# CHORD: Cascaded and *A Contrario* Method for Hole Crack Detection

Jierong Cheng[1], Wei Xiong[1], Yue Wang[1], Shue Ching Chia[1], Wenyu Chen[1], Jia Du[1], Ying Gu[1], and Victor Ter Shen Kow[2]
[1] Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore
[2] National University of Singapore, Singapore
Email: chengjr@i2r.a-star.edu.sg

***Abstract*—We propose a cascaded and *a contrario* hole crack detection (CHORD) method for defect inspection in digital images of turbine blade surfaces. This is the first time an automatic image processing-based method is proposed for such a task. It consists of two major steps: first, the appearance of holes is approximated by ellipses and cascaded pose regression is used to estimate the position and orientation of the holes; Second, we define hole cracks as geometrical structures and *a contrario* method is used to assess the meaningfulness of each crack. The model-based CHORD method fully considers the features of hole cracks including the characteristics of brightness, length, and orientation, and therefore can accurately detect cracks in the images. The threshold on the strength of cracks is determined automatically and the computational time is about five seconds for each image.**

## I. Introduction

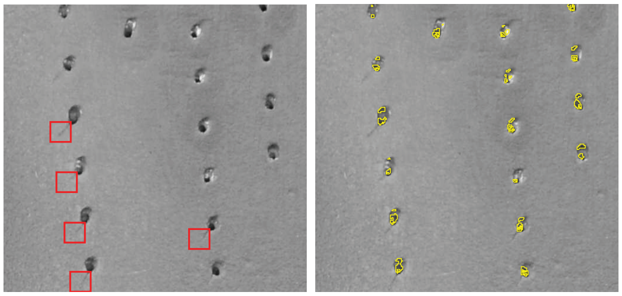
Surface defect inspection is of primary importance for engineering part quality inspection. Surface defects affect not only the appearances of parts, but also their functionality, efficiency and stability. Crack is one of the most common types of defects among all kinds of surface defects. Despite of their extremely small sizes, hole cracks (cracks which extend from holes) can develop into critical conditions in machine. Therefore, hole crack detection is an important task in mechanical part surface inspection. Up to present, this task is carried out by human visual inspection by experienced inspectors, which is time consuming, subjective, and lacks of quantification. Hence, automatic inspection of the surfaces by using image processing techniques is highly desired. Crack detection is a very challenging problem because of the low contrast between cracks and the background, the intensity inhomogeneity along the cracks, and the presence of other structures (e.g. dents, scratches) with similar intensity to the cracks. Automatic method for hole crack detection in turbine blade images has not been reported in the existing literature and there is no public database for such images.

In this work, cascaded and *a contrario* hole crack detection (CHORD) method is proposed for defect inspection in digital images of turbine blade surfaces. The process consists of two major tasks: hole detection and crack detection. First, we use extended-minima transform and cascaded pose regression [1] to localize the holes. Then, we define hole cracks as geometrical structures in discrete images and each candidate crack is assigned with a value of *strength* computed from image intensities. Finally, we rely an *a contrario* methodology [2] to decide automatically which cracks are to be detected in an image, depending on their *meaningfulness* under some null hypothesis $\mathcal{H}_0$.

## II. Related work

In the past decade, various computer vision and image processing methods have been used for the crack detection on different surfaces such as infrastructure and manufacturing parts. These methods can be roughly groups into global based and local grid based methods [3]. Global based methods were applied on the whole image to search for cracks. For example, a two-step approach was proposed in [4] for the detection of surface defects: global estimation by Phase Only Transform (PHOT) and local refinement based on the distributions of pixel intensities. Local grid based methods partition the image into small grid cells, and find the potential crack cells. Classification methods are often used to classify the grid cells into crack or non-crack cells. Efficient features were proposed to describe the characteristics of cracks, e.g. Hough transform based feature [3] and distance histogram based shape descriptor [5]. In [6], possible crack regions are distinguished from background image by using image processing techniques including Guassian filtering, thresholding, and morphological operations. Then the existence of cracks are identified automatically by backpropagation neural network. Two kinds of approaches have been implemented in [7]: the image approach which classifies an image as a whole, and the object approach which classifies each component or object in an image into cracks and noise. However, the above mentioned methods are all designed for crack detection, not for hole crack detection. The contrast around the crack is insignificant in comparison to the contrast of holes, which brings difficulties to global based and local based methods (see Fig. 1 for illustration). A fastener hole crack detection method was proposed in [8], which works by correlating digital images of the structure surfaces in unloaded and loaded states. This method is unable to handle the hole crack detection in normal non-load holes on part surfaces, for instance air holes in blade which do not have loaded states to compare with.

