

Segmentation tool for images of cracks

Andrii Kompanets^{1,3}, Remco Duits^{2,3}, Davide Leonetti^{1,3}, Nicky van den Berg², and H.H. (Bert) Snijder¹

¹ Eindhoven University of Technology, Department of the Built Environment, Eindhoven, the Netherlands

² Eindhoven University of Technology, Department of Mathematics and Computer Science, Eindhoven, the Netherlands

³ Eindhoven Artificial Intelligence Systems Institute, Eindhoven University of Technology, the Netherlands

a.kompanets@tue.nl d.leonetti@tue.nl

Abstract. Safety-critical infrastructures, such as bridges, are periodically inspected to check for existing damage, such as fatigue cracks and corrosion, and to guarantee the safe use of the infrastructure. Visual inspection is the most frequent type of general inspection, despite the fact that its detection capability is rather limited, especially for fatigue cracks. Machine learning algorithms can be used for augmenting the capability of classical visual inspection of bridge structures, however, the implementation of such an algorithm requires a massive annotated training dataset, which is time-consuming to produce. This paper proposes a semi-automatic crack segmentation tool that eases the manual segmentation of cracks on images needed to create a training dataset for machine learning algorithm. Also it can be used to measure the geometry of the crack. This tool makes use of an image processing algorithm, which was initially developed for the analysis of vascular systems on retinal images. The algorithm relies on a multi-orientation wavelet transform, which is applied to the image to construct the so-called ‘orientation scores’, i.e. a modified version of the image. Afterwards, the filtered orientation scores are used to formulate an optimal path problem that identifies the crack. The globally optimal path between manually selected crack endpoints is computed, using a state-of-the-art geometric tracking method. The pixel-wise segmentation is done afterwards using the obtained crack path. The proposed method outperforms fully automatic methods and shows potential to be an adequate alternative to the manual data annotation.

Keywords: image segmentation · crack detection · computer vision · image processing · fatigue crack measurement · steel bridge inspection

1 Introduction

There are more than a million bridges in Europe and the USA, according to [1,2] and every bridge structure is subjected to gradual deterioration which may lead to its collapse. In order to prevent tragic events and to detect critical structure deterioration in time, periodic inspections of bridges have to be conducted

[3]. Visual inspections are the most frequent type of bridge inspection, during which, trained personnel visually examines the surface of the structure [4]. This procedure may be costly and time-consuming. Apart from the obvious expenses for inspector training and work, in some cases, there are implicit financial consequences to the regional economy related to the necessity to temporarily restrict or fully shut down traffic across a bridge.

In a recent paper from Campbell et al. [5], it has been highlighted that the variability of the outcome of visual inspections is substantial, and that the detection rate of fatigue cracks strongly depends on the inspection conditions and the proneness of the inspector to report potential damage. The authors concluded that, by using the current techniques and procedures, visual inspection should not be regarded as a reliable method for bridge inspection.

The substantial cost and the low reliability of manual bridge inspections explain recent efforts directed at the designing of automatic systems for bridge inspections [6]. This is because automated visual inspection systems have the potential to augment the efficiency of existing inspection practices and to reduce its cost. In recent works, special attention was paid to automatic fatigue crack detection, since fatigue is one of the most frequent and dangerous types of bridge damage [7].

Many computer vision algorithms have been developed for crack detection and measurement in different structures and materials, see for example overviews of this topic in [8,9]. These algorithms can be divided into two major groups, namely a) geometric image processing algorithms, and b) machine learning algorithms. Crack segmentation is a common approach extensively used in earlier works on automatic visual inspections, that besides crack detection, also allows measurement of the detected crack geometry. Image segmentation is the process of partitioning a digital image into multiple image segments, where each pixel is assigned to one of the segments. In the case of crack segmentation, two types of segments are considered: the segment of the image background type, and one or more segments of the crack type. Images of critical locations in bridge structures normally contain several elements that complicate the crack segmentation task. This may be surface corrosion, paint peeling, dirt on the surface, fasteners, and other bridge parts and joints. The performance of a fully automatic image processing algorithm dealing with such images is potentially affected by these aspects, e.g. it may result in a high false call rate by labeling a gap between adjoint elements of the considered structure as a crack, as mentioned in [10,11]. In contrast, computer vision machine learning algorithms not only can recognize geometrical patterns of cracks but can also distinguish them from other crack-like-looking parts of the image. However, the main drawback of machine learning algorithms is that they require a labeled dataset for training and testing. A large number of researchers developing machine learning algorithms for crack segmentation uses labeled datasets created by manual labeling. In this case, the annotation consists of a manual draw of the contour of the crack, most often made with a hand-held mouse [12,13]. Clearly, such a procedure may take a lot of time, especially in the case of a large number of high-resolution images.

Taking this into consideration, a manual annotation procedure forms a bottleneck in the process of development of a state-of-the-art machine learning algorithm for crack segmentation. In this paper, we propose to make use of image processing techniques to develop a semi-automatic crack segmentation tool. The developed tool has the advantage of making the process of labeling cracks on images simpler and faster, which is needed to train a machine learning algorithm for crack segmentation. Moreover, the proposed tool may also be used to extract the geometrical parameters (length, width, curvature, etc.) of a crack needed for either research or more accurate inspection. Unlike the fully automatic image processing technique for crack segmentation, the developed method requires manual input, i.e. the crack end-points. By adding such a manual input it is possible to significantly increase the accuracy of the crack segmentation algorithm, potentially allowing it to be used as a data labelling tool for machine learning algorithms. Implementation of the developed algorithm is available at <https://github.com/akomp22/crack-segmentation-tool>

2 Methods

Datasets to train a machine learning algorithm for crack segmentation are often manually labeled, since it is the most reliable method for pixel-wise image labeling. However, it is time consuming, because it requires a human expert to manually indicate the contours of the crack.

Fully automatic image processing algorithms do a crack segmentation much faster and without human involvement. A wide range of different image processing techniques have been created aiming at the development of a fully automatic crack segmentation algorithm. Commonly used techniques include thresholding [14,15,16,17,18], filtering [19,20], and image texture analysis [21,22,23].

