

Crack detection in randomly textured surfaces using spiral search

Du-Ming Tsai, Yi-Jun Xie

Abstract—Cracks are critical defects in material surfaces such as steel and concrete. Many of the currently available methods mainly focus on crack detection in uniform or non-textured surfaces. This paper presents a machine vision scheme for crack detection in randomly textured surfaces, which could arise from the natural pattern of a material surface, or the result of a high-resolution image. Dark objects in the image are presented by cracks and the background texture. It makes the detection task very difficult.

The proposed method involves preprocessing and pixel-connection. It first uses the automatic thresholding and morphological thinning to convert dark objects to single pixel thickness. A spiral search then performs to link dark pixel points based on their neighboring distances and direction angles. The distance and angle in the search process can be obtained from a pre-constructed spiral look-up-table. No computation of the geometric features is required, and thus the search of the closest neighboring point is very fast. Experimental results on a test set of 120 metal images show the proposed method can detect all cracks without false alarms. It is also demonstrated that the proposed method can be applied to non-textured images, such as concrete surfaces.

Index Terms—Crack detection; Defect inspection; Edge linkage; Spiral search.

I. INTRODUCTION

Cracks are severe defects in material surfaces such as steel plates and concrete structures. Crack detection prevents the damages, degradations, or failures of end-user products. Non-contact inspection of cracks using machine vision techniques has attracted great attention in industry in the last decade.

For metal surfaces in the manufacturing industry, spatial and spectral methods have been proposed to detect cracks in steel materials. Landstrom and Thurley (2012) presented a morphology-based crack detection method for steel slabs. The 3D profile image is first segmented using the morphological operations. The resulting connected regions are assigned a crack probability using a logistic regression model. Senthikumar et al. (2014) used the iterative thresholding technique to detect cracks in metal surfaces. Choi et al. (2012) and Choi et al. (2014) used the Gabor filter to detect seam cracks in steel plates. The geometric and gray-level features of suspected cracks are further evaluated using the support vector machine (SVM) to remove noise. Their proposed methods are restricted to the detection of vertical and horizontal cracks. Malekian et al. (2012) proposed a series of image processing operations to detect

and localize cracks in hot steel slabs. The wavelet transform is first used to sharpen the crack edge and weaken noise. Anisotropic diffusion filter and adaptive Gaussian filter are then applied to smooth the image. An edge detector is further used to detect edge points of potential cracks. Morphological filtering is carried out to remove the background noise and fill the gaps of a crack. K-means clustering with color features, and neural networks with geometrical features of clustered regions are also proposed as an alternative to detect cracks in steel slabs. The extracted features in their paper focus only on vertical and horizontal cracks. Aarathi et al. (2013) applied the wavelet transform to detect cracks in metal surfaces. Statistical features in the Haar-wavelet decomposed image and the SVM classifier are used to show the pixel points of a crack. Neogi et al. (2014) reviewed vision-based methods for steel surface inspection. It discussed spatial domain- and spectral domain-based methods for the detection of various types of defects in steel surfaces.

For concrete surfaces in the construction industry, the image processing approach becomes important and effective for non-destructive testing and monitoring. Cracks in concrete surfaces are usually dark objects that show thin lines and local symmetry across their center axes. Nguyen et al. (2014) utilized the two crack properties to design a filter for crack detection in the concrete surface image. They then applied thresholding to segment the filtered image, followed by morphological thinning and cubic splines to fit the pixel points of a possible crack. Zhang et al. [2014] proposed an automatic crack detection method for subway tunnel monitoring. Morphological operations and thresholding are used to initially segment crack and non-crack background. Distance histogram-based shape descriptor and a classifier are then further used to remove mis-identified objects. Fujita and Hamamoto [2011] studied crack detection from noisy concrete surfaces. They first subtracted the original image from its media-filtered image to extract possible cracks. The multi-scale line filter with the Hessian matrix is then applied to retain line-shaped objects. Probabilistic relaxation and a locally adaptive thresholding are finally performed to detect cracks. The computation of the line-filter and the probability are very time-intensive, and the detected pixels of a crack may not be connected. Yamaguchi et al. [2008] and Yamaguchi and Hashimoto [2010] considered the crack point connection in the concrete image as a percolation model. The central pixel in a local window is evaluated according to a cluster formed by the percolation processing using the brightness and shape criteria of a crack. Choudhary and Dey [2012] used fuzzy logic and neural networks to classify cracks in concrete surfaces. Prasanna et al. [2012] extracted the gray-level histogram-based features from the concrete image. The SVM classifier is then applied to identify cracks in the concrete surface. Hu et al. [2010] proposed a Hough transform (HT) based method to extract crack features in concrete or metal surfaces. The SVM classifier with the extracted HT features is

Du-Ming Tsai, Department of Industrial Engineering & Management, Yuan-Ze University, Tao-Yuan, Taiwan, Tel: (+886) 463-8907

Yi-Jun Xie, Department of Industrial Engineering & Management, Yuan-Ze University, Tao-Yuan, Taiwan, Tel: (+886) 463-8907

then used to identify crack points in the image.