Desolneux et al. proposed an *a contrario* method for detecting geometric structures in an image without any prior information [2]. The main idea is that an observed geometric



(a) (b)

Fig. 1. (a) Original image. Hole cracks are highlighted by red squares. (b) Defect detection results by applying the method introduced in [4]. The 'defect areas' found (outlined in yellow contours) are all on the holes while non of the cracks are detected.

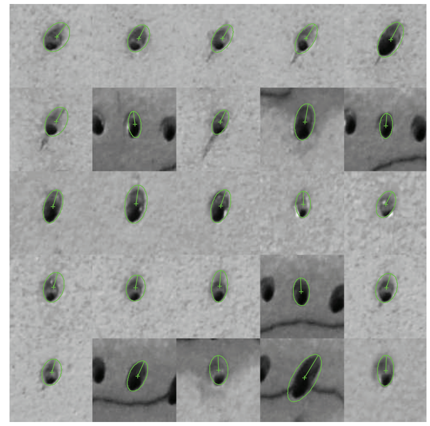


Fig. 2. Hole detection results by using cascaded pose regression.

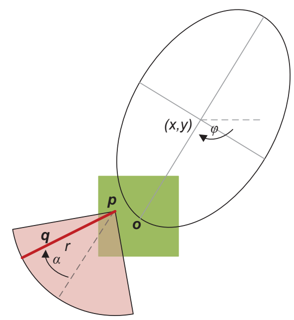



Fig. 3. Representation of a hole crack $c_{\boldsymbol{p}}(r, \alpha)$ (in red line).

event is "meaningful" if the expectation of its occurrences would be very small in a random image. This methods has been used by Xia et al. for validation during junction detection in natural images [9]. Although the appearance of hole crack is similar to Y-junctions, there are difficulties when applying junction detectors directly on hole crack detection. First, the cracks are string-like structures which won't be detected by model-based junction detectors making use of the direction of the pixel gradient [10], [9]. Second, removing the false detections generated by signal-based and boundary-based junction detectors is a very challenging task. We propose to use extended-minimas transform and cascaded pose regression to localize the holes as the first step. Thereby, crack detection is carried out in each image patch of hole instead of in the whole image, which not only reduces false detections but also provides a good initial guess of the crack position.

## III. Methodology

The details of the CHORD method is presented in this section.

### A. Preprocessing

First of all, sub-images containing the blade region are cropped from the original scanned images manually. As air holes on the blade appear to be elliptical dark spots, they are located by applying extended-minima transform on the image, which gives the regional minima of the H-minima transform. Centered at the centroid of each regional minima, a $100 \times 100$ image patch $I$ which contains a hole is cropped from the image.

### B. Cascaded pose regression

In this step, the pose of the holes is determined by using cascaded pose regression (CPR) [1]. Each hole $\theta$ is approximated by an ellipse at location $(x, y)$ with orientation $\varphi$, major axis $a$ and minor axis $b$. A training sample consists of an image $I_i$ and a true pose $\hat{\theta}_i$, $i = 1, 2, ..., N$.

