0.83 px and standard deviation of 0.68 px for 50 images. For the mean absolute error, the probability density function calculated using the Silverman bandwidth method is shown in Fig. B.28. The low error and variance obtained for these quantitative results indicate the potential of our methodology to solve the crack kinematic problem.

To verify the stability of our method, to perform a sensitivity analysis of the hyper-parameters used in our methodology, and to provide a guide on how to choose them, we ran Monte Carlo simulations for the finite edge segment approach considering four scenarios. In the first scenario (*Scenario* 1), we used the hyper-parameters k, μ, and λ. For the second (*Scenario* 2), we used the hyper-parameters η, μ, and λ.

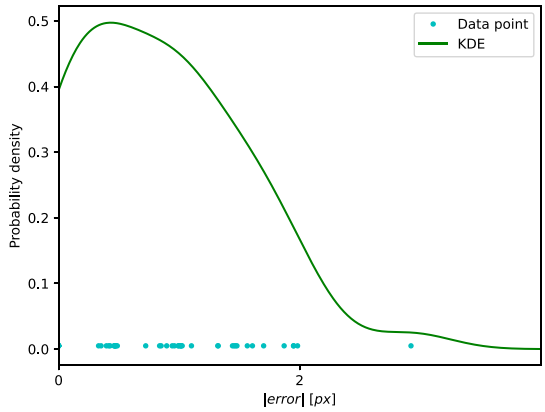


Fig. B.28. Probability density function for mean absolute error calculated using a kernel density estimation (KDE) and full edges for the 50 synthetic contour-based crack patterns.

Table B.7
Monte Carlo simulations of probability density functions. Mean and standard deviation (std) for the absolute error and registration time for four different scenarios.

	Mean absolute error		Registration time	
	mean [px]	std [px]	mean [s]	std [s]
Scenario 1	1.30	1.55	0.42	0.85
Scenario 2	1.32	1.40	0.40	0.69
Scenario 3	1.87	3.83	0.59	0.81
Scenario 4	2.27	5.29	0.57	0.77

The third and fourth scenarios (*Scenario* 3 and *Scenario* 4) used the same hyper-parameters as the first two scenarios except for λ (not an input hyper-parameter in this scenario), which was selected via the heuristic Pareto-like optimization algorithm. We randomly selected the hyper-parameter values for the four scenarios in the following intervals: $k = [30, 250]$, $\mu = [1.0, 3.0]$, $\lambda = [1, 10]$, and $\eta = [1, 5]$. To assess the influence of the hyper-parameters on the analysis results for the four scenarios, we performed a Monte Carlo simulation with a total of 5000 samples for each scenario, which represent different combinations of hyper-parameters and crack segments. To select the crack segment, we randomly sampled one point on the skeleton of one of the 50 synthetic contour-based crack patterns and registered the k-nearest neighbors of the edge $\mathbf{E}_0$ over the edge $\mathbf{E}_1$. We then determined the error for this sample according to Eq. (B.1).

For each of the four scenarios presented in Fig. B.29, the probability density functions for the mean absolute error and registration time were calculated using the Silverman method for the bandwidth. The mean and standard deviation for the absolute error and registration time obtained in terms of pixels and seconds respectively are presented in Table B.7. These values together with the distribution shape shown in Fig. B.29 of the probability density functions affirms that our methodology is stable and that the variation of the parameters in general does not significantly alter the final result. As expected, *Scenario* 3 and *Scenario* 4 had the largest error and variance, as the use of heuristics does not always result in the best solution. Nonetheless, the use of heuristics reduces computational costs as compared to the use of $\lambda = 1$ such that the application of our algorithm requires consideration of the trade-off between computational cost and accuracy. Note that the probability density functions show that some samples had a large error in comparison to the mean value. This occurs where the skeleton points are sampled from regions close to the image borders and one of the two opposite edges lays outside the patch. In this case, the finite segment edge will be registered incorrectly, producing the large errors observed in Fig. B.29.