The currently available machine vision methods for crack detection in material surfaces generally assume that the surface in the sensed image is uniform or non-textured. Many of the methods are based on local crack features, and detect only individual pixels of a crack in the image. They may not connect the detected pixels to form the complete shape of the crack. This could be a serious problem when parts of a crack are missing. The discontinuous crack may not be fully connected in nature, or is due to the low-contrast intensity with respect to its surroundings in the image.

In this study, we present a machine vision scheme for crack detection in randomly textured surfaces. The texture could be the natural pattern of a material surface. It could also be the result of magnifying micro-cracks in a high-resolution image. Many metal surfaces under high-resolution imaging create detailed texture patterns in the images. The proposed method will be evaluated with micro-crack detection in casting surfaces. Figure 1 demonstrates the metal surfaces in high-resolution images, where (a) is a defective surface that contains a fine, thin crack, and (b) is a defect-free surface. The metal surface under high-resolution imaging presents a random texture. The crack is line-shaped, but does not appear as a straight line. Some portions of the crack are disconnected. The crack in Figure 1(a) is indeed darker than its surroundings. However, the background also presents random dark line-shaped objects. Figures 1(c) and (d) further display the Sobel edge images of the metal surfaces images in (a) and (b), respectively. The dark edges are randomly presented in the gradient images. The gradient magnitudes cannot visually distinguish the crack from the background edges. As seen in Figure 1(c), the gradient magnitudes of most crack edges are smaller than those of the randomly-textured edges.

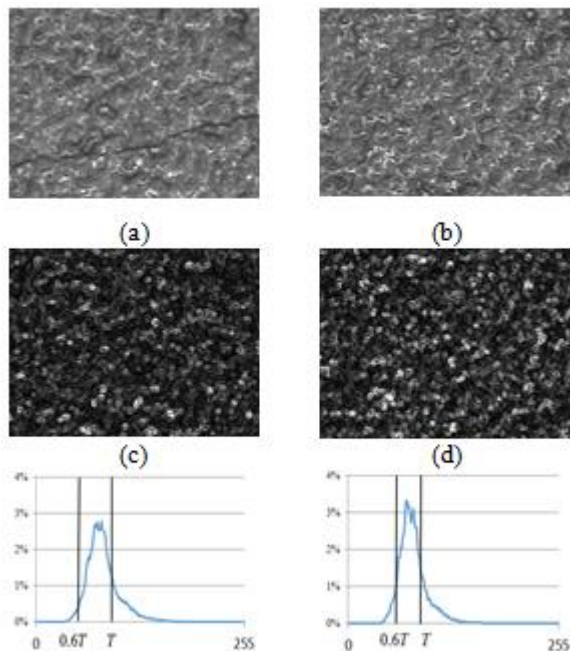


Figure 1. Randomly textured metal surfaces: (a) defective image with a crack; (b) defect-free image; (c), (d) respective gradient images of (a) and (b); (e), (f) respective gray-level histograms of (a) and (b).

To detect and locate irregular and discontinuous cracks in

randomly-textured surfaces, we propose a fast spiral search algorithm to connect the individual pixel points of a crack. The original gray-level image of a metal surface is converted to a binary image using a simple thresholding to segment the dark objects in the image. The morphological thinning is then applied to the binary image so that the resulting dark objects are 1-pixel width. The spiral search process is performed to the thinned dark points. To trace the pixel points along a crack, an object point in the thinned image is taken as the center of the spiral. The spiral then emanates from the center and the first encountered point gives its closest neighbor. It is well suited to trace the discontinuous segments of the crack. If there are more than one neighboring points in the search, the one with the coherent direction is chosen for the connection. The irregular and discontinuous line cracks can thus be effectively extracted from the random background.

To efficiently trace the neighboring points with the shortest distance and the most consistent direction using the spiral search, a spiral Look-Up-Table (LUT) is constructed off-line. It stores the x - and y -coordinates, distance and direction angle of each spiral point in the sequence with respect to the spiral center. The point number in the spiral sequence is used as the index of the LUT. It allows fast access to the geometric features without any computation. The search of the closest neighboring point can then be simplified. The proposed method can be effectively and efficiently applied to crack detection in either non-textured or randomly-textured surfaces.

The rest of the paper is organized as follows. Section 2 presents the creation of a digital spiral and the construction of the spiral look-up-table. It then describes the spiral search procedure for the linkage of crack points. Section 3 discusses the experimental results and the performance of the proposed crack detection method on randomly-textured metal surfaces. The concluding remarks are addressed in Section 4.

II. THE CRACK DETECTION METHOD

The proposed method for crack detection in randomly-textured surfaces involves four phases, which are:

- 1) Thresholding: convert the gray-level image into a binary image using automatic thresholding, where the black objects are possible cracks and random background textures.
- 2) Thinning: use the morphological thinning to reduce all black objects in the binary image to single pixel thickness.
- 3) Linking object pixels by spiral search: use the spiral search to link dark pixels, either connected or dis-connected, in the thinned image into line segments.
- 4) Connecting line segments: connect line segments to form a complete crack based on the neighborhood distances and directions of individual line segments.