A major drawback of these fully automatic algorithms is that they are often designed to mark dark elongated structures of the image as a crack. On complex images, for example the image of a steel bridge presented in the Fig. 1, these algorithms will suffer from the significant amount of false crack detections. Hence, these algorithms can not be used as an image pixel-wise labeling tool.

The semi-automatic algorithm described in this section uses an image processing technique, where the false call rate is minimized thanks to additional information about the crack location provided by a human annotator.

The first major step of the proposed method is crack path tracking, described in Section 2.1 of this paper. The algorithm aims at finding the path of a crack between two manually selected crack endpoints. The retrieved crack path allows to distinguish a particular elongated dark structure of the image (which is a crack) from other elements in the image that look like a crack (e.g. shading, paint, structure edge etc.). Few earlier works considering crack tracking having selected points as input. For instance, in [24], a crack path was identified by a step-by-step movement from one of the crack points in the direction that has a locally minimal pixel intensity. A disadvantage of this tracking method is that



Fig. 1: Example of a steel bridge structure image where common crack detection algorithms tend to have a high false call rate.

it is not able to find a globally optimal path. It can easily be misled by dark image regions near the crack and follow a locally optimal wrong path.

In [25] the crack path was found using only one internal crack point as input. A fast marching algorithm was used to find a distance map relative to the selected internal point. The lowest gradient ascent path was chosen as a crack path.

Crack minimal path selection approaches was also used in [26,27]. In these articles, Dijkstra's algorithm was used to find a crack path between multiple points inside crack contour. However, points inside crack contour were selected automatically, using the threshold method, so the problem of the high amount of false detections that is inherent to fully automatic image processing algorithms is also present in these methods.

The proposed crack tracking method uses an anisotropic fast marching algorithm to find a crack path as a globally optimal path (minimizing geodesic) in an orientation score (a multi-orientation wavelet transform) and is described in Section 2.1 of this paper.

To obtain a crack segmentation, the width of the crack along the retrieved path needs to be measured. One way to do this is to get an intensity profile perpendicular to the crack path, and determine crack edges based on this profile, as was done in [24]. However, the crack edges retrieved with this method have irregular shapes and may deviate significantly from the actual edges. A width expansion approach was used in [26,27,28], and is also used in the proposed algo-

rithm and explained in Section 2.2. An alternative width measurement method is also proposed (tracking algorithm for edges as described in Section 2.2) which works better on images of cracks in steel bridges.

2.1 Crack path tracking

In [29] a general image processing method was proposed for line tracking in 2D images via 3D multi-orientation distributions (so-called ‘orientation scores’). Such an orientation score [30] lifts the image-domain from a 2D position space to a 3D space \mathbb{M}_2 of 2D-positions and orientations, as can be seen in Fig. 2. Where a 2D image $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ assigns a greyvalue $f(x_1, x_2)$ to a position (x_1, x_2) , an orientation score $U_f : \mathbb{M}_2 \rightarrow \mathbb{C}$ assigns a complex value $U_f(x_1, x_2, \theta)$ to a ‘local orientation’ $p = (x_1, x_2, \theta)$. The real part of U_f gives the measure of alignment of the image local line structures with the specific orientation, whereas the imaginary part can be interpreted as the measure of alignment of image local edge structures with the specific orientation.

In [31] the tracking in the orientation scores (TOS) is done by formal optimal curve algorithms. The curve extraction is done with a single optimal control problem, allowing to include local contextual alignment models of oriented image features. In other words, optimality of the curve is determined not only by the intensity of pixels it lies on, but also by the alignment of this curve with the image’s local anisotropic structures.

These contextual geometric models in \mathbb{M}_2 improve tracking when cracks are partly visible, which is often the case. Furthermore, cracks often have sudden changes in local orientation, and do not have cusps like in [31]. Therefore, the used algorithm relies on the improved forward motion model also applied in [32] for blood vessel tracking. For efficient sufficiently accurate computation anisotropic fast marching algorithm [33] is used to compute distance maps in \mathbb{M}_2 . A subsequent steepest descent provides the optimal paths, i.e. optimal geodesics being paths with minimal data-driven length.

Such geometric TOS [32] gives a few advantages in the application of tracking cracks in steel bridges:

- When a crack is only partly visible in the image as in Fig. 3 the aforementioned contextual image processing models are needed to still recognize the crack
- Crossing structures (e.g. a crack and a sub-surface due to shading/paint) are disentangled in the orientation score (e.g. Fig. 4a and Fig. 5b). The crack and the edge/line crossing are separated because they align with different orientations, thus being on different “ θ levels” in the orientation score. Thereby a tracking algorithm will stay on a true crack trail.
- Cracks can suddenly change orientation and the model based on [34] automatically accounts for that.

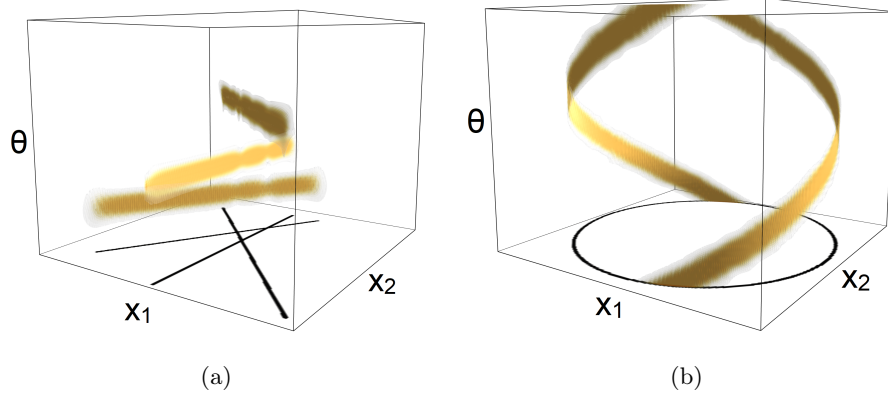


Fig. 2: Examples of the orientation scores: a) Orientation score of an image with lines; b) Orientation score of an image with a circle.



Fig. 3: Close up view of a crack with their calculated crack path retrieved by two methods: The red line shows the track obtained by tracking in \mathbb{R}^2 while the white line shows the spatially projected track obtained by tracking in $\mathbb{M}_2 = \mathbb{R}^2 \times S^1$.