Next, we clarified the sensitivity of our algorithm to the hyper-parameters, which guided the selection of appropriate values. This

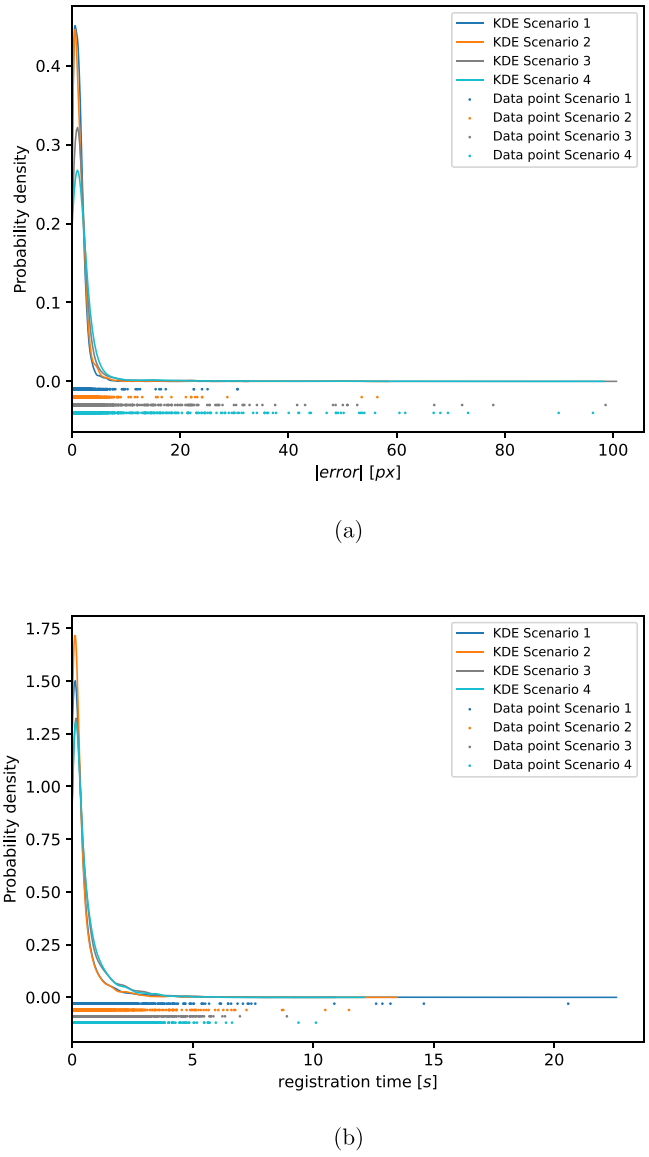


Fig. B.29. Probability density functions calculated using KDE for 5000 random samples in multidimensional spaces of hyper-parameters and using finite segment edges applied to the 50 synthetic contour-based crack patterns: (a) mean absolute error and (b) mean registration time.

was done using the variation of the mean of the absolute error and registration time as a function of the hyper-parameters for each scenario, presented in Figs. B.30 and B.31, respectively. Reasonably, the algorithms tend to produce smaller registration errors with a larger finite segment $\mathbf{E}_0$ (the larger k or smaller η; Fig. B.30a,b), as the larger the segment, the more features it will contain and therefore the better the registration. Despite this, larger finite segments increase the computational time, as shown in Fig. B.31a,b. In relation to the size of the segment $\mathbf{E}_1$ to which $\mathbf{E}_0$ is registered, the figures show that selecting a 50% bigger segment generally improved the results ($\mu \approx 1.5$; Fig. B.30c). For *Scenario* 1 and *Scenario* 2, selecting values of $\mu \geq 1.5$ results in approximately constant error values, as would be expected if the optimization result was the same every time. Predictably, this was not the case when a Pareto-like optimization algorithm was used (*Scenario* 3 and *Scenario* 4), because the algorithm can easily find a local minimum rather than a global minimum when the edge $\mathbf{E}_1$ is larger. In terms of computational cost, larger edge segments $\mathbf{E}_1$ (larger μ) proportionally increase registration time (Fig. B.31). Finally, for the

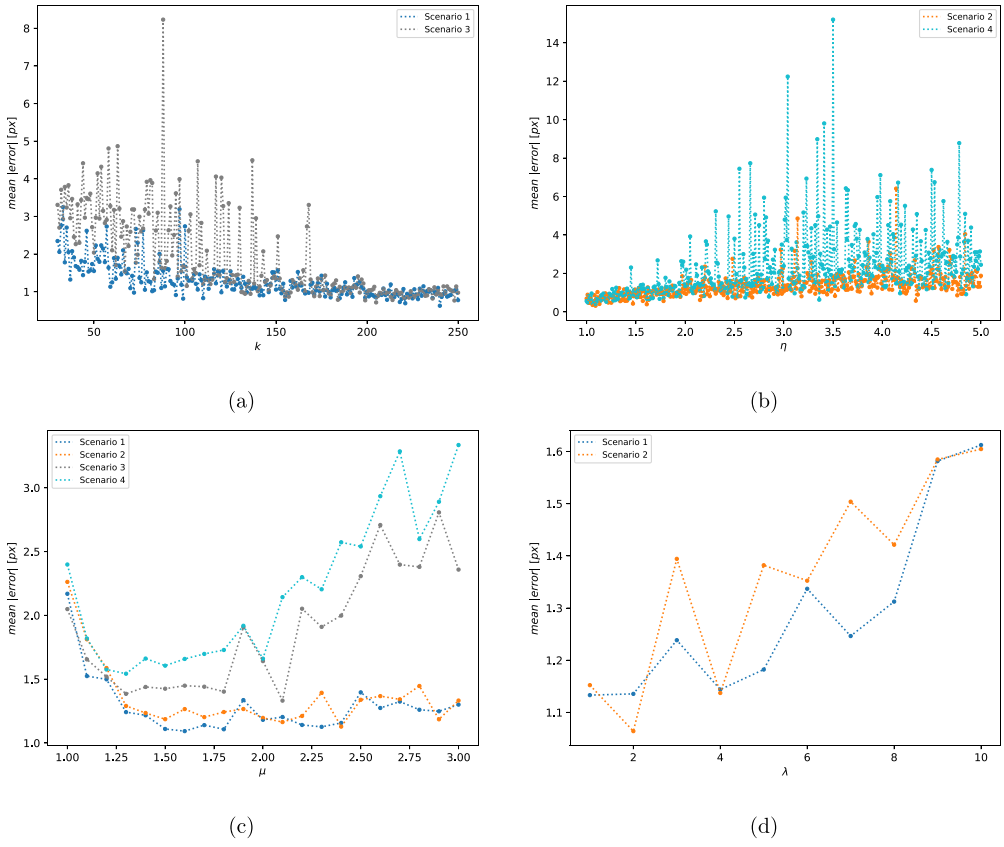


Fig. B.30. Mean of the absolute error during the Monte Carlo simulation for hyper-parameters used in the finite segment edge approach under the defined four scenarios: (a) k, (b) η, (c) μ, and (d) λ.

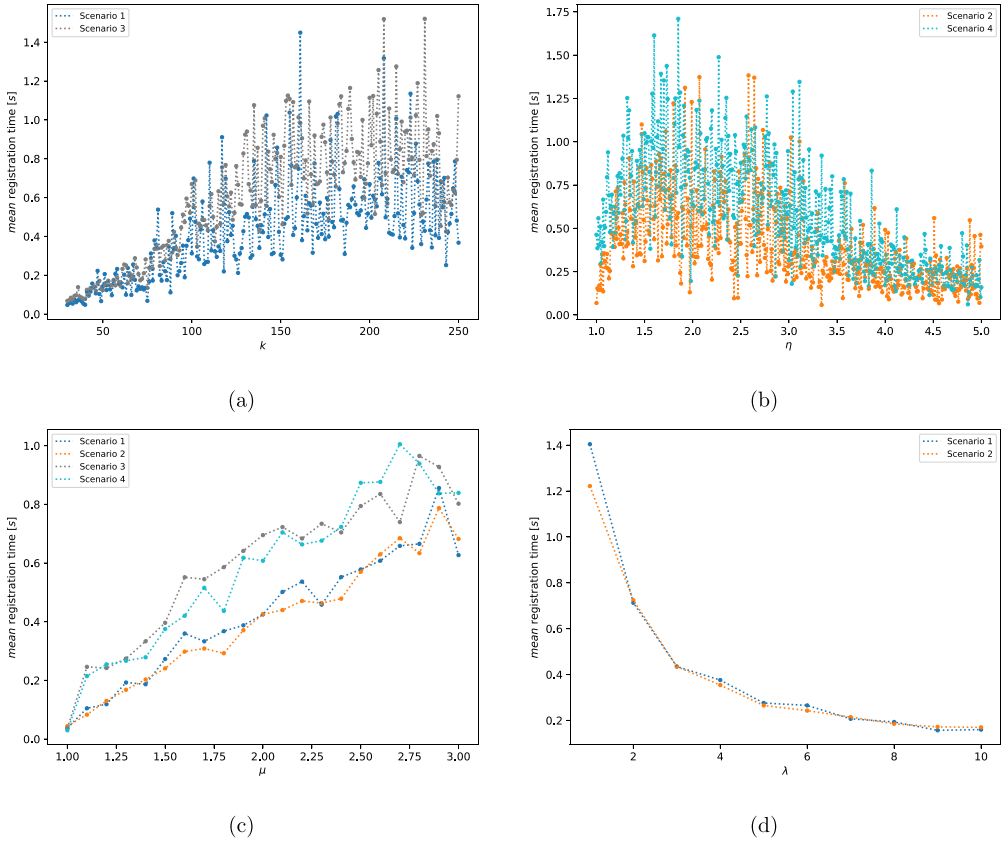
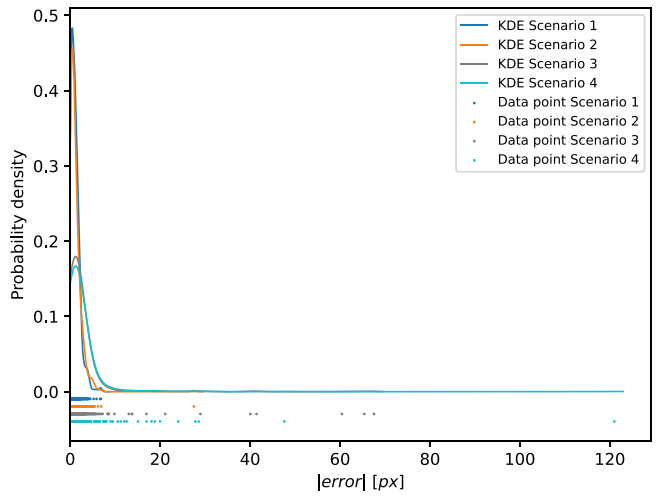
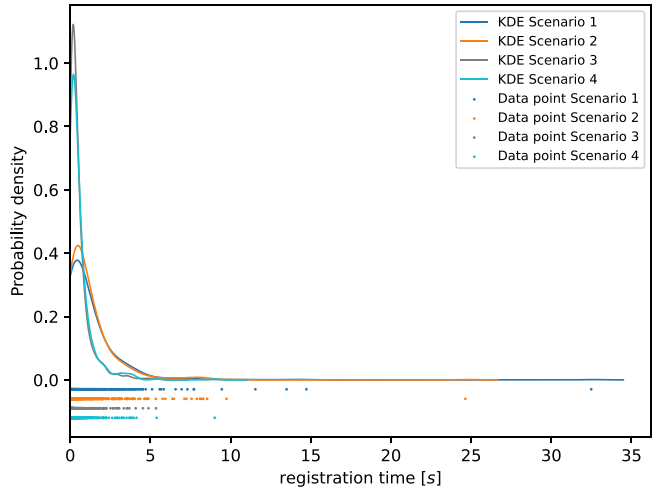


Fig. B.31. Mean of the registration time during the Monte Carlo simulation for hyper-parameters used in the finite segment edge approach under the defined four scenarios: (a) k, (b) η, (c) μ, and (d) λ.