A cascaded regressor $R = (R^1, R^2, ..., R^T)$ consists of $T$ weak regressors. Given an image $I$ and an initial pose $\theta^0$, each regressor generates a pose increment vector $\delta\theta$ to update the previous pose and the output of regressors $R^t$ depends on image $I$ and the previous boundary $\theta^{t-1}$:

$$\begin{aligned} \theta^t &= \theta^{t-1} \circ \delta\theta^t, \\ \text{with } \delta\theta^t &= R^t(I, \theta^{t-1}), t = 1, 2, ..., T. \end{aligned} \tag{1}$$

Each regressor is trained to minimize the difference between the true pose and the new pose updated by the regressor, i.e.,

$$R^t = \arg\min_R \sum_{i=1}^{N} \|\hat{\theta}_i - (\theta_i^{t-1} + R(I_i, \theta_i^{t-1}))\|_2. \tag{2}$$

We use random ferns [11] as weak regressors in the cascade. Simple pose-indexed features [1] are used to learn each regress from a set of training data. For each pose $\theta = [x, y, \varphi, a, b]$, an associated $3 \times 3$ homography matrix $H_\theta$ can be defined to express location $p$ in homogeneous coordinates. Each control point feature $h_{p_1,p_2}$ is defined by two image locations $p_1$ and $p_2$ and pose-indexed features are evaluated by computing $h_{p_1,p_2}(\theta, I) = I(H_\theta p_1) - I(H_\theta p_2)$.

Examples of hole detection results by CPR are displayed in Fig. 2.

### C. A contrario hole crack detection

After the hole location and pose are found, meaningful cracks are detected based on an *a contrario* method. In this step, we make two assumptions on the basis of the mechanism of hole cracks in blade: (1) there is no more than one crack extending from each hole, and (2) the crack starts from the lower vertex of a hole and extends approximately along the major axis of the hole.

*1) Crack representation:* A crack is represented by a discrete line segment $c : \{\boldsymbol{p}, r, \alpha\}$, characterized by its starting point $\boldsymbol{p}(x, y)$, its length $r \in \mathbb{N}$ and its direction $\alpha \in [0, 2\pi)$. See Fig. 3 for illustration.

In order to be robust to shading changes on the blade, we locally normalize the intensity image by dividing it by its median on a small neighborhood. That is, for $\boldsymbol{q}(x, y)$, we define a normalized intensity image

$$\hat{I}(\boldsymbol{q}) = \frac{I(\boldsymbol{q})}{\mathrm{median}(I(\mathcal{N}(\boldsymbol{q})))}. \tag{3}$$

where $\mathcal{N}(\boldsymbol{q})$ is a square neighborhood of size $11 \times 11$ around $\boldsymbol{q}$. A null hypothesis $\mathcal{H}_0$ is defined as

- $\forall \boldsymbol{q} \in \Omega$, $\hat{I}(\boldsymbol{q})$ follows a distribution which can be estimated from its histogram, and
- $\{\hat{I}(\boldsymbol{q})\}_{\boldsymbol{q} \in \Omega}$ is made of independent random variables.

The strength of a crack candidate is defined as the negative sum of normalized intensity of pixels on the crack

$$t\big(c_{\boldsymbol{p}}(r, \alpha)\big) = -\sum_{\boldsymbol{q} \in c_{\boldsymbol{p}}(r, \alpha)} \hat{I}(\boldsymbol{q}) \tag{4}$$

The strength value is supposed to be high for real cracks.

Let $\mathcal{C}$ be the set of all possible cracks in the discrete image patch $I$. For $\epsilon > 0$, a crack $c$ is said to be $\epsilon$-meaningful if its number of false alarm (NFA) under the hypothesis $\mathcal{H}_0$ satisfies

$$\mathrm{NFA}(c) \equiv \#\mathcal{C} \cdot F_c(t(c)) \leq \epsilon \tag{5}$$

where $F_c(t)$ is the probability that a random variable $\boldsymbol{t}$ is larger than a given threshold $t$. The value $\epsilon$ corresponds to an expected number of false detections under the hypothesis $\mathcal{H}_0$. The smaller the quantity NFA($c$) is, the more meaningful the crack $c$.