Construction of the orientation score In the orientation score, the image domain of positions $\mathbf{x} := (x_1, x_2) \in \mathbb{R}^2$ is extended to the domain of positions and orientations (\mathbf{x}, θ) , where $\theta \in [0, 2\pi]$ denotes the orientation. The orientation score can be built from the initial image using the anisotropic cake wavelets shown in Fig. 4b (which thank their name to their shape in the Fourier domain [29,30]). The convolution of an image with a cake wavelet that has a specific orientation θ will filter the local line elements, giving high response in positions where the line structures are aligned with the applied wavelet orientation. For example, the responses in Fig.5a are obtained by applying the wavelet filters shown in Fig.4b to the image shown in Fig.4a.

A grayscale image can be considered as a (square integrable) function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ that maps a position $\mathbf{x} \in \mathbb{R}^2$ to a grayscale value $f(\mathbf{x})$. Similarly, the orientation score U_f is represented by the function $U_f : \mathbb{M}_2 \rightarrow \mathbb{C}$, where $\mathbb{M}_2 = \mathbb{R}^2 \times S^1$ and $S^1 := \{\mathbf{n}(\theta) = (\cos \theta, \sin \theta) \mid \theta \in [0, 2\pi)\}$. The used cake wavelets ψ are complex-valued (their real part detects lines whereas their imaginary part detects edges). The orientation score U_f of an image f is given by:

$$U_f(\mathbf{x}, \theta) = \int_{\mathbb{R}^2} \overline{\psi(R_\theta^{-1}(\mathbf{y} - \mathbf{x}))} f(\mathbf{y}) d\mathbf{y}, \text{ for all } \mathbf{x} \in \mathbb{R}^2, \theta \in [0, 2\pi),$$

where ψ is the complex-valued wavelet aligned with an a priori axis (say the vertical axis $\theta = 0$), and the rotation matrix R_θ rotates this wavelet counter-clockwise with the required angle θ and is defined by: $R_\theta = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$. A careful mathematical design of the cake wavelets [29,30] allows to preserve all image information after “lifting” the image to the orientation score. With a proper choice of parameters that are used to construct the cake-wavelet, the approximate reconstruction of the image f from the orientation score can be achieved by a simple integration:

$$f(\mathbf{x}) = \int_0^{2\pi} U_f(\mathbf{x}, \theta) d\theta.$$

Shortest paths (geodesics) in the orientation score After the input image is lifted to the orientation scores, a tracking algorithm is applied that finds the shortest path, or geodesic, between the endpoints of the crack in the orientation score, see Fig. 6. The geodesic can be represented by a parameterised curve $\gamma(t) = (\mathbf{x}(t), \mathbf{n}(t))$ in the orientation scores, the length of which is defined as the Riemannian distance between the chosen endpoints $\mathbf{p} = (\mathbf{x}_0, \mathbf{n}_0)$ and $\mathbf{q} = (\mathbf{x}_1, \mathbf{n}_1)$. Here, \mathbf{p} and \mathbf{q} are points in the lifted space of positions and orientations $\mathbb{M}_2 := \mathbb{R}^2 \times S^1$ and $\mathbf{n}_t := \begin{pmatrix} \cos \theta_t \\ \sin \theta_t \end{pmatrix}$. The asymmetric version of the Riemannian distance, that were used in the experiments, is defined by:

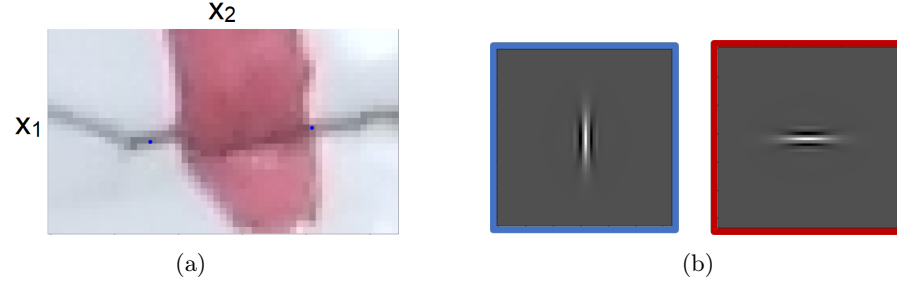


Fig. 4: a) Input image with chosen endpoints; b) Real part of the cake wavelets with angle θ equal to 0 and $\frac{\pi}{2}$ respectively.

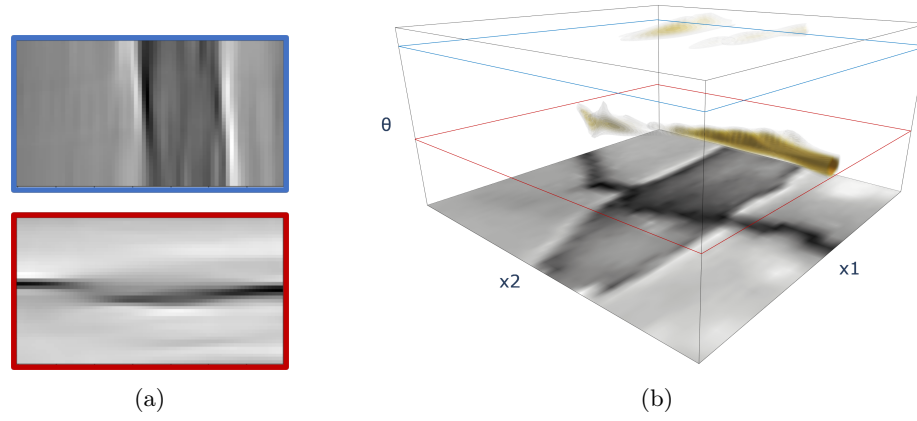


Fig. 5: a) Results of applying filters from figure Fig. 4b to the image shown in Fig. 4a; b) Orientation score of image shown in Fig. 4a. θ -levels represented by blue and red rectangles correspond to a).

$$d_G(\mathbf{p}, \mathbf{q}) = \inf_{\substack{\gamma(\cdot) = (\mathbf{x}(\cdot), \mathbf{n}(\cdot)) \in \Gamma \\ \gamma(0) = \mathbf{p}, \gamma(1) = \mathbf{q} \\ \dot{\mathbf{x}}(\cdot) \cdot \mathbf{n}(\cdot) \geq 0}} \int_0^1 \sqrt{G_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt \quad (1)$$

where the space of curves Γ (are all piecewise continuously differentiable curves $\gamma : [0, 1] \rightarrow \mathbb{M}_2$) over which we optimize, and with the Riemannian metric given by

$$G_{\mathbf{p}}(\dot{\mathbf{p}}, \dot{\mathbf{p}}) = C^2(\mathbf{p}) \left(\xi^2 |\dot{\mathbf{x}} \cdot \mathbf{n}|^2 + \frac{\xi^2}{\zeta^2} (\|\dot{\mathbf{x}}\|^2 - |\dot{\mathbf{x}} \cdot \mathbf{n}|^2) + \|\dot{\mathbf{n}}\|^2 + \lambda \frac{\max_{\|\dot{\mathbf{q}}\|=1} |HU|_{\mathbf{p}}(\dot{\mathbf{p}}, \dot{\mathbf{q}})|^2}{\max_{\substack{\|\dot{\mathbf{q}}\|=1, \\ \|\mathbf{p}\|=1}} |HU|_{\mathbf{p}}(\dot{\mathbf{p}}, \dot{\mathbf{q}})|^2} \right) \quad (2)$$

with $\mathbf{p} = (\mathbf{x}, \mathbf{n}) \in \mathbb{M}_2$ being a position and orientation, and $\dot{\mathbf{p}} = (\dot{\mathbf{x}}, \dot{\mathbf{n}})$ a velocity attached to the point \mathbf{p} , and where $C^2(\mathbf{p})$ is the output of the application of the multi-scale crossing/edge preserving line filter to the orientation score, for details see [32, App.D], [35], and acts as a cost function. The Hessian of the orientation score is denoted by HU . Parameter $\xi > 0$ influences the stiffness or curvature of the geodesics. Parameter $0 < \zeta \ll 1$ puts high cost on ξ^2/ζ^2 on sideward motion relative to the cost ξ^2 for forward motion. Parameter $\lambda > 0$ regulates the influence of the data-driven term relying on the Hessian of the orientation scores. In order to find the optimal curve that minimizes the distance, the anisotropic fast marching algorithm [33] is applied to solve the eikonal PDE [34]. The algorithm computes the distance map $d_G(\mathbf{p}, \cdot)$ given by Eq. (1) from one of the crack endpoints, \mathbf{p} , that serves as an initial condition (i.e. ‘seed point’) using an efficient propagating front approach. Afterwards, it finds the geodesics by backtracking using the steepest descent method [34] from the other endpoint \mathbf{q} , also called the ‘tip point’, using the computed distance map. The HFM library [36] was used to employ the described tracking algorithm efficiently.

Next, extra motivation for the curve optimization model given by Eq. (1) is provided. When considering crack propagation as a random Brownian process [37,38] on \mathbb{M}_2 then Brownian bridges concentrate on geodesics [39]. Furthermore, the model given by Eq. (1) corresponds to a curvature-adaptive extension of the model in [34] whose optimal geodesics include ‘in-place rotations’ that the model automatically places at optimal locations during the geodesic distance front-propagation. Such ‘in-place rotations’ (due to the constraint $\dot{\mathbf{x}} \cdot \mathbf{n} \geq 0$ in Eq. (1)) are natural for 2D pictures of cracks in bridges, as there are often sudden changes in direction of a crack.

2.2 Crack width detection

Two distinct approaches are considered that allow to obtain a crack segmentation while having the crack path. The first approach is called ‘width expansion’ (WE) and the second approach ‘edge tracking’ (ET). The WE approach is expected to work better with cracks that have a conspicuous grain structure along their edges, e.g. cracks in pavement. In contrast, the ET approach gives better results for cracks that have relatively smooth edges such as cracks in steel.

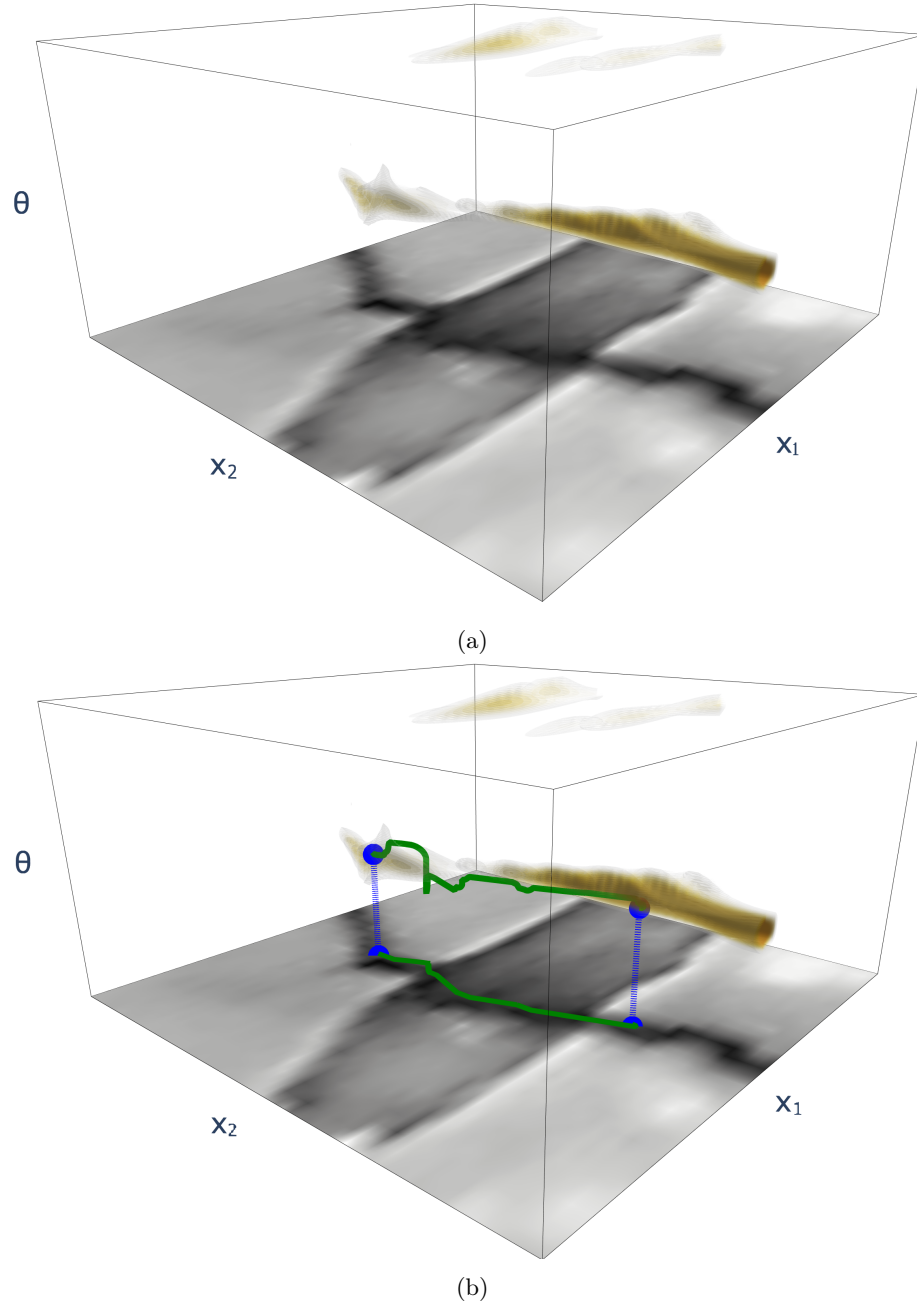


Fig. 6: a) Orientation score; b) An example of a geodesic (the shortest path γ from Eq. 1) in the orientation score and its projection onto \mathbb{R}^2 .

Crack width expansion This approach has been used in automatic crack segmentation algorithms [26,27], and is applied to the image in the \mathbb{R}^2 domain without any preliminary transformations. In the first iteration of the WE algorithm, the pixels that lie exactly on the obtained crack path are assigned to a crack segment. In the subsequent iterations, pixels neighboring to the existing crack segment are assigned to a crack segment if their intensity value is below a threshold value $T_w = \mu_w - K_w \cdot \sigma_w$, where μ_w and σ_w are mean and standard deviation respectively of the existing crack segment pixels gray value distribution, and where K_w is a manually chosen parameter. The iterations continue until no new pixels are added to the crack segment. This approach is useful when the crack has an irregular shape as is the case for e.g. cracks in pavement, due to the noticeable grains of the material. However, the output of this approach depends significantly on the threshold value K_w . The necessity to accurately choose K_w restricts the autonomy of the method, since, its optimal value may vary significantly depending on the image.

Crack edge tracking Similarly as for crack path tracking, for the ET a geometric tracking algorithm is applied to find an optimal path between crack endpoints, but with a few major differences.

First of all, a cost function $C(\mathbf{p})$ for the tracking algorithm is built that forces it to follow the crack edges instead of the crack centerline. To build this cost function, the crack path is divided into small segments for which crack path orientation is determined. Afterwards, around each path segment, a square part of image is chosen with a predetermined dimensions in parallel and perpendicular to the segment orientation. To these pieces of the image, an edge filter (Gaussian first order derivative) oriented in accordance with the crack path orientation is applied. Results of this operations are summed to construct a cost function for the whole image. In this way, the edge filter highlights mainly the edges of the crack and gives lower responses to edges not parallel to the crack. An example of a resulting image is presented in Fig. 7a.

Additionally, in the proposed ET method we use a version of the tracking algorithm described in Section 2.1 that in fact is the Dijkstra's algorithm in the image \mathbb{R}^2 domain. As a filter was applied that was specifically designed to highlight crack edges and ignore other edges on the image, it is expected to avoid the problems listed in Section 2.1 (namely crossing structures and poor visibility of the filtered crack edges). Hence, instead of doing tracking in the \mathbb{M}_2 domain as was done for crack path tracking, here the fast marching algorithm [33] for geodesic curve optimisation is directly applied in the image \mathbb{R}^2 domain that is computationally less expensive.

3 Results

3.1 Tracking performance

First, the performance of the crack path tracking algorithm is evaluated against the steel structures dataset. The used dataset consists of images of actual steel



Fig. 7: a) Oriented edge filter applied to the image shown in the Fig. 4a; b) Crack edges detected with the ET method.

bridge structures with fatigue cracks that was collected by Dutch infrastructure authorities. Ground truth crack tracks (actual track of a crack as it is visible on the image) were drawn manually for 19 randomly selected images from the dataset. In order to numerically estimate the performance of the tracking method, the metric $m = \frac{A}{L^2}$ is introduced, where A is the area between the ground truth track and retrieved track, measured in squared pixel, and L is the length of the ground truth track determined as the number of pixels it passes. The area A between two tracks represents a measure of how one track deviates from another. Division by L^2 makes the metric resolution-invariant. Lower values of the metric m mark lesser deviation of the retrieved crack from the ground truth track. Table 1 shows the mean value of the introduced metric.

The described tracking method was compared with the alternative methods. Table 1 shows the results of this comparison, where the considered algorithms are identified as:

- FF — the "FlyFisher" algorithm described in [24]. In this method a track propagates a track from the starting point and led by the local gray-scale pixel values patterns;
- DT — Tracking 2D paths using Dijkstra's algorithm in the \mathbb{R}^2 domain of the input image to minimize the cost function defined by the Frangi filter [40];
- TOS — Tracking in the Orientation Score: Tracking optimal paths defined by Eq. (1) in the orientation score via anisotropic fast marching algorithm (described in Subsection 2.1).

Table 1: Average deviation of three different crack tracking algorithms from the ground truth crack on 19 images of cracks in steel structures.

	FF	DT	TOS
$m \cdot 10^5$	4.46	0.980948	0.784308

The results demonstrate that the proposed TOS method outperforms DT and FF algorithms. The path retrieved by the FF method follows a locally optimal direction (defined by a darker image region) but easily misses a globally optimal path. This problem results in the worst performance (see Table 1). In contrast, the TOS and the DT algorithm use a two step approach. First, they calculate the distance map on a Riemannian manifold (M, G) from the starting point and then perform a backtracking using the steepest descent algorithm. In case of TOS, the manifold $M = \mathbb{M}_2$ and metric tensor field G given by Eq. (2) while in the DT algorithm, the base manifold $M = \mathbb{R}^2$ and the metric tensor field \mathcal{G} is now given by $\mathcal{G}_{\mathbf{x}}(\dot{\mathbf{x}}, \dot{\mathbf{x}}) = V(\mathbf{x})\|\dot{\mathbf{x}}\|^2$ with V the Frangi filter [40].

This two step approach allows us to find the optimal path resulting in significantly lower values of the metric m in Table 1. Here the DT algorithm follows the path with the lowest cumulative Frangi filter values V -intensity of pixels. The TOS algorithm improves upon this by using a crossing-preserving V -intensity [35], and moreover it includes alignment of locally oriented image features as explained in Section 2.1, resulting in better tracking performance (see Table 1).

3.2 Segmentation on the AigleRN dataset

The AigleRN dataset contains images of cracks in pavement and provides ground truth crack segmentations [41]. In order to segment cracks on images from this dataset, the TOS method is used with the WE as described in Section 2.2. Seventeen images were randomly selected from the AigleRN dataset for the evaluation. To evaluate the performance of the tracking algorithm, criterions were used as in [28], namely precision, recall and F1-value. Recall shows what fraction of the crack pixels were retrieved by an algorithm. Recall equal to 1 means that all pixels that belong to a crack according to ground truth were identified by an algorithm as crack pixels. The precision value indicates what fraction of the pixels that were identified as crack by an algorithm, actually belong to a crack segment. Finally, the F1-value is the harmonic mean of precision and recall. Results are provided in Table 2.

In [41] crack segmentation results retrieved by minimal path selection (MPS) [27] and free-form anisotropy (FFA) [42] algorithms are also provided. Using these provided segmentation results we also evaluated the performance of the MPS and the FFA methods which are also shown in Table 2 for the same seventeen images.

Table 2: Crack segmentation results on the AigleRN dataset.

	FFA	MPS	TOS+WE
Precision	0.61	0.82	0.91
Recall	0.32	0.6	0.75
F1	0.36	0.63	0.82

The FFA algorithm measures a local texture anisotropy for every pixel of the image. The pixels where an anisotropy is higher than a threshold value are assigned to a crack segment. This approach has the lowest performance.

The MPS algorithm finds potential crack pixels using a threshold of local minima in grayscale. Afterwards, the paths between these points are computed using Dijkstra’s algorithm. These paths constitutes the so-called ‘skeleton’ of the crack. To find crack segments, this skeleton is used as a basis for the WE algorithm as explained in Section 2.2. The MPS algorithm does not allow to detect cracks with poor visibility, and this explains the low recall value. This can also be observed in Fig. 8, where the middle part of the crack with partial visibility was not detected by the MPS method. Also, in Fig. 8 we can observe false detections by the MPS algorithm, whereas the TOS+WE algorithm avoids false detections and maintains the connectivity of the crack. Because of this effects the proposed TOS+WE algorithm outperforms its two counterparts by all three metrics as can be seen on Table 2.

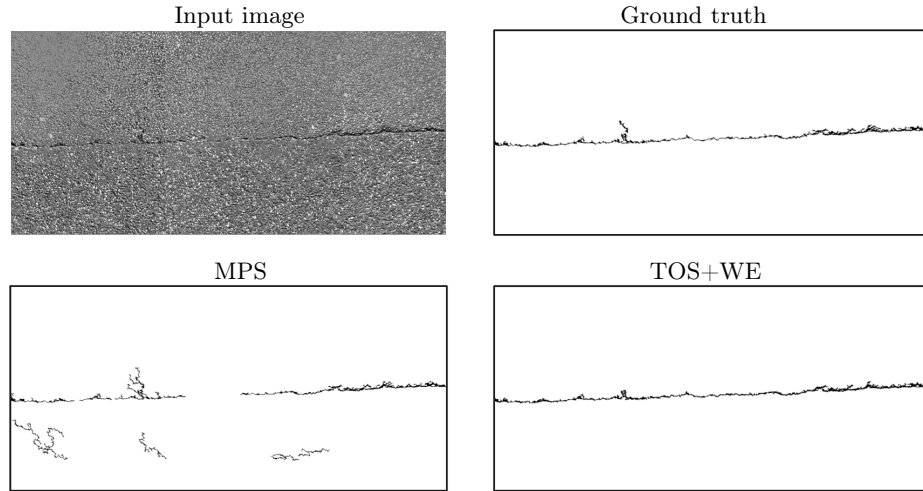


Fig. 8: Typical example in the AigLRN dataset with segmentations. The output of the TOS+WE algorithm is more connected and closer to ground truth than the MPS-output. This qualitatively supports the performance differences in Table 2.

3.3 Segmentation on steel structures images

Finally, results are provided of the application of the segmentation algorithm to the images of steel bridge structures with cracks from the dataset introduced in Section 3.1. Ten images with visible crack edges from the dataset were used for the evaluation. Unlike in pavement, crack edges in steel structures have a smooth shape, and previously it was stated that the ET algorithm should work

better than the WE algorithm in this case. Here this is shown by comparing the TOS+ET and TOS+WE algorithms in Table 3.

In Table 3, also the results for the MPS algorithm are added for comparison. As was explained in the introduction of this paper, the fully automatic MPS method tends to have lots of false crack pixels identifications on images with non-crack dark elongated elements as can be seen on Fig. 9. This leads to only 49% precision for the MPS algorithm and low F1-value.

In contrast, with the proposed algorithms (TOS+WE and TOS+ET) the image structure of interest is identified at first using the crack path tracking. Thus, the precision of the proposed algorithms is higher than that of the MPS algorithm.

Furthermore, TOS+WE method has a slightly higher precision value than the TOS+ET method, but significantly lower recall value meaning that TOS+WE algorithm skips more crack pixel, marking them as background pixels. Thereby, by the F1-value the TOS+ET method shows the best performance on the images of steel structures from our dataset.

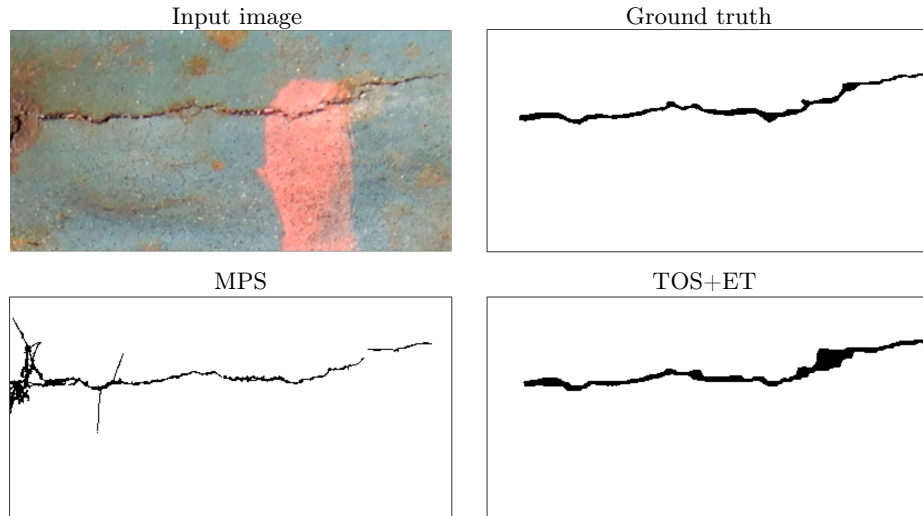


Fig. 9: Example of a steel bridge image and the corresponding results of its segmentation. The TOS+ET method suffers less from neighboring/crossing structures (paint, rust) because of the multi-orientation decomposition.

4 Conclusion

An algorithm for the crack segmentation was proposed, consisting of two major parts. The first part of the algorithm measures a crack track between manually selected crack endpoints. The second part of the algorithm performs the crack

Table 3: Crack segmentation results on the dataset of images of steel structures.

	MPS	TOS+WE	TOS+ET
Precision	0.49	0.9	0.86
Recall	0.66	0.69	0.85
F1	0.52	0.78	0.83

segmentation using a crack track obtained in the first part as a basis. The crack tracking part of the algorithm adapts the approach initially developed for the analysis of vascular systems on retinal images where an efficient fast marching algorithm performs line tracking on the image lifted into space of positions and orientations. For the width measurement part of the algorithm the novel approach was proposed for the crack edge tracking and it was compared with the width expansion approach that was used in earlier works.

The proposed semi-automatic algorithm allows for a crack segmentation of images with improved accuracy compared to the fully automatic algorithms, such as minimal path selection and free form anisotropy algorithms and resolves the problem of high false detections which is inherent to fully automatic crack detection image processing algorithms. While sacrificing autonomy of the segmentation algorithm, the accuracy significantly increases.

Potentially the developed algorithm may be used as a data labeling tool, to label images with cracks to train a machine learning algorithm. However, results show that the proposed algorithm does not fully reproduce the manual pixel-wise labeling. It should be studied how much this deviation affects the performance of the machine learning algorithm trained on the data labeled with the semi-automatic segmentation tool. Also, the algorithm can find its use in cases when it is necessary to measure a crack's geometry (length, width, curvature etc.) when the location of this crack is known, e.g. for research or for the purpose of more accurate inspection.

Acknowledgement

Authors would like to thank the Dutch bridge infrastructure owners "ProRail" and "Rijkswaterstaat" for their support. The research is primarily funded by the Eindhoven Artificial Intelligence Systems Institute, and partly by the Dutch Foundation of Science NWO (Geometric learning for Image Analysis, VI.C 202-031).

References

1. K. Brady, M. O'Reilly, L. Bevc, A. Znidaric, E. O'Brien, and R. Jordan. Procedures required for assessing highway structures-background, details and progress. Technical Report COST 345, European commission directorate general transport and energy, 2003.

2. Y. Kitane and A. J. Aref. Sustainable replacement of aging bridge superstructures using fiber-reinforced polymer (FRP) composites. *Advanced Composites in Bridge Construction and Repair*, pages 287–322, 5 2014.
3. R. Helmerich, B. Jan, J. R. Casas, P. Cruz, G. Holm, H. Geir, C. Kammel, M. Krüger, E. Niederleithinger, P. Rawa, et al. Guideline for Inspection and Condition Assessment of Existing European Railway Bridges. Technical Report SB-ICA, European commission, 11 2007.
4. M. Rossow. *FHWA Bridge Inspector's Manual: Bridge inspection program*. Continuing Education and Development, Inc., 2012.
5. L. E. Campbell, R. J. Connor, J. M. Whitehead, and G. A. Washer. Benchmark for evaluating performance in visual inspection of fatigue cracking in steel bridges. *Journal of Bridge Engineering*, 25(1), 2020.
6. B.-W. Jo, Y. Sung Lee, J.-H. Kim, and K.-W. Yoon. A review of advanced bridge inspection technologies based on robotic systems and image processing. *International Journal of Contents*, 14(3), 2018.
7. B. M. Imam and M. K. Chryssanthopoulos. Causes and consequences of metallic bridge failures. *Structural Engineering International*, 22(1):93–98, 2012.
8. A. Mohan and S. Poobal. Crack detection using image processing: A critical review and analysis. *Alexandria Engineering Journal*, 57(2):787–798, 2018.
9. X. W. Ye, T. Jin, and C. B. Yun. A review on deep learning-based structural health monitoring of civil infrastructures. *Smart Structures and Systems*, 24(5):567–585, 2019.
10. T. Tong, J. Lin, J. Hua, F. Gao, and H. Zhang. Crack identification for bridge condition monitoring using deep convolutional networks trained with a feedback-update strategy. *Maintenance, Reliability and Condition Monitoring*, 1(2):37–51, 2021.
11. C. M. Yeum and S. J. Dyke. Vision-based automated crack detection for bridge inspection. *Computer-Aided Civil and Infrastructure Engineering*, 30(10):759–770, 2015.
12. J. J. Kim, A. R. Kim, and S. W. Lee. Artificial neural network-based automated crack detection and analysis for the inspection of concrete structures. *Applied Sciences (Switzerland)*, 10:1–13, 11 2020.
13. W. Qiao, B. Ma, Q. Liu, X. Wu, and G. Li. Computer vision-based bridge damage detection using deep convolutional networks with expectation maximum attention module. *Sensors (Switzerland)*, 21:1–18, 2 2021.
14. A. M. A. Talab, Z. Huang, F. Xi, and L. HaiMing. Detection crack in image using otsu method and multiple filtering in image processing techniques. *Optik*, 127(3):1030–1033, 2016.
15. Q. Li and X. Liu. Novel approach to pavement image segmentation based on neighboring difference histogram method. In *2008 Congress on Image and Signal Processing*, volume 2, pages 792–796. IEEE, 2008.
16. Z. Yiyang. The design of glass crack detection system based on image preprocessing technology. In *2014 IEEE 7th Joint International Information Technology and Artificial Intelligence Conference*, pages 39–42. IEEE, 2014.
17. L. Li, Q. Wang, G. Zhang, L. Shi, J. Dong, and P. Jia. A method of detecting the cracks of concrete undergo high-temperature. *Construction and Building Materials*, 162:345–358, 2018.
18. H. Oliveira and P.L. Correia. Crackit—an image processing toolbox for crack detection and characterization. In *2014 IEEE international conference on image processing ICIP*, pages 798–802. IEEE, 2014.

19. M. Vidal, M. Ostra, N. Imaz, E. García-Lecina, and C. Ubide. Analysis of sem digital images to quantify crack network pattern area in chromium electrodeposits. *Surface and Coatings Technology*, 285:289–297, 2016.
20. T. C. Hutchinson, M. Asce, and Z. Chen. Improved image analysis for evaluating concrete damage. *Journal of Computing in Civil Engineering*, 20:1–7, 2006.
21. Y. Huang and B. Xu. Automatic inspection of pavement cracking distress. *Journal of Electronic Imaging*, 15:13–17, 1 2006.
22. T. S. Nguyen, M. Avila, and S. Begot. Automatic detection and classification of defect on road pavement using anisotropy measure; automatic detection and classification of defect on road pavement using anisotropy measure. In *17th European Signal Processing Conference*, pages 617–621, Glasgow, Scotland, 2009. IEEE.
23. Y. Hu. A local binary pattern based methods for pavement crack detection. *Journal of Pattern Recognition Research*, 1:140–147, 2010.
24. P. Dare, H. Hanley, C. Fraser, B. Riedel, and W. Niemeier. An operational application of automatic feature extraction: the measurement of cracks in concrete structures. *The Photogrammetric Record*, 17(99):453–464, 2002.
25. V. Kaul, A. Yezzi, and Y. Tsai. Detecting curves with unknown endpoints and arbitrary topology using minimal paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1952–1965, 2011.
26. R. Amhaz, S. Chambon, J. Idier, and V. Baltazart. A new minimal path selection algorithm for automatic crack detection on pavement images. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 788–792. IEEE, 2014.
27. R. Amhaz, S. Chambon, J. Idier, and V. Baltazart. Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection. *IEEE Transactions on Intelligent Transportation Systems*, 17(10):2718–2729, 2016.
28. Y. Chen, J. Liang, X. Gu, Q. Zhang, H. Deng, and S. Li. An improved minimal path selection approach with new strategies for pavement crack segmentation. *Measurement*, 184:109877, 2021.
29. E. Bekkers, R. Duits, T. Berendschot, and B. T. H. Romeny. A multi-orientation analysis approach to retinal vessel tracking. *Journal of Mathematical Imaging and Vision*, 49:583–610, 2014.
30. R. Duits. *Perceptual organization in image analysis: a mathematical approach based on scale, orientation and curvature*. PhD thesis, Eindhoven University of Technology, 2005.
31. E. J. Bekkers, R. Duits, A. Mashtakov, and G. R. Sanguinetti. A pde approach to data-driven sub-riemannian geodesics in $SE(2)$. *SIAM Journal on Imaging Sciences*, 8(4):2740–2770, 2015.
32. Nicky van den Berg, Bart Smets, Gautam Pai, Jean-Marie Mirebeau, and Remco Duits. Geodesic tracking via new data-driven connections of cartan type for vascular tree tracking. *arXiv e-prints*, pages arXiv–2208, 2022.
33. J.-M. Mirebeau. Anisotropic fast-marching on cartesian grids using lattice basis reduction. *SIAM journal of numerical analysis*, 52:1573–1599, 1 2014.
34. R. Duits, S. P. L. Meesters, J.-M. Mirebeau, and J. M. Portegies. Optimal paths for variants of the 2d and 3d reeds–shepp car with applications in image analysis. *Journal of Mathematical Imaging and Vision*, 60:816–848, 7 2018.
35. Julius Hannink, Remco Duits, and Erik Bekkers. Crossing-preserving multi-scale vesselness. In Polina Golland, Nobuhiko Hata, Christian Barillot, Joachim Hornegger, and Robert Howe, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*, pages 603–610, Cham, 2014. Springer International Publishing.

36. J.-M. Mirebeau. Adaptive grid discretization - A set of tools for discretizing anisotropic PDEs on cartesian grids kernel description.
37. R. Duits and E. M. Franken. Left invariant parabolic evolution equations on $SE(2)$ and contour enhancement via invertible orientation scores, part I: Linear left-invariant diffusion equations on $SE(2)$. *Quarterly of Applied mathematics, AMS*, 68:255–292, June 2010.
38. J. Zhang, R. Duits, B. M. Romeny, and G. R. Sanguinetti. Numerical approaches for linear left-invariant diffusions on $SE(2)$, their comparisons to exact solutions, and their applications in retinal imaging. *Numerical Mathematics: Theory Methods and Applications*, 9(1):1–50, January 2016.
39. O. Wittich. An explicit local uniform large deviation bound for Brownian bridges. *Statistics & probability letters*, 73(1):51–56, 2005.
40. A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale vessel enhancement filtering. In *International conference on medical image computing and computer-assisted intervention*, pages 130–137. Springer, 1998.
41. Rabih Amhaz, Sylvie Chambon, Jérôme Idier, and Vincent Baltazart. Aiglern dataset. https://www.irit.fr/~Sylvie.Chambon/Crack_Detection_Database.html. Accessed on 2022-05-8.
42. T. S. Nguyen, S. Begot, F. Duculty, and M. Avila. Free-form anisotropy: A new method for crack detection on pavement surface images. In *2011 18th IEEE International Conference on Image Processing*, pages 1069–1072. IEEE, 2011.