(a)



(b)

Fig. B.32. Comparison between Pareto-like optimization algorithm and greedy approach with $\lambda = 1$. Probability density functions calculated using KDE on 500 random samples in multidimensional spaces of hyper-parameters and with the finite segment edge approach applied to the 50 synthetic contour-based crack patterns: (a) mean absolute error and (b) mean registration time.

two first scenarios that use λ, Figs. B.30d and B.31d predictably show the best error values when this hyper-parameter tends to unity, though this is at the expense of longer computation times. When $\lambda = 1$, the algorithm looks for all the possibilities of registering $\mathbf{E}_0$ over $\mathbf{E}_1$ and takes the best. Therefore, the use of heuristics is important for finding an appropriate solution that runs faster without sacrificing significant accuracy.

To test the efficiency of the Pareto-like algorithm, we ran four new Monte Carlo experiments with 500 samples under nearly identical conditions as in the previous four scenarios. The only difference between *Scenario* 1 and *Scenario* 2 was that the hyper-parameter λ was always equal to unity, which makes for a fair comparison of the improvement that the heuristic algorithm achieves in contrast to the greedy optimization approach, in which all registration possibilities between the edges $\mathbf{E}_0$ and $\mathbf{E}_1$ are examined ($\lambda = 1$). Fig. B.32 shows the probability density functions for the mean absolute error and registration time. The mean and standard deviation for the absolute error and registration time in terms of pixels and seconds, respectively, are presented in Table B.8. This result demonstrates an improvement in computational cost with the heuristic algorithm, though at the cost of increasing the error in the results. This again indicates the trade-off between processing time and precision that must be considered when deciding on an approach.

Table B.8
Monte Carlo simulations of probability density functions to measure the efficiency of the Pareto-like algorithm. Mean and standard deviation for the absolute error and registration time for four different scenarios.

	Mean absolute error		Registration time	
	mean [px]	std [px]	mean [s]	std [s]
Scenario 1	1.11	0.99	1.23	2.16
Scenario 2	1.15	1.61	1.24	1.80
Scenario 3	2.29	5.94	0.52	0.71
Scenario 4	2.27	6.47	0.57	0.83

As shown in [15], displacement prediction using DIC information may achieve errors of less than a pixel. When such numbers are compared to those in Tables B.7 and B.8, the accuracy of the approach provided here may be questioned. Nonetheless, our methodology's error acceptance is justified since it is intended to be employed in real-world applications where the environment is nearly impossible to control, as is the case with DIC approaches. As is well known, in order to employ DIC, certain circumstances, such as proper camera setup, controlled lighting, and speckle surfaces, are necessary. Furthermore, DIC achieves sub-pixel resolution through polynomial optimization and interpolation on a domain defined by pixels in integer positions. To achieve sub-pixel resolution, our technique would need to improve its optimization processes, which will be the subject of future research.

References

[1] A. Ebrahimkhanlou, A. Athanasiou, T.D. Hrynyk, O. Bayrak, S. Salamone, Fractal and multifractal analysis of crack patterns in prestressed concrete girders, J. Bridge Eng. 24 (7) (2019) 04019059, http://dx.doi.org/10.1061/(asce)be.1943-5592.0001427, URL https://ascelibrary.org/doi/10.1061/%t28ASCE%t29BE.1943-5592.0001427.

[2] K.B. Broberg, Differences between mode I and mode II crack propagation, Pure Appl. Geophys. 163 (9) (2006) 1867–1879, http://dx.doi.org/10.1007/s00024-006-0101-7.

[3] Y. Yao, E.T. Shue-Ting, B. Glisic, Crack detection and characterization techniques—An overview, Struct. Control Health Monit. 21 (2014) 1387–1413, http://dx.doi.org/10.1002/stc.1655, URL https://onlinelibrary.wiley.com/doi/epdf/10.1002/stc.1655?saml_referrer.

[4] B.F. Spencer, V. Hoskere, Y. Narazaki, Advances in computer vision-based civil infrastructure inspection and monitoring, Engineering 5 (2) (2019) 199–222, http://dx.doi.org/10.1016/j.eng.2018.11.030.

[5] T. Nishikawa, J. Yoshida, T. Sugiyama, Y. Fujino, Concrete crack detection by multiple sequential image filtering, Comput.-Aided Civ. Infrastruct. Eng. 27 (1) (2012) 29–47, http://dx.doi.org/10.1111/j.1467-8667.2011.00716.x, URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8667.2011.00716.x.

[6] T. Yamaguchi, S. Hashimoto, Fast crack detection method for large-size concrete surface images using percolation-based image processing, Mach. Vis. Appl. 21 (5) (2010) 797–809, http://dx.doi.org/10.1007/s00138-009-0189-8, URL https://link.springer.com/article/10.1007/s00138-009-0189-8.

[7] W. Zhang, Z. Zhang, D. Qi, Y. Liu, Automatic crack detection and classification method for subway tunnel safety monitoring, Sensors 14 (10) (2014) 19307–19328, http://dx.doi.org/10.3390/s141019307, URL https://www.mdpi.com/1424-8220/14/10/19307.

[8] A.P. Tewkesbury, A.J. Comber, N.J. Tate, A. Lamb, P.F. Fisher, A critical synthesis of remotely sensed optical image change detection techniques, Remote Sens. Environ. 160 (2015) 1–14, http://dx.doi.org/10.1016/j.rse.2015.01.006, URL https://www.sciencedirect.com/science/article/pii/S0034425715000152?via%3Dihub.

[9] M. Hussain, D. Chen, A. Cheng, H. Wei, D. Stanley, Change detection from remotely sensed images: From pixel-based to object-based approaches, ISPRS J. Photogramm. Remote Sens. 80 (2013) 91–106, http://dx.doi.org/10.1016/j.isprsjprs.2013.03.006.

[10] R.J. Radke, S. Andra, O. Al-Kofahi, B. Roysam, Image change detection algorithms: A systematic survey, IEEE Trans. Image Process. 14 (3) (2005) 294–307, http://dx.doi.org/10.1109/TIP.2004.838698, URL https://ieeexplore.ieee.org/document/1395984.

[11] B. Kim, S. Cho, Automated vision-based detection of cracks on concrete surfaces using a deep learning technique, Sensors 18 (10) (2018) http://dx.doi.org/10.3390/s18103452.

[12] Y.J. Cha, W. Choi, O. Büyüköztürk, Deep learning-based crack damage detection using convolutional neural networks, Comput.-Aided Civ. Infrastruct. Eng. 32 (5) (2017) 361–378, http://dx.doi.org/10.1111/mice.12263, URL https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12263.

[13] A. Zhang, K.C.P. Wang, B. Li, E. Yang, X. Dai, Y. Peng, Y. Fei, Y. Liu, J.Q. Li, C. Chen, Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network, Comput.-Aided Civ. Infrastruct. Eng. 32 (10) (2017) 805–819, http://dx.doi.org/10.1111/mice.12297, URL https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12297.

[14] A. Rezaie, R. Achanta, M. Godio, K. Beyer, Comparison of crack segmentation using digital image correlation measurements and deep learning, Constr. Build. Mater. 261 (2020) 120474, http://dx.doi.org/10.1016/j.conbuildmat.2020.120474.

[15] A. Rezaie, M. Godio, K. Beyer, Investigating the cracking of plastered stone masonry walls under shear-compression loading, Constr. Build. Mater. 306 (2021) http://dx.doi.org/10.1016/j.conbuildmat.2021.124831, URL https://www.sciencedirect.com/science/article/pii/S0950061821025800.

[16] N. Gehri, J. Mata-Falcón, W. Kaufmann, Refined extraction of crack characteristics in large-scale concrete experiments based on digital image correlation, Eng. Struct. 251 (October 2021) (2022) 113486, http://dx.doi.org/10.1016/j.engstruct.2021.113486.

[17] N. Gehri, J. Mata-Falcón, W. Kaufmann, Automated crack detection and measurement based on digital image correlation, Constr. Build. Mater. 256 (2020) 119383, http://dx.doi.org/10.1016/j.conbuildmat.2020.119383, URL http://www.sciencedirect.com/science/article/pii/S095006182031388X.

[18] G. Ruocci, C. Rospars, G. Moreau, P. Bisch, S. Erlicher, A. Delaplace, J.M. Henault, Digital image correlation and noise-filtering approach for the cracking assessment of massive reinforced concrete structures, Strain 52 (6) (2016) 503–521, http://dx.doi.org/10.1111/str.12192.

[19] A.F. Cinar, S.M. Barhli, D. Hollis, M. Flansbjer, R.A. Tomlinson, T.J. Marrow, M. Mostafavi, An autonomous surface discontinuity detection and quantification method by digital image correlation and phase congruency, Opt. Lasers Eng. 96 (February) (2017) 94–106, http://dx.doi.org/10.1016/j.optlaseng.2017.04.010.

[20] P.A. Korswagen, M. Longo, E. Meulman, J.G. Rots, Crack Initiation and Propagation in Unreinforced Masonry Specimens Subjected to Repeated In-Plane Loading During Light Damage, Vol. 17, Springer Netherlands, 2019, pp. 4651–4687, http://dx.doi.org/10.1007/s10518-018-00553-5.

[21] G.R. Irwin, Analysis of stresses and strains near the end of a crack traversing a plate, J. Appl. Mech. 24 (3) (1957) 361–364, http://dx.doi.org/10.1115/1.4011547.

[22] C.L. Rountree, R.K. Kalia, E. Lidorikis, A. Nakano, L. Van Brutzel, P. Vashishta, Atomistic aspects of crack propagation in brittle materials: Multimillion atom molecular dynamics simulations, Annu. Rev. Mater. Sci. 32 (2002) 377–400, http://dx.doi.org/10.1146/annurev.matsci.32.111201.142017.

[23] P.C. Hansen, V. Pereyra, G. Scherer, Least Squares Data Fitting with Applications, in: Least Squares Data Fitting with Applications, Johns Hopkins University Press, 2013.

[24] J.T. Betts, Solving the nonlinear least square problem: Application of a general method, J. Optim. Theory Appl. 18 (4) (1976) 469–483, http://dx.doi.org/10.1007/BF00932656, URL https://link.springer.com/article/10.1007%2FBF00932656.

[25] C. Maple, Geometric design and space planning using the marching squares and marching cube algorithms, in: Proceedings - 2003 International Conference on Geometric Modeling and Graphics, GMAG 2003, 2003, pp. 90–95, http://dx.doi.org/10.1109/GMAG.2003.1219671.

[26] T.Y. Zhang, C.Y. Suen, A fast parallel algorithm for thinning digital patterns, Commun. ACM 27 (3) (1984) 236–239, http://dx.doi.org/10.1145/357994.358023, URL https://dl.acm.org/doi/pdf/10.1145/357994.358023.

[27] T. Lee, Building skeleton models via 3-D medial surface/axis thinning algorithms, 1994, http://dx.doi.org/10.1006/gmip.1994.1043.

[28] M. Shaqfa, K. Beyer, Pareto-Like sequential sampling heuristic for global optimisation, Soft Comput. 25 (14) (2021) 9077–9096, http://dx.doi.org/10.1007/s00500-021-05853-8.

[29] L.P. Coelho, Mahotas: Open source software for scriptable computer vision, J. Open Res. Softw. 1 (1) (2013) e3, http://dx.doi.org/10.5334/jors.ac, arXiv:1211.4907.

[30] A. Rezaie, R. Achanta, M. Godio, K. Beyer, Dataset of cracks on DIC images, Zenodo (20.0) (2020) http://dx.doi.org/10.5281/zenodo.4307686, URL https://doi.org/10.5281/zenodo.4307686.

[31] E. Miranda, S. Brzev, N. Bijelic, Z. Arbanas, M. Bartolac, V. Jagodnik, D. Lazarevic, S. Mihalic Arbanas, S. Zlatovic, A. Acosta, Petrinja, Croatia December 29, 2020, Mw 6.4 earthquake, Joint Reconnaissance Report PRJ-2959, 2021, pp. 0–2, http://dx.doi.org/10.3929/ethz-b-000465058.

[32] J. Atalić, M. Uroš, M. Šavor Novak, M. Demšić, M. Nastev, The mw5.4 zagreb (Croatia) earthquake of march 22, 2020: impacts and response, Bull. Earthq. Eng. 19 (9) (2021) 3461–3489, http://dx.doi.org/10.1007/s10518-021-01117-w.