The core of the proposed method is mainly based on the spiral search. The off-line creation of a digital spiral sequence and the construction of the spiral Look-Up-Table are first described. The detailed procedure of the crack detection is then presented.

A. Creation of digital spiral and LUT

A simple form of spirals is given by the Archimede's spirals, which is mathematically defined by

$$r = a \cdot \theta \quad (1) \quad 8\text{-connected}$$

where r is the radial distance of a point on the spiral with respect to the spiral center;

θ is the polar angle, measured in radians;

a is a constant that controls the spacing between successive turnings.

A point p with polar angle $\theta(p)$ on the spiral can be defined in Cartesian coordinates of a digital image by

$$x(p) = \text{Int}[a \cdot \theta(p) \cdot \cos\theta(p)] \quad (2a)$$

$$y(p) = \text{Int}[a \cdot \theta(p) \cdot \sin\theta(p)] \quad (2b)$$

A complete spiral must cover all pixel points of a given region in the digital image. The distance between two successive turnings of the spiral is given by $a \cdot 2\pi$. This distance should be equal to 1 pixel in the digital image to make the spiral complete. Thus, the spiral constant a must not be larger than $1/2\pi$. Figure 2 demonstrates the spiral sequences with varying constant values of a in the digital images. As seen in Figure 2(a), the spiral sequence does not cover the full region of a given spiral radius when the constant is set at 0.5. Some pixel points in the image will never be traversed. In Figure 2(b), the region is complete, and no empty space is found when the constant is set to a value less than $1/2\pi$.

The simplified algorithm for generating the spiral sequence in a digital image is listed as Algorithm I below. Given the current point number p in the spiral sequence, the polar angle of the subsequent spiral point $p+1$ is initially set to $1/(a \cdot \theta(p))$ since a 2π circle of radius $r(p)$, i.e. $a \cdot \theta(p)$, containing the current point p has $2\pi \cdot r(p)$ points on the circle. A binary search is then applied to iteratively update the increment of the polar angle $\theta(p+1)$ from $\theta(p)$ until the consecutive points p and $p+1$ are 8-connected.

Algorithm I: Creation of spiral sequence with the spiral

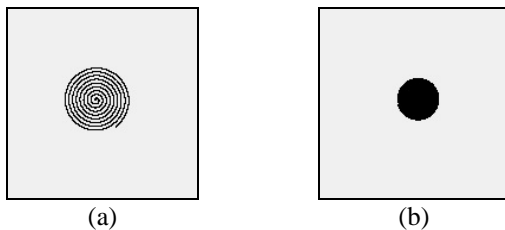


Figure 2. The spirals in digital images: (a) $a = 0.5$; (b) $a = 0.1$.

center at $(0, 0)$

Let p_{\max} be the maximum point number of the spiral sequence. For a predetermined spiral size of radius r_{\max} , it is given by $p_{\max} = \text{Max}\{p \mid r(p) \leq r_{\max}\}$.

Let $(x(0), y(0)) = (0, 0), \theta(0) = 0$, and $a < 1/2\pi$

For $p = 0, 1, 2, \dots, p_{\max} - 1$

$$\theta(p+1) = \begin{cases} \theta(p) + 1/(a \cdot \theta(p)), & \text{if } p > 0 \\ \pi/4, & \text{otherwise} \end{cases}$$

$$\theta_{\text{New}} = \theta(p+1); \theta_{\text{Old}} = \theta(p)$$

Do until $\text{Max}\{|x(p+1) - x(p)|, |y(p+1) - y(p)|\} = 1$ (i.e.

$$\varepsilon_{\theta} = |\theta_{\text{New}} - \theta_{\text{Old}}|$$

$$\theta_{\text{Old}} = \theta(p+1)$$

$$x(p+1) = \text{Int}[a \cdot \theta(p+1) \cdot \cos\theta(p+1)]$$

$$y(p+1) = \text{Int}[a \cdot \theta(p+1) \cdot \sin\theta(p+1)]$$

If

$$|x(p+1) - x(p)| + |y(p+1) - y(p)| = 0,$$

$$\theta(p+1) \leftarrow \theta(p+1) + \varepsilon_{\theta} / 2$$

If

$$[(x(p+1) - x(p))^2 + (y(p+1) - y(p))^2]^{1/2} > \sqrt{2},$$

$$\theta(p+1) \leftarrow \theta(p+1) - \varepsilon_{\theta} / 2$$

$$\theta_{\text{New}} = \theta(p+1)$$

EndDo

Next p

Once the digital spiral sequence is created, the geometric features of a point p on the spiral are given by the pixel location $(x(p), y(p))$, the distance $d(p)$ with respect to the spiral center, and the direction angle $\theta^0(p)$ measured in degrees and restricted in the range between 0° and 360° , i.e.

$$d(p) = [x(p)^2 + y(p)^2]^{1/2}$$

$$\theta^0(p) = [\theta(p) \bmod 2\pi] \cdot 360^\circ / 2\pi$$

where *mod* is the modulo operation. The spiral Look-Up-Table is thus constructed by the point coordinates $(x(p), y(p))$, distance $d(p)$ and rotation angle $\theta^0(p)$ with the point number p as the index. Table 1 shows the sketch of the spiral LUT. Note that the spiral point number p also implicitly indicates the distance or neighborhood from the spiral center. That is, a smaller point number p is closer to the spiral center. With the spiral LUT, the computation of distances and direction angles, and the sorting of distances can be eliminated in the search of neighboring points.

B. Preprocessing with thresholding and thinning

As seen in Figure 1(a) and (b), the crack intensities are darker than their surroundings. Figure 1(e) and (f) present the corresponding gray-level histograms of the defective metal image 1(a), and the defect-free image 1(b), respectively. The two gray-level histograms of the metal images with random textures are unimodal. There is no such a threshold that can clearly segment the crack from the background texture. However, the threshold value should be selected so that most of the pixels associated with the crack are preserved, while the noisy background pixels are removed as many as possible. In order to automatically find the adaptive threshold for each individual test image, we initially use the Otsu's minimum within-group variance [Otsu, 1979] to find the preliminary threshold value T . The resulting value from the Otsu's method is generally in favor of a large portion of dark regions. We thus reduce the threshold value, and take $0.6 \cdot T$ as the final threshold value for the binary segmentation. For a given type of test surfaces, the reduction coefficient can be empirically determined and is fixed in the inspection process.

For the demonstration images in Figure 1(a) and (b), the selected threshold values T and $0.6 \cdot T$ are displayed in the histograms in Figure 1(e) and (f). The resulting binary images with the threshold values T and $0.6 \cdot T$ are shown in Figure 3, where 3(a) and (b) are the same defective and defect-free test images as those in Figure 1(a) and (b), and 3(c) and (d) are the respective binary images with the Otsu's threshold T , and 3(e) and (f) are the resulting binary images with the reduced threshold $0.6 \cdot T$. The thresholding result of the defective metal image reveals that the Otsu's threshold value cannot segment the crack, as seen in Figure 3(c). The reduced threshold $0.6 \cdot T$ can well segment the dark crack, as shown in Figure 3(e). The thresholding process also generates irregular

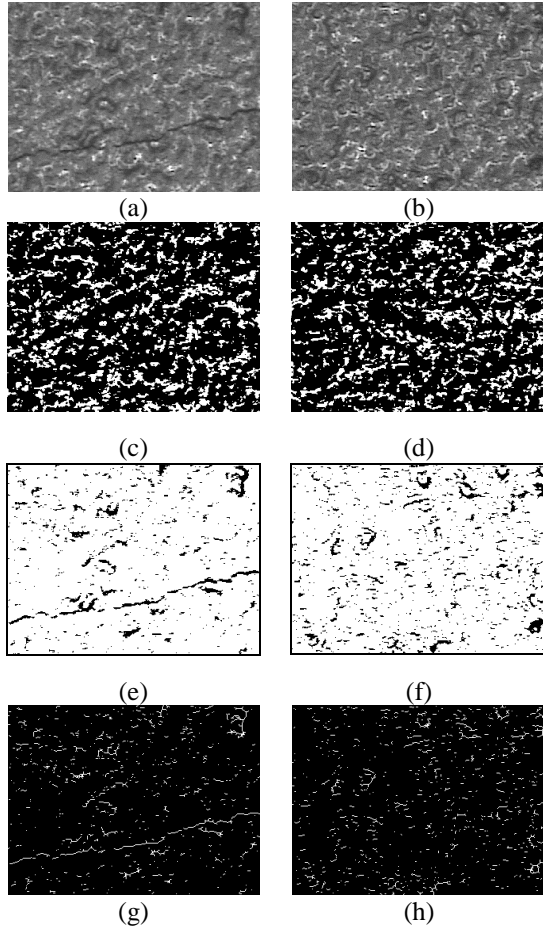


Figure 3. Preprocessing results of the metal surfaces: (a), (b) defective and defect-free images; (c), (d) respective binary thresholding results from the Otsu's threshold T ; (e), (f) respective binary images from $0.6T$; (g), (h) respective results of morphological thinning.

line- and curve-shaped dark objects, along with random noisy points, from the background textures for both defective and defect-free metal images.

For the resulting binary image with the reduced threshold value $0.6 \cdot T$, the morphological thinning operation [Gonzalez and Woods, 2008] is further applied to make the black objects one-pixel thickness. Figure 3(g) and (h) show the thinning results of the binary images in Figure 3(e) and (f), respectively. The crack in the thinned image is visually observable in the highly noisy background. Using the thresholding and thinning as preprocessing tools for the randomly textured image is due to their simplicity and

computational efficiency. Although random noisy objects are present in the resulting thinned image, the crack is also well preserved. The proposed spiral search can effectively connect the pixels of a crack from the noisy background. It is robust enough to link continuous or discontinuous crack lines, and is not sensitive to the preprocessing results.

C. Spiral search for pixel linkage

Let $t(x, y)$ be the resulting thinned image, where $t(x, y)=1$ if (x, y) is a black object pixel. It is otherwise set to zero for the background. Since the neighboring pixel points of a crack may not be connected in the thinned image, a spiral of a small radius r_{max} is used to search for the closest neighbor.

The radius r_{max} defines the maximum distance between two discontinuous points of a crack. In this study, r_{max} is set at 4 pixels.

Multiple line segments of different orientations may be present in the thinned image $t(x, y)$. The spiral search is applied to the end point of a currently connected line segment, i.e. the end point is taken as the spiral center. If the first encountered point is 8-connected to the end point of the line segment, it is then linked to the line segment. If no 8-connected points can be found, it then finds a neighboring point that is closest to the end point and shows a consistent orientation of the current line segment.

Let $L(l) = \{p_i^l = (x_i^l, y_i^l), i = 0, 1, 2, \dots, e\}$ be the line segment labeled by l that contains a collection of connected pixel points p_i^l 's. $p_e^l = (x_e^l, y_e^l)$ is the end point of the line segment $L(l)$. To find the subsequent neighboring point of the line segment $L(l)$, its end point (x_e^l, y_e^l) is taken as the spiral center, and the spiral search proceeds. Denote by $\bar{\theta}_e^l$ the direction angle towards the end of the line segment $L(l)$. It is given by

$$\bar{\theta}_e^l = \frac{1}{K} \sum_{k=1}^K \theta_{e-k}^l$$

where θ_i^l is the direction angle of point p_i^l , and is obtained from the LUT in the spiral search; K is the support length. The support K is currently set to 4 in this study, i.e. the last four points of a line segment is used to calculate the mean direction angle.

In the spiral search, the minimum number of p in the spiral sequence (i.e. the minimum distance from the spiral center) with an acceptable angular difference is selected and is linked to the current segment $L(l)$. That is,

$$p^* = \arg \min_p \{d(p) \mid \Delta\theta_p^0 \leq T_\theta \ \& \ t(x_e^l + x(p), y_e^l + y(p)) = 1\}$$

where $\Delta\theta_p^0 = |\bar{\theta}_e^l - \theta^0(p)|$; $\theta^0(p)$ is the rotation angle of spiral point p and is obtained from the spiral LUT. T_θ is the angular tolerance threshold. In this study, it is given by 30° . Note that the sorting for the minimum distance is not required with the spiral search. The point p^* with coordinates $(x_e^l + x(p^*), y_e^l + y(p^*))$ is then assigned to the line segment $L(l)$. It is the new end point of the line segment. Record also $\theta^0(p^*)$ as the direction angle of the new connected point. The spiral search then repeats to link the

unassigned pixel points in the thinned image. Figure 4 shows the sketch of dark-pixels linkage, where p_i is the neighboring pixel of the end point (x_e^l, y_e^l) and shows better consistency with the direction of the line segment

The simplified algorithm of the spiral search process is presented by the pseudo-code in Algorithm II below.

Algorithm II: Spiral search for pixel linkage

Let $L(l)$ be the line segment, and (x_e^l, y_e^l) the current end point of the line segment.

$$\text{Let } (c_x, c_y) = (x_e^l, y_e^l); \bar{\theta}_e^l = \frac{1}{K} \sum_{k=1}^K \theta_{e-k}^l$$

For $p = 1, 2, \dots, P_{\max}$

$$x = c_x + x(p)$$

$$y = c_y + y(p)$$

If $t(x, y) = 1$ and $(x, y) \& (c_x, c_y)$ are 8-connected, Or

$$t(x, y) = 1 \text{ and } \Delta\theta_p^l = |\bar{\theta}_e^l - \theta^0(p)| \leq 30^\circ,$$

Then

$$L(l) \leftarrow L(l) \cup (x, y)$$

$$e \leftarrow e + 1$$

$$(x_e^l, y_e^l) = (x, y), \text{ and } \theta_e^l = \theta^0(p)$$

EndIf

Next p

Repeat the spiral search above with the new spiral center (x_e^l, y_e^l) . If no pixel point can be connected in the spiral region of radius r_{\max} , terminate the linkage of line segment l . Connect the unassigned pixel points in the thinned image with

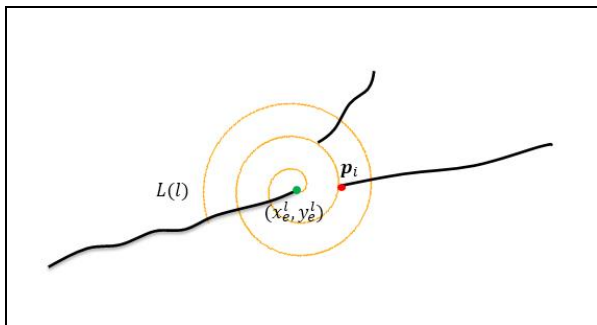


Figure 4. Spiral search of a coherent neighboring point from the end point of a line segment.

the new line segment labeled by $l+1$, and repeat the spiral search above. Note that $(x(p), y(p))$ and $\theta^0(p)$ in the algorithm are directly retrieved from the spiral LUT. The first encountered p that meets the requirements gives the minimum distance to the end point of the line segment. Figure 5(a) and (b) presents, respectively, the results of pixel linkage for the defective and defect-free metal images in Figure 3(g) and (h), where individual line segments are displayed with different colors, and the line segments with small lengths (less than 20 pixels) are discarded from further consideration.

D. Connection of line segments

In the dark-pixel linkage process above, a discontinuous crack may result in a few disconnected line segments. They

should be connected to present the full length of the crack. As seen in the thinned images of the metal surfaces in Figure 3(g) and (h), the length of a crack is longer than the random dark objects in the background. Therefore, the line segment with the maximum length after the pixel linkage process is the most suspicious fragment of a crack. That is,

$$l^* = \arg \max_l \{ |L(l)| \}$$

where $|L(l)|$ denotes the length of line segment $L(l)$.

The line segments nearby are connected to $L(l^*)$ if they are within a given neighborhood distance and show consistent directions. Since a linked line segment is generally not a straight line, its direction is given by the mode, instead of the mean, of the angles for all pixels in the line segment. Let $L(l) = \{p_i^l = (x_i^l, y_i^l), i = 0, 1, 2, \dots, e\}$ and θ_i^l the rotation angle of pixel point p_i^l . The rotation angle is between 0° and 360° . The full range of angles is divided into 12 intervals (i.e. a resolution of 30°). The angular mode of line segment $L(l)$ is thus given by

$$\theta_{\text{mode}}(l) = \text{Mode}\{\text{Int}[\theta_i^l / 30^\circ + 1], \forall i = 0, 1, 2, \dots, e\}$$

A small connection window of size $\Delta h \times \Delta w$ with direction angle $\theta_{\text{mode}}(l^*)$ is placed at either end of the main line segment $L(l^*)$. The connection window is currently given by 20×10 pixels in this study. Figure 6 depicts the placement of the connection window $\Delta h \times \Delta w$ at the two end points of the line segment $L(l^*)$. If there is any line segment $L(k)$ within the connection window and the angular difference $\Delta\theta_{\text{mode}}$ with

$$\Delta\theta_{\text{mode}} = |\theta_{\text{mode}}(l^*) - \theta_{\text{mode}}(k)|$$

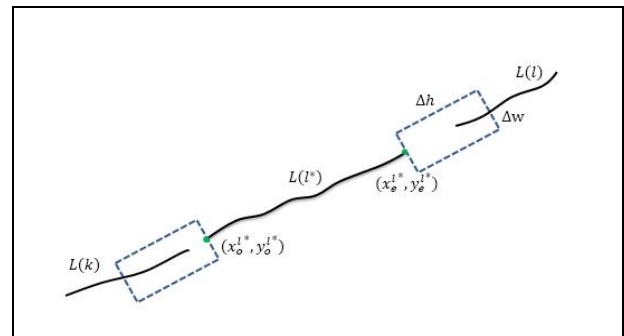


Figure 6. Sketch of the connection of line segments and the placement of the connection window $\Delta h \times \Delta w$.

is within the angular resolution, i.e. 30° , line segment $L(k)$ is then connected to $L(l^*)$. The maximum line segment $L(l^*)$ is then expanded, and the connection process is repeated until no connection of line segments is possible. If the final length of the connected line segment is larger than a length threshold, a crack is detected. Otherwise, the surface is defect-free. Figure 5(c) is the final connected line segments of a crack from Figure 5(a). Figure 5(e) illustrates the detection result by superimposing the connected line segments of the crack on the original metal surface in Figure 1(a). Figure 5(d) shows that no crack is detected from the defect-free metal surface in Figure 1(b). Algorithm III presents the procedure of line-segment connection.

III. EXPERIMENTAL RESULTS

If you are using *Word*, use either the Microsoft Equation Editor or the *MathType* add-on (<http://www.mathtype.com>) for equations in your paper (Insert | Object | Create New | Microsoft Equation or MathType Equation). “Float over text” should *not* be selected.

In this section, we present the performance of the proposed crack detection method. All algorithms were encoded in the C# programming language and implemented on a personal computer equipped with an Intel i7-3770 3.4GHz processor. The images used in the experiment were 320×240 pixels wide with 8-bit gray levels. The computation time from the binary thresholding to the final line-segment connection for a test image is only 0.168 seconds on average. In the experiment, the parameter values were fixed for all test images. The maximum spiral radius r_{\max} is set to 4 pixels in the pixel-linkage process, and the connection window $\Delta h \times \Delta w$ is set to 20×10 pixels in the line-segment connection process.

There are a total of 120 test images used in the experiment, of which 48 are defect-free and 72 are defective. The defective surfaces contain cracks of varying lengths and directions at different positions in the images. Figure 7(a1)-(a4) shows 4 images of the 72 defective surfaces, and Figure 8(a1)-(a4) demonstrates 4 of the 48 defect-free surfaces used in the experiment. Note that the cracks are different in length and locations, and are irregular and discontinuous. The defect-free samples present highly random patterns in the surfaces. They result in random black, short, and irregular line segments in the thinned images. Figure 7(b1)-(b4) presents the detection results by superimposing the detected cracks on the original textured images for the 4 defective test samples in Figure 7(a1)-(a4), respectively. Figure 8(b1)-(b4) are the detection results of the defect-free samples in Figure 8(a1)-(a4), where the resulting black images indicate no pixels of cracks are detected. With the given parameter settings, the proposed method claims all 48 defect-free test images contain no cracks, and all 72 defective test images contain cracks. The recognition rate of the proposed method is 100% based on the 120 test samples used in the experiment.

The proposed method can also be applied to crack detection in non-textured concrete surfaces. Figure 9(a1) and (b1) shows two concrete surface images that contain thin cracks on the surfaces. Since cracks show high-contrast intensities from their background surfaces, the binary thresholding value is set at $0.9 \cdot T$, where T is the threshold value obtained from the Otsu's method. All the remaining parameter settings are the same as those used for randomly textured metal surfaces in the previous experiment. Figure 9(a2) and (b2) presents the binary thresholding results, and Figure 9(a3) and (b3) shows the thinned images. Figure 9(a4) and (b4) illustrates the final results by superimposing the detected cracks on the concrete surfaces. The shapes and locations of the cracks are well detected.

IV. CONCLUSIONS

In this paper, we have proposed a machine vision scheme for crack detection in randomly textured metal surfaces. It uses the binary thresholding and morphological thinning as the preprocessing to obtain single pixel thickness of dark

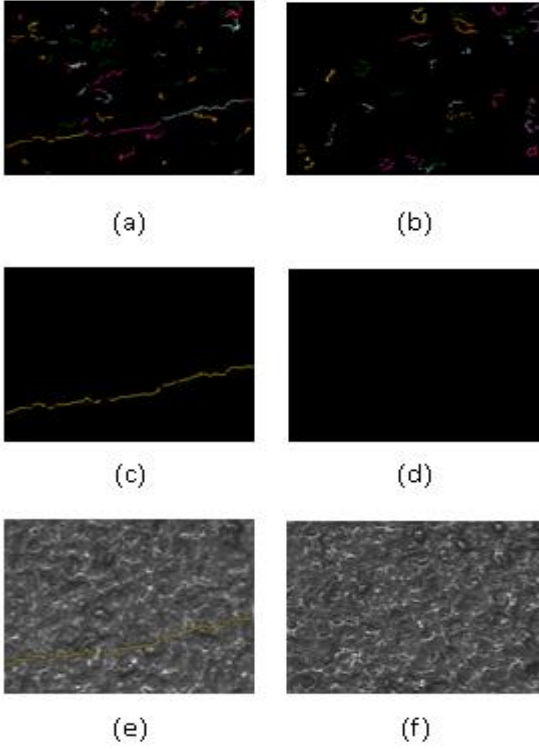


Figure 5. Pixel linkage and line-segment connection: (a), (b) respective pixel-linkage results of Figure 3(g) and (h); (c), (d) respective line-segment connection results of (a) and (b); (e), (f) superimposing the detected line segments of cracks on the original images.

Algorithm III: Connection of line segments

Let $l^* = \arg \max_l \{ |L(l)| \}$. (x_0^l, y_0^l) and (x_e^l, y_e^l) are the two end points of $L(l^*)$.

Initially, $\theta_{\text{mode}}^o(l^*) = \theta_{\text{mode}}^e(l^*) = \theta_{\text{mode}}(l^*)$

1. Extend the window $\Delta h \times \Delta w$ with orientation $\theta_{\text{mode}}^o(l^*)$ from end point (x_0^l, y_0^l) .

If $\exists L(k) \in \Delta h \times \Delta w$,

and $\Delta \theta_{\text{mode}} = |\theta_{\text{mode}}^o(l^*) - \theta_{\text{mode}}(k)| \leq 1$ (i.e. 30°),

then $L(l^*) \leftarrow L(l^*) \cup L(k)$

$(x_0^l, y_0^l) = (x_0^k, y_0^k)$ and $\theta_{\text{mode}}^o(l^*) = \theta_{\text{mode}}(k)$

2. Extend the window $\Delta h \times \Delta w$ with orientation $\theta_{\text{mode}}^e(l^*)$ from end point (x_e^l, y_e^l) .

If $\exists L(k) \in \Delta h \times \Delta w$,

and $\Delta \theta_{\text{mode}} = |\theta_{\text{mode}}^e(l^*) - \theta_{\text{mode}}(k)| \leq 1$ (i.e. 30°),

then $L(l^*) \leftarrow L(l^*) \cup L(k)$

$(x_e^l, y_e^l) = (x_e^k, y_e^k)$ and $\theta_{\text{mode}}^e(l^*) = \theta_{\text{mode}}(k)$

Repeat the steps above until no connection is possible.

objects. The spiral search is then performed on the thinned image to link object pixels into line segments. The linkage of neighboring pixel points is based on the distance and direction angle of the object points. The distance and angle of the neighboring points can be retrieved from the spiral look-up-table. The first encountered object point gives the closest neighboring point. No computation and sorting of distances are required, and thus the spiral search for object point linkage is very efficient. The individual line segments are further connected based on their neighboring distance and the mode of direction angles to find the complete shape of the crack.

Experimental results have revealed that the proposed method can detect thin cracks of varying lengths and directions in randomly textured metal surfaces. The random dark objects from the textured background do not affect the detection of true cracks. No false alarms are generated based on the test images used in the experiment. It is also demonstrated that the proposed method can be used to detect cracks in non-textured images, such as concrete surfaces.

In the proposed crack detection algorithms, the size of the search spiral r_{\max} and the size of the connection window $\Delta h \times \Delta w$ are empirically determined by analyzing the possible disconnected distance of a crack in the metal surface. It may be required to adjust the parameter values for different material surfaces. It is worth further investigation to set the parameter values automatically and adaptively.

REFERENCES

- [1] Aarthi, T., M. Karthi, M. Abinesh, Detection and analysis of surface defects in metals using wavelet transform, International Journal of scientific and Research Publications 3, 1-6.
- [2] Choi, D.-C., Y.-J. Jeon, J. P. Yun, S. W. Yun, S. W. Kim (2012), An algorithm for detecting seam cracks in steel plates, World Academy of Science, Engineering and Technology 6, 1478-1481.
- [3] Choi, D.-C., Y.-J. Jeon, S. J. Lee, J. P. Yun, S. W. Kim (2014), Algorithm for detecting seam cracks in steel plates using a Gabor filter combination method, Applied Optics 22, 4856-4872.
- [4] Choudhary, G. K., S. Dey (2012), Crack detection in concrete surfaces using image processing, fuzzy logic, and neural networks," In: IEEE International Conference on Advanced Computational Intelligence, pp. 404-411.
- [5] Fujita, Y., Y. Hamamoto (2011), A robust automatic crack detection method from noisy concrete surfaces, Machine Vision and Applications 22, 245-254.
- [6] Gonzalez, R. C., R. E. Woods (2008), Digital Image Processing, Person Prentice Hall, Upper Saddle River, NJ.
- [7] Hu, H., G. Gu, J. Zhou (2010), HTF: a novel feature for general crack detection, In: International Conference on Image Processing, pp. 1633-1636.
- [8] Landstrom, A., M. J. Thurley (2012), Morphology-based crack detection for steel slabs, IEEE Journal of Selected Topics in Signal Processing 6, 866-875.
- [9] Malekian, V., R. Amirfattahi, M. Rezaeian, A. Aghaei, P. Rahimi (2012), Automatic detection and localization of surface cracks in continuously cast hot steel slabs using digital image analysis techniques, International Journal of ISSI 9, 30-40.
- [10] Neogi, N., D. K. Mohanta, P. Dutta (2014), Review of vision-based steel surface inspection systems, EURASIP Journal on Image and Video Processing, 2014:50.
- [11] Nguyen, H.-N., T.-Y. Kam, P.-Y. Cheng (2014), An automatic approach for accurate edge detection of concrete crack utilizing 2D geometric features of crack, Journal of Signal Processing Systems 77, 221-240.
- [12] Otsu, N., (1979), A threshold selection method for gray-level histograms, IEEE Trans. Systems, Man and Cybernetics SMC-9, 62-66.
- [13] Prasanna, P., K. Dana, N. Gucunski, B. Basily (2012), Computer vision based crack detection and analysis, In: Proceedings of the SPIE

8345, Sensors and Smart Structures Technologies for Civil, Mechanical and Aerospace Systems.

- [14] Senthikumar, M., V. Palanisamy, J. Jaya (2014), Metal surface defect detection using iterative thresholding technique, In: International Conference on Current Trends in Engineering and Technology, pp. 561-564.
- [15] Yamaguchi, T., S. Hashimoto (2010), Fast crack detection method for large-size concrete surface images using percolation-based image processing, Machine Vision and Applications 21, 797-809.
- [16] Yamaguchi, T., S. Nakamura, R. Saegusa, S. Hashimoto (2008), Image-based crack detection for real concrete surfaces, Transactions on Electrical and Electronic Engineering 3, 128-135.
- [17] Zhang, W., Z. Zhang, D. Qi, Y. Liu (2014), Automatic crack detection and classification method for subway tunnel safety monitoring, Sensors 14, 19307-19328.

Du-Ming Tsai received the B.S. degree in Industrial Engineering from the Tunghai University, Taiwan in 1981, and the M.S. and Ph.D. degrees in Industrial Engineering from Iowa State University, Ames, Iowa in 1984 and 1987, respectively. From 1988 to 1990, he was a Principal Engineer of Digital Equipment Corporation, Taiwan branch, where his work focused on process and automation research and development. Currently he is a Professor of Industrial Engineering and Management at the Yuan-Ze University, Taiwan. His research interests include automated visual inspection, object recognition and texture analysis.

Yi-Jun Xie received the B.S. degree in Industrial Engineering and management from the Yuan-Ze University, Taiwan in 2015 and the M.S. degrees in Industrial Engineering and management from the Yuan-Ze University in 2016.