*2) Crack detection:* To search for crack candidates, we first set the starting point $\boldsymbol{p}$ in the $w \times w$ window $\mathcal{N}^w(\boldsymbol{o})$ centered at the lower vertex $\boldsymbol{o}$ on the major axis of the ellipse representing a hole.

The length of crack $r$ ranges discretely from $r_{\min}$ to $r_{\max}$: $r \in [r_{\min}, r_{\max}]$. The crack direction $\alpha$ is within a $\Delta\alpha$ range from the ellipse orientation: $\alpha \in [\varphi - \frac{\Delta\alpha}{2}, \varphi + \frac{\Delta\alpha}{2}]$. The number of discrete direction at a given crack length $r$ is set to be $K(r) = \Delta\alpha \cdot r$. Therefore, the number of all possible cracks in an image patch $I$ is

$$\#\mathcal{C} = \#\Big\{\boldsymbol{p} \in \mathcal{N}_w(\boldsymbol{o})\Big\} \sum_{r=r_{\min}}^{r_{\max}} K(r) = w^2 \sum_{r=r_{\min}}^{r_{\max}} \Delta\alpha \cdot r \tag{6}$$

To compute the distribution $\mu$ of the strength of a crack, empirical distribution of $\hat{I}(\boldsymbol{q})$ is estimated from its histogram. Then the law of the strength of a crack $c_{\boldsymbol{p}}(r, \alpha)$, i.e. sum of the $r$ pixel contributions by our definition, is obtained by convolving $r$ copies of the distribution $\mu$:

$$F_c(t, r) \equiv \mathbb{P}_{\mathcal{H}_0}\big[\boldsymbol{t}(c) \geq t\big] = \int_t^{\infty} d \left( \mathop{*}_{j=1}^{r} \mu \right) \tag{7}$$

where $*$ denotes the convolution operator. Finally, a crack is detected as $\epsilon$-meaningful in image patch $I$ if its strength $t(c)$ is larger than the threshold

$$\tilde{t}(r, \epsilon) \equiv \min\Big\{t; F_c(t, r) \leq \frac{\epsilon}{\#\mathcal{C}}\Big\}. \tag{8}$$

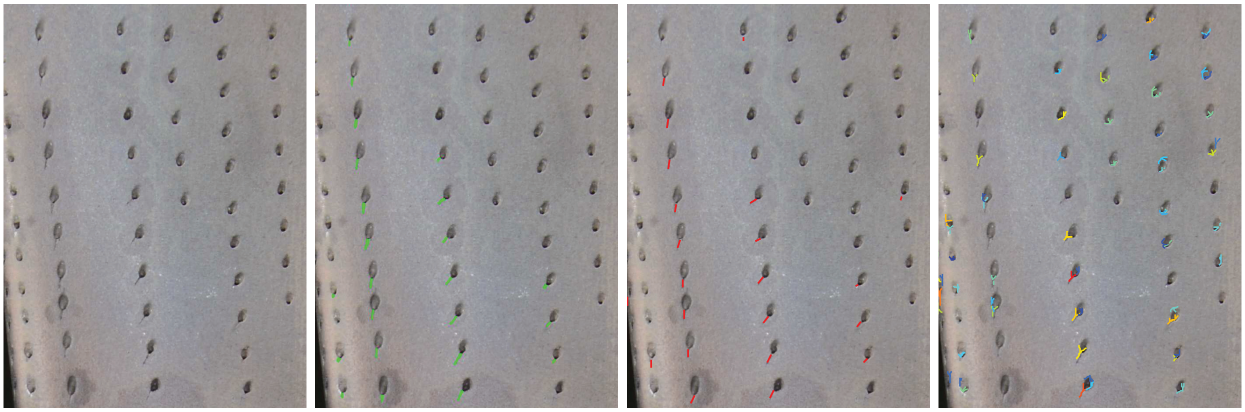
The overall *a contrario* crack detection procedure is described in Algorithm 1.

**Algorithm 1** *A contrario* crack detection

**Input:** An image patch $I$, ellipse orientation $\varphi$ and lower vertex $\boldsymbol{o}$, and parameters $\epsilon$, $\Delta\alpha$, $w$, $r_{\min}$, $r_{\max}$
**Output:** The maximal $\epsilon$-meaningful crack $c^*$ in $I$ (if any).
Compute $\hat{I}$ at each pixel using Eqn. (3);
Compute $\#\mathcal{C}$ by using Eqn. (6);
For each value of $r$ between $r_{\min}$ and $r_{\max}$, compute $F_c(t, r)$ by using Eqn. (7);
Let C be an empty set of crack candidates;
**for all** pixels $\boldsymbol{p} \in \mathcal{N}^w(\boldsymbol{o})$ **do**
  **for** $r = r_{\min}$ to $r_{\max}$ **do**
    **for** $\alpha = \varphi - \frac{\Delta\alpha}{2}$ to $\varphi + \frac{\Delta\alpha}{2}$ **do**
      Add $c_{\boldsymbol{p}}(r, \alpha)$ to C;
      Compute the crack strength $t\big(c_{\boldsymbol{p}}(r, \alpha)\big)$ by using Eqn. (4);
      Compute $\log \mathrm{NFA}(c)$ by using Eqn. (5);
    **end for**
  **end for**
**end for**
Find the most meaningful crack by computing $c^* = \arg\min_{c \in \mathtt{C}} \log \mathrm{NFA}(c)$;
**if** $\log \mathrm{NFA}(c^*) \leq \log \epsilon$ **then**
  Output the maximal $\epsilon$-meaningful crack $c^*$.
**end if**

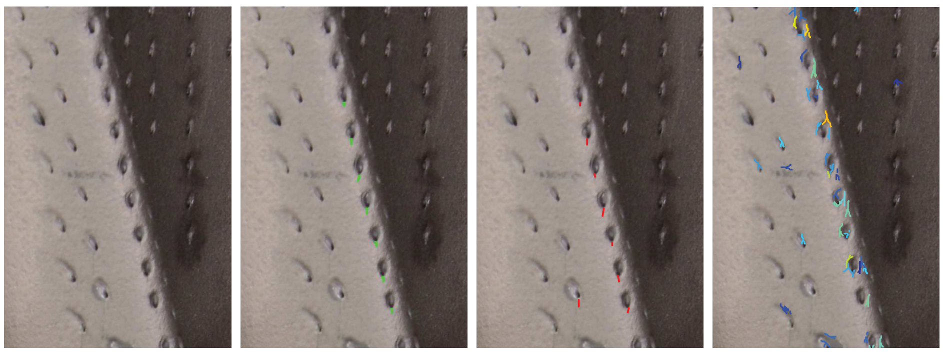
## IV. Experimental results

In this section, the experimental results of cascaded and *a contrario* hole crack detection (CHORD) method is presented. Our method is implemented in Matlab and C. All test were run on a PC with Intel Core 2 Quad CPU at 2.83GHz and 8GB RAM. The experiment is carried out on 22 digital images of turbine blade, taken from two different imaging angles (60° and 140°) to capture the hole cracks on the front and the side of the blades. The images are taken by a Nikon D800 DSLR with 24mm-120mm f/4G VR Lens. Besides the constant environment light, the illumination is mainly contributed by direct light source which is an array of point light model, parallel transmission. The original image size is 4912×7360 pixels and the image resolution is 31.25 $\mu$m per pixel. The



(a) (b) (c) (d)

Fig. 4. (a) Original image of front view. (b) Cracks manually drawn as ground truth. (c) Cracks detected by CHORD. (d) Cracks detected by ACJ.



(a) (b) (c) (d)

Fig. 5. (a) Original image of side view. (b) Cracks manually drawn as ground truth. (c) Cracks detected by CHORD. (d) Cracks detected by ACJ.

width of the cracks present in the images are between 4 to 6 pixels (=120$\sim$180 $\mu$m).

100 image patches containing holes are randomly selected as training data (these images are not included during the testing of our method), on which ellipses are manually drawn around holes. The parameters are set as follows: $r_{\min} = 5$, $r_{\max} = 20$, $\Delta\alpha = \pi/2$, $w = 5$, and $\epsilon = 1$.

There are a total of 1726 holes in the testing images, of which 1707 (98.9%) are successfully detected. 374 hole cracks are existent on these holes, by common consent among five human inspectors. The accuracy of hole crack detection in terms of Precision, Recall, and F-score is shown in Table I. In Figs. 4 and 5, we can see that CHORD can detect most of the hole crack correctly in spite of the low contrast on these tiny thin structures and the uneven background intensity. The Y-junctions detected by *a contrario* junction detector (ACJ) [9] are also shown. Although ACJ can find some obvious hole cracks as displayed in Fig. 4(d), it fails when the contrast on cracks is very low. Furthermore, ACJ generates many false detections round the holes, whereas the cracks are not detected as one of the three branches of a Y-junction. The computational time of CHORD is around five seconds per image, which is only 8% of ACJ's.

TABLE I
ACCURACY AND COMPUTATIONAL TIME (FOR ONE IMAGE OF 1903×1411 PIXELS) OF HOLE CRACK DETECTION.

| Method | Precision | Recall | F-score | Time (s) |
|---|---|---|---|---|
| CHORD | 0.800 | 0.770 | 0.785 | 5 |
| ACJ | 0.138 | 0.706 | 0.231 | 64 |

## V. Conclusion

We propose a cascaded and *a contrario* hole crack detection (CHORD) method in this paper. Extended-minima transform and cascaded pose regression are used to localize the holes as the first step, providing a good initial guess of the crack position. Hole cracks are defined as geometrical structures and detected based on their meaningfulness evaluated by an *a contrario* method. Specifically, the threshold on the crack strength is determined automatically by requiring that the number of false alarms for each hole is less than 1. Therefore, CHORD has the advantages of low false detection rate and low computational time. The experimental results tested on digital images of turbine blade surfaces show that the proposed method performs well in terms of both hole and hole crack detection accuracy.

## References


[1] P. Dollár, P. Welinder, and P. Perona, "Cascaded pose regression," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1078–1085.

[2] A. Desolneux, L. Moisan, and J.-M. Morel, "Meaningful alignments." *International Journal of Computer Vision*, vol. 40, no. 1, pp. 7–23, 2000.

[3] H. Hu, Q. Gu, and J. Zhou, "HTF: a novel feature for general crack detection," in *Proceedings of IEEE Conference on Image Processing*. IEEE, 2010, pp. 1633–1636.

[4] J. Choi and C. Kim, "Unsupervised detection of surface defects: A two-step approach," in *Proceedings of IEEE Conference on Image Processing*. IEEE, 2012, pp. 1037–1040.

[5] W. Zhang, Z. Zhang, D. Qi, , and Y. Liu, "Automatic crack detection and classification method for subway tunnel safety monitoring," *Sensors*, vol. 14, no. 10, pp. 19 307–19 328, 2014.

[6] H.-G. Moon and J.-H. Kim, "Inteligent crack detecting algorithm on the concrete crack image using neural network," in *Proceedings of the 28th ISARC*, 2011, pp. 1461–1467.

[7] G. K. Choudhary and S. Dey, "Crack detection in concrete surfaces using image processing, fuzzy logic, and neural networks," in *Proceedings of IEEE Fifth International Conference on Advanced Computational Intelligence*, 2012, pp. 404–411.

[8] T. Hutt and P. Cawley, "Feasibility of digital image correlation for detection of cracks at fastener holes," *NDT&E International*, vol. 42, pp. 141–149, 2009.

[9] G.-S. Xia, J. Delon, and Y. Gousseau, "Accurate junction detection and characterization in natural images," *International Journal of Computer Vision*, vol. 106, no. 1, pp. 31–56, 2014.

[10] E. D. Sinzinger, "A model-based approach to junction detection using radial energy," *Pattern Recogn.*, vol. 41, no. 2, pp. 494–505, Feb. 2008.

[11] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua, "Fast keypoint recognition using random ferns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 448–461, 2010.