IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 25, 2019, accepted October 6, 2019, date of publication October 22, 2019, date of current version November 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2948876

Automatic Twining Direction Recognition of Lockwire on Engine

JUNHUA SUN[1], (Member, IEEE), LEWEI HUANG[1], AND JIE ZHANG[2]
[1]School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing 100191, China
[2]School of Computer and Information Engineering, Beijing Technology and Business University, Beijing 100048, China

Corresponding author: Jie Zhang (zhangjie@btbu.edu.cn)

ABSTRACT Lockwires are often used as mechanical locking components for bolts on an engine. Inspecting its twining direction relative to the bolt is vital for ensuring an engine runs normally. However, the dominant manual inspection method is inefficient and limited by the professional skills of the inspector. In this paper, we propose a method that can be used to automatically determine the twining direction of a lockwire. First, a coarse-to-fine scheme for localizing the bolt is designed. Then, rough segmentation regions of the lockwire are acquired by a novel elongated shape descriptor. The entire area containing the lockwire can be obtained by connecting images from segmentation regions. An innovative skeleton tree and texture-based selection strategy are used to extract the lockwire centerline. An efficient vector fitting approach is proposed for computing the extension direction of the centerline. As a result, the twining direction can be identified via the extension direction and the bolt's position. The experimental results show that our method is very effective in determining the twining direction of a lockwire, with an accuracy reaching 95%. In addition, the results also show that our method can deal with low-quality images of lockwires at different angles.

INDEX TERMS Lockwire, twining direction, automatic recognition, engine.

I. INTRODUCTION

Preventing threaded fasteners from loosening or falling out due to vibration and other forces [1] is extremely important as these events create safety hazards in mechanical systems, especially engines. To this end, the lockwire [2]–[4] shown in Fig. 1 is commonly used in the aviation and automotive industries, as a secondary locking mechanism [5], [6] for the bolt on an engine. Generally, the lockwire is assembled in bolted joints where the use of locknuts is inconvenient, and its twining direction should be parallel to the direction in which the bolt is tightened, i.e., clockwise. Instead, the wrong twining direction will speed up the bolt-loosening [7], which could cause lethal security hazards. Therefore, it is essential to carefully inspect the lockwire twining direction.

Currently, the primary approach determining the twining direction of a lockwire is manual inspection. This has several disadvantages. Specifically, inspectors are required to concentrate on this serious task for a long time. Thus, it could be difficult to ensure a reliable inspection. Furthermore, this inefficient approach cannot satisfy great demand in modern industrial production.

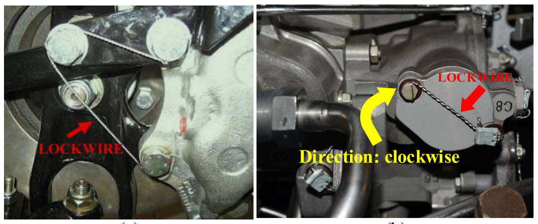

FIGURE 1. Examples of lockwire with correct twining direction (a) on a race car and (b) on an aircraft engine. Note that the lockwire can prevent the bolt from loosening only when it is assembled clockwise.

On the other hand, very few investigations on lockwires were conducted and their focus is on other aspects rather than the identification of the lockwire twining direction. A comparative analysis of lockwires with other locking devices was performed previously in [8], [9], and the impact of the lockwire material on locking strength was studied in [10]. Although they provide some instructions for preventing a bolt from loosening under vibration, the risks caused by the incorrect installation are still unavoidable.

The associate editor coordinating the review of this manuscript and approving it for publication was Dong Wang.

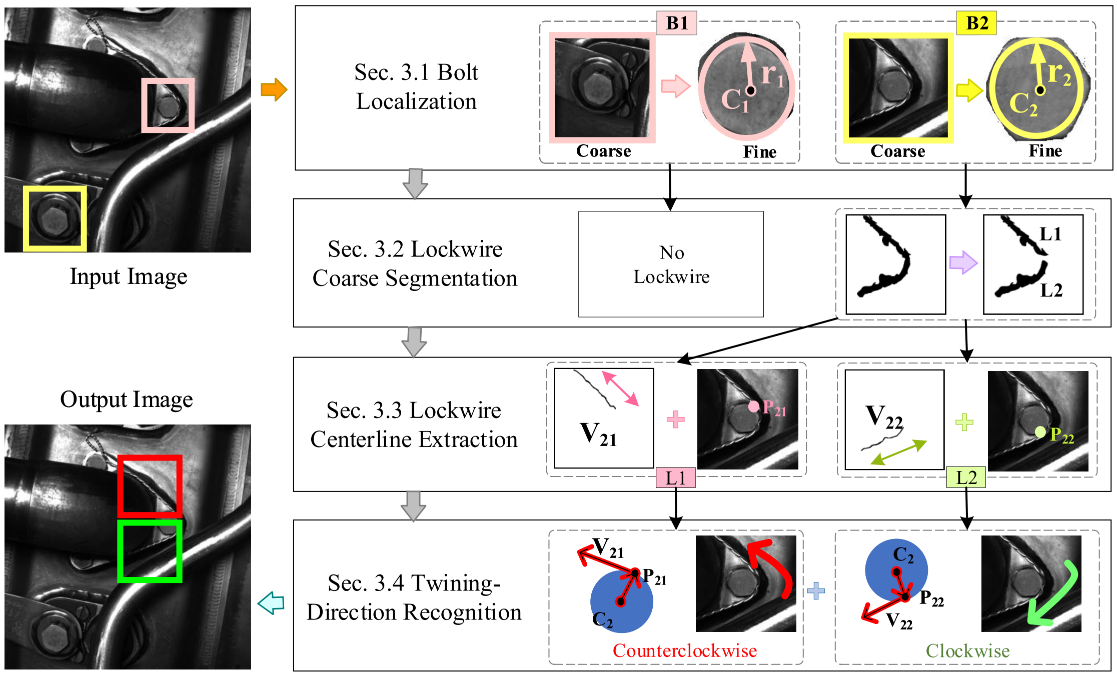

FIGURE 2. **Overview of our LTDR method. The first stage is used for bolt localization (B1 and B2, where c_* is the center and r_* is the radius). The coarse elongated regions of the lockwire (L1 and L2) are determined in the second stage. Based on these regions, the lockwire centerlines (V_{21} and V_{22}, where P_{21*} and P_{22} are the intersection of the centerline and the bolt) can be extracted in the third stage. Finally, the extension direction of the centerlines is computed. The twining direction is identified via the extension direction and the bolt's fine position.**

The two aforementioned issues motivated us to conduct a study on automatic Lockwire Twining Direction Recognition (LTDR). Unlike common recognition tasks in industrial production, such as crack identification [11] and pipeline recognition [12], LTDR is more challenging. Firstly, common recognition tasks only need to recognize one target. However, the lockwire twining direction is determined by a pair of lockwires and locked bolts, requiring accurately localization and their one-to-one correspondence. Secondly, the lockwire not only has a braided structure [13], but also an elongated irregular shape [14]. This special structure makes it intractable to completely segment the lockwire out, which causes great difficulty for lockwire identification. Lastly, the background is more complicated due to many types of components on the outside of the engine. Therefore, it is more difficult to accurately find the lockwire and match it to the corresponding bolt.

To accomplish such a challenging task, this paper proposes an automatic LTDR method based on machine learning and image processing. The bolt is initially localized using a coarse-to-fine scheme. The entire region containing the lockwire is subsequently acquired by locating different bright spots. Afterwards, the lockwire centerline in this region is extracted, and its extension direction is computed. Finally, the twining direction is identified from the extension direction and the bolt's position.

The rest of this paper is organized as follows. Our LTDR method is presented in Sec. 2, and each step is detailed in Sec. 3. Our dataset, experimental evaluation, and comprehensive analyses are presented in Sec. 4. Sec. 5 concludes the paper.

II. OVERVIEW

An overview of our LTDR method is shown in Fig. 2. The method includes bolt localization (Sec. 3.1), lockwire coarse segmentation (Sec. 3.2), lockwire centerline extraction (Sec. 3.3), and twining direction recognition (Sec. 3.4).

A. BOLT LOCALIZATION

We propose a coarse-to-fine scheme to precisely localize the bolt. Specifically, an Adaboost-based model is used for coarse localization. The fine position is determined using edge detection in the polar image.

B. LOCKWIRE COARSE SEGMENTATION

In this stage, the separated bright spots of the lockwire are obtained from the bolt's coarse localization using the maximally stable extremal regions (MSER) algorithm [15]. We propose a novel shape descriptor that uses these spots in continuous segmentation.

C. LOCKWIRE CENTERLINE EXTRACTION

A novel skeleton tree and texture-based selection strategy are proposed for extracting the lockwire centerline from the elongated segmentation regions determined in the previous stage.

D. TWINING DIRECTION RECOGNITION

The random sample consensus (RANSAC) algorithm [16] is an efficient vector fitting approach. We use RANSAC to compute the lockwire's extension direction. Finally, the twining direction is determined from the extension direction and the bolt's fine position.

III. METHODS

A. BOLT LOCALIZATION

Recognizing the twining direction requires localizing the bolt and lockwire simultaneously. Generally, it is reasonable to localize the bolt first because the lockwire can be determined from the bolt's position.

We propose a coarse-to-fine scheme for localizing the bolt in the image. Specifically, we designed a detection model based on the Adaboost algorithm [17] to obtain the bolt's coarse localization first. In this model, the decision tree is used as the weak classifier, and the HOG descriptor [18] is used as the feature descriptor. The Adaboost algorithm uses simple weak classifiers for detection, and we can choose from a variety of weak classifiers. Usually, Adaboost can get a robust classifier even with a small-scale dataset.

We approximate the bolt in the image as a circle and perform accurate edge detection on its polar image. Because the bolt remains circular even if it is deformed, the edge of the bolt can always remain in the vertical direction in its polar image. Compared with common strategies for circle detection like Hough transformation, the proposed method can detect bolts that are not round enough, thus it has better universality and robustness. Fig. 3(a) can be approximated as a vertical line in Fig. 3(b). For visual comparison, Fig. 3(d) and (e) show the transformation of a standard hexagonal bolt using two relevant circles.

The original image is transformed into a polar image using two steps. The first step requires transforming the original image from the image coordinate system to a Cartesian coordinate system, and subsequently to a Polar coordinate system. Fig. 4 shows the origin and axes in these three coordinate systems. Formally, let H and W be some number of rows and columns. Moreover, let (u, v), (x, y), and (m, n) denote the coordinates in the image, Cartesian, and polar coordinate systems, respectively. Then, (x, y) is defined as

$$x = u - W/2, \quad y = -v + H/2. \tag{1}$$

To eliminate the impact of the changes in image scales, the images should have the same size in the image and polar coordinate systems. Therefore, we defined an angle scaling factor $\Delta\theta = 2\pi/H$ and a length scaling factor $\Delta\theta = 1$. Then, (x, y) can be written in polar coordinates as follows

$$m = \rho/\Delta\rho, \quad n = \theta/\Delta\theta, \tag{2}$$

where $\rho = \sqrt{(x^2 + y^2)}$ is the length of the vector consisting of (x, y) and the origin, and $\theta = arctan((y)/(x))$ is the angle between this vector and the x-axis.

Afterwards, the Sobel operator is used to detect the edge along the vertical direction in the polar image. The edge of the bolt in the original image can be determined by transforming the result back to the original image coordinate system. Because the bolt's edge is approximated as a circle, its fine position can be represented in terms of its center and radius.

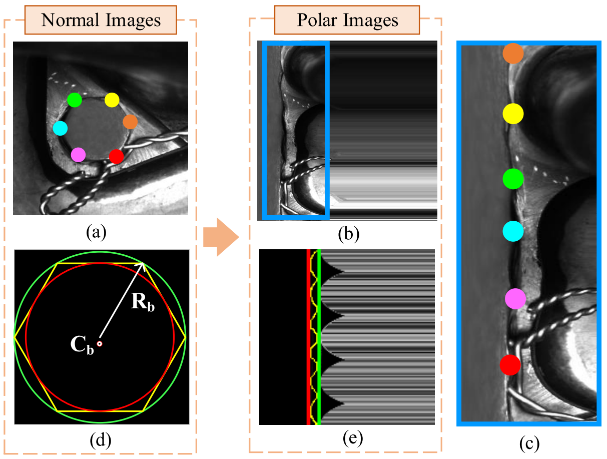

FIGURE 3. A hexagonal bolt in different coordinate systems. (a) Normal image showing hexagonal bolt. (b) The polar image of (a). (c) A partially magnified image of (b). (d) An image showing the outline of a hexagonal bolt and standard circles. The yellow hexagon shows the edge of the bolt, and the red and green circles show its inscribed and circumscribed circles. (e) Polar image of (d). The six colored points in (a) and (c) represent the six corners of the hexagonal bolt.

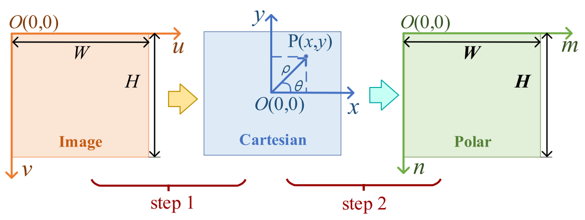

FIGURE 4. Defining the origin and axis in the three coordinate systems.

B. LOCKWIRE COARSE SEGMENTATION

Due to the constrained spatial position between the lockwire and bolt, the region containing the lockwire can be identified in the image produced from the Adaboost-based detection model. However, as shown in Fig. 3(a-2), directly segmenting the lockwire is intractable because it is quite inhomogeneous and has an irregular elongated shape. To eliminate the difficulty, we first perform coarse segmentation by utilizing the special braid structure of the lockwire. This is achieved using the MSER algorithm. Regardless of the segmentation grey thresholds of an image, the segmentation result of MSER (maximally stable extremal regions) is almost constant. These regions have either very large or very small gray values. In our case, an illuminated lockwire usually exhibits obvious bright spots with large gray values. Therefore, as shown in Fig. 5, these spots can be extracted using MSER.

However, the extracted bright spots might be separated due to the braided structure of the lockwire. In general, bright spots in an image of a lockwire should have a similar shape, small size, and strong directional correlation. Based on these characteristics, we propose a novel elongated shape

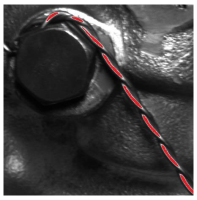

FIGURE 5. An example of bright spot extraction (red lines) on a lockwire using the MSER algorithm.

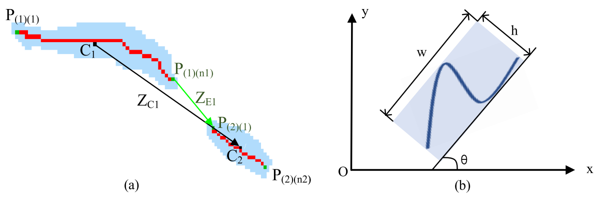

FIGURE 6. Schematic diagram of the parameters used in the shape descriptor. (a) Two kinds of distances: Z_C (black arrow) and Z_E (green arrow). The blue areas are two adjacent bright spots and the red curves are the skeletons. Black points show the center points and green points show the endpoints. (b) Minimum bounding rectangle. The light blue rectangle indicates the minimum rectangle that circumscribes the dark blue curve.

descriptor to determine the continuous region on the lockwire. The shape descriptor is represented by three parameters, which are detailed as follows (T_* is a constant, sgn is the symbolic function):

(i) Distance

As shown in Fig. 6(a), for each region's contour $l_i \in L$, we define $P_{(i)(1)} = \left\{\left(x_{(i)(1)}, y_{(i)(1)}\right) | i = 1, 2, \ldots, k\right\}$ as the first endpoint and $P_{(i)(n_i)} = \left\{\left(x_{(i)(n_i)}, y_{(i)(n_i)}\right) | i = 1, 2, \ldots, k\right\}$ as the last endpoint. Moreover, we define two kinds of vectors: $Z_{C_i} = \{\|C_i - C_{i+1}\| \, | i = 1, 2, \ldots, k-1\}$ and $Z_{E_i} = \left\{\left\|P_{(i)(n_i)} - P_{(i+1)(1)}\right\| \, | i = 1, 2, \ldots, k-1\right\}$, where C indicates the centroid. The two types of distances are $d_i = \left|Z_{C_i}\right|, (i = 1, 2, \ldots, k-1)$ and $d_i' = \left|Z_{E_i}\right|, (i = 1, 2, \ldots, k-1)$. Finally, we define a compositive distance

$$F_{\text{distance}} = [1 - sgn(d_i - T_1)] + \left[1 - sgn(d_i' - T_2)\right] \tag{3}$$

(ii) Aspect ratio

The minimum circumscribed rectangle of an arbitrary curve is shown in Fig. 6(b). Its aspect ratio is $R_i = h_i/w_i$, where w and h are the long and short sides of the rectangle, respectively. We define the following compositive aspect ratio

$$\mathrm{F_{ratio}} = [sgn(\mathrm{R_i} - \mathrm{T_{r1}}) + 1] * [sgn(\mathrm{R_i} - \mathrm{T_{r2}}) - 1] \; (\mathrm{T_{r1}} < \mathrm{T_{r2}}) \tag{4}$$

(iii) Angle

In Fig. 6(b), let θ denote the angle between the long side of the rectangle and the x-axis. Similar to Eq. (4), we define a compositive angle as:

$$F_{\text{angle}} = [sgn(\theta_i - T_{\theta 1}) + 1] * [sgn(\theta_i - T_{\theta 2}) - 1] \; (T_{\theta 1} < T_{\theta 2}) \tag{5}$$

TABLE 1. Correspondence between points on the skeleton and nodes on the skeleton tree.

Skeleton points	Tree Nodes
Endpoints	Root node, leaf nodes
Branch points	Child nodes
Middle points	None

As a consequence, the candidate spots on the lockwire can be selected by adjusting T_*, which changes the distance, ratio, and angle accordingly. Then, the candidate spots on the lockwire with suitable T∗ value can be selected. To connect the two candidate regions of a lockwire more smoothly, we apply the cubic spline interpolation. Formally, let $\nabla f|_{P_{(\mathrm{i})(n_i)}}$ denote the gradient of $P_{(i)(n_i)}$, and $\nabla f|_{P_{(\mathrm{i+1})(1)}}$ be the gradient of $P_{(i+1)(1)}$. The objective function of the cubic spline interpolation S_i (x) is defined as:

$$\begin{cases} S_i(x) = A_i + B_i(x - x_i) + C_i(x - x_i)^2 + D(x - x_i)^3 \\ S_{(i)(0)}'(x_0) = \nabla f|_{P_{(i)(n_i)}} \\ S_{(i)(m-1)}'(x_m) = \nabla f|_{P_{(i+1)(1)}} \end{cases} \tag{6}$$

where $S_{(i)(0)}'(x_0)$ and $S_{(i)(m-1)}'(x_m)$ indicate the gradients of the first and last points in the discontinuous space, respectively.

Eventually, the coarse segmentation of the lockwire can be performed by connecting all candidate regions.

C. LOCKWIRE CENTERLINE EXTRACTION

We design a three-step scheme to precisely extract the lockwire centerline. The first step aims to obtain the skeleton from the elongated region determined in the previous stage (Sec. 3.2). A novel skeleton tree is used to obtain a smooth skeleton in the second step. Then, a texture-based selection strategy is used to extract an accurate centerline from many candidates in the third step. A refining algorithm is used in the first step [14]. The second and third steps are described below.

Generally, the segmentation result from the previous stage will contain a significant amount of noise, and the noise will appear as small burrs on the skeleton. Hence, we propose using a novel skeleton tree to remove these burrs. Table 1 shows the correspondence between points on the skeleton and nodes on the skeleton tree. In addition to the endpoints, the points on the skeleton are divided into branch points and middle points. There are multiple directions to traverse at branch points, while there is only a single direction to traverse at a middle point.

Fig. 7 shows the process of building a skeleton tree to remove burrs. First, the refined result is processed as a single-pixel skeleton (Fig. 7(a)). Then, the skeleton is traversed to build the corresponding skeleton tree. The traversal is detailed as follows. The starting point can be chosen as E_1 or E_2, which should be one of the two endpoints separated by the greatest distance. The traversal direction should be the skeleton's longest possible traversal distance, as shown

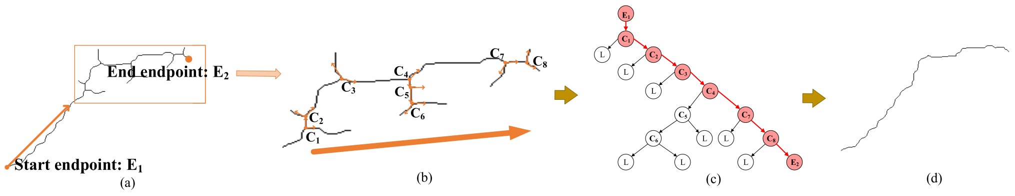

FIGURE 7. The process of removing burrs from the skeleton. (a) One skeleton with eight branch points named C1-C8. (b) Partially magnified image of (a). (c) The skeleton tree. (d) The filtered smooth skeleton that start from the lower left endpoint E_1 to the upper right endpoint E_2. The red nodes and red arrows make up the longest path on the skeleton tree. Points that remain along the red path constitute the smooth skeleton.

in the arrows of Fig. 7(b). Traversal ends when all points in the skeleton are traversed. Meanwhile, the coordinates of each child node in the skeleton tree and the number of the traversed middle points are recorded. As a result, the points on the longest path of this skeleton tree are the points of the centerline. The skeleton tree in Fig. 7(a) is shown in Fig. 7(c), and the smoothed skeleton is shown in Fig. 7(d). It is clear that all burrs on the original skeleton have been removed.

There may be multiple candidates for the lockwire centerline. As one bolt has two lockwires at most, we utilize a texture-based strategy to select the correct candidates. Two kinds of texture features, i.e., contrast and entropy, are used in this strategy. Specifically, contrast reflects the clarity of the image and the depth of the groove while entropy indicates the complexity of the image texture. Both of these texture features are calculated on the basis of the gray level co-occurrence matrix. Assume k is the number of candidates. Let Tex_c denote the contrast, and Tex_e be the entropy. Then, we define a compositive texture feature F_{tex} as

$$\begin{cases} Tex_{cm} = \max\left(Tex_{ci}\right) \\ F_{\text{tex}} = Tex_c + \exp\left[\alpha_c - \min((Tex_{cm} - Tex_{ci})/Tex_{cm})\right] Tex_e \\ (i = 1, 2, \ldots, k) \end{cases} \tag{7}$$

where $\alpha_c = 0.5$. Let $P^{\lambda}_{\Phi,d}(a, b)$ be the value of the co-occurrence matrix. Then, Tex_c and Tex_e are defined as follows

$$Tex_c = \sum\nolimits_{a,b} |a - b|\, P^{\lambda}_{\Phi,d}(a, b) \tag{8}$$

$$Tex_e = \sum\nolimits_{a,b} P_{\Phi,d}(a, b) log_2 P_{\Phi,d}(a, b) \tag{9}$$

where $\kappa = 2$ and $\lambda = 1$. We calculated the texture features of a square window with a size of 100 pixels, i.e., $d \in [1, 99]$. The gray level is compressed from 256 to 16 levels to improve the computational efficiency, thus $a \in [0, 15]$ and $b \in [0, 15]$. We use the gray level co-occurrence matrix at four angles $\phi = 0°, 45°, 90°$, or $135°$ and their mean value are used to get the texture feature. Usually, an image with lockwires has larger contrast and entropy because the image is more complex.

D. TWINING DIRECTION RECOGNITION

In this stage, we determine the twining direction by identifying the relative direction of the extension direction of the lockwire and center-intersection vector of the bolt.

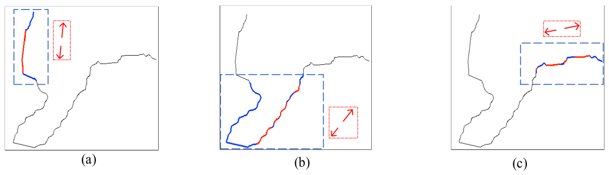

FIGURE 8. Different parts in a curve and their possible directions. The line segment inside the blue rectangle is input into RANSAC. The red points are along the direction vector and the blue points are not along the direction vector.

The former is obtained using the RANSAC algorithm. The latter is determined from the bolt's fine position.

The lockwire centerline may be irregular, leading to multiple extension directions in different centerline segments (Fig. 8). Thus, we select a valid segment based on the intersection of the centerline and the bolt. RANSAC is then used to efficiently fit the direction of the valid segment. Specifically, for a line segment, if the mathematical model of this line segment is set to a straight line, the RANSAC algorithm can be used to estimate a point on the direction vector (Fig. 8) of the segment by iteration. We used the two points on the direction vector that are separated by the largest distance to compute the extension direction-vector in our experiment. Moreover, the direction is reliable because the RANSAC algorithm is robust against noise.

The center-intersection vector of the bolt is defined as $\vec{V_1} = \vec{C_b C_l}$, where C_b denotes the center of the bolt and C_l represents the intersection of the centerline and the bolt. Let $\vec{V_2}$ be the extension direction of the lockwire centerline. Then, as shown in Fig. 9, the lockwire twining direction W can be determined using the signum of the cross product of the two vectors:

$$W = \text{sgn}(\vec{V_1} \times \vec{V_2}) \tag{10}$$

If W is positive, then the twining direction is clockwise. Otherwise, the twining direction is counterclockwise. $W = 0$ if V_1 and V_2 are collinear.

IV. EXPERIMENTS AND RESULTS

We first describe the dataset used in our experiments and evaluation indicators are presented in this section. We then present the overall qualitative and quantitative results and the robustness performance on rotating and low-quality images.

TABLE 2. Distribution of our dataset.

Image Type	Train Set		Test Set		
	Images	Hexagonal Bolts	Images	Hexagonal Bolts	Lockwires
Single-locking	92	104	38	55	54
Double-locking	117	370	35	79	74
Multiple-locking	81	298	28	82	102
Total	290	772	101	216	230

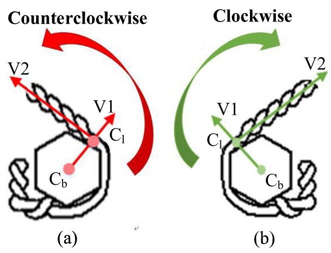

FIGURE 9. Lockwire twining direction identification: (a) false lockwire twining direction and (b) correct lockwire twining direction. The center point of the hexagonal blot is C_b and the lockwire hole is C_l. V_2 is the unit extension direction vector of lockwire.

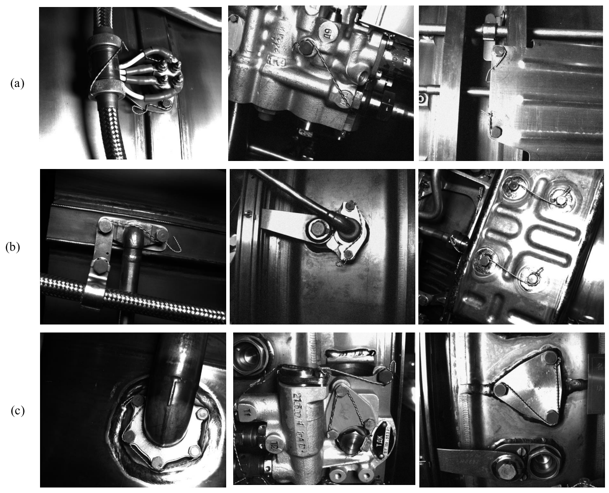

FIGURE 10. Example images from our dataset: (a) single-locking; (b) double-locking; (c) multiple-locking.

A. DATASET DESCRIPTION

To the best of our knowledge, there is no publicly available dataset on lockwires at present. Therefore, we collected 391 images of an aircraft engine using an industrial camera (1 MER-504-10GM-P, Imavision) and lens (1 M1214-MP2, Computar) with 12 mm focal length. This camera provided images with 2048×2048 pixels.

The images were randomly separated into training and testing sets with 290 and 101 images, respectively. The images were divided into three categories based on the number of locked bolts: single-locking, double-locking, and ultiple-locking. The distribution of our dataset is shown in Table 2, and example images are shown in Fig. 10. The test set contained 122 lockwires with correct twining direction and 108 lockwires with incorrect twining direction.

The training set was used to build an Adaboost-based model. All hexagonal bolts in the training images are labeled

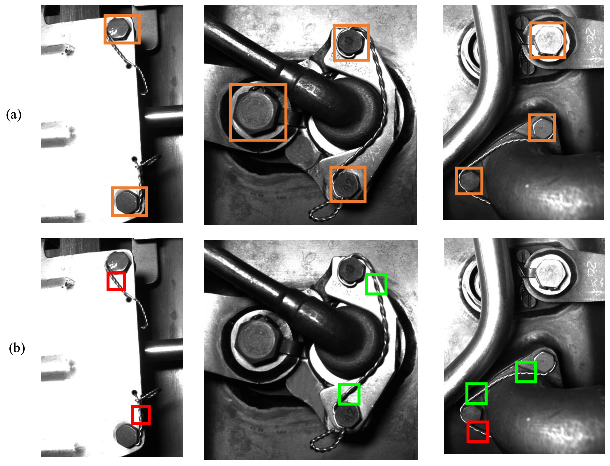

FIGURE 11. Example result images: (a) hexagonal bolt detection results and (b) lockwire twinging direction recognition results. Images in the first column have a single lockwire. Images in the second column have two lockwires, and images in the third column have multiple lockwires.

TABLE 3. Some parameters of the pdollar toolbox.

Parameter Name	Value	Parameter Name	Value
Stride	4	Cell	4
Cascade	[32 128 512 2048]	Block	2
Size	[50 400]	Orients	6

similar to the format used in the PASCAL VOC dataset [19]. It is worth noting that there may be occluded hexagonal bolts in some images. These occluded bolts may be intact in other images because the images were captured at different shooting angles. Thus, these hexagonal bolts were not labeled.

As shown in Table 2, the number of lockwires in the test images is less than the number of hexagonal bolts. The reason is that some hexagonal bolts are not assembled with lockwires. These hexagonal bolts are omitted in the following steps related to the lockwire.

B. EVALUATION INDICATORS

We used different performance evaluation measures to evaluate different steps. The bolt detection and lockwire detection results are evaluated using recall. The recognition result is assessed using the accuracy measure.

If the test set contains a hexagonal bolts and m lockwires, and only b hexagonal bolts and n lockwires are detected, then c hexagonal bolts and k lockwires were detected incorrectly. Meanwhile, the twining direction of p lockwires was misidentified. The recall of hexagonal bolt detection can be defined

TABLE 4. Hexagonal bolt detection results.

Image Type	Total Hexagonal Bolts	Undetected	Suspected	False
Single-locking	55	0	55	0
Double-locking	79	0	79	0
Multiple-locking	81	1	80	0

TABLE 5. Lockwire detection and twining direction recognition results.

Image Type	Total Lockwires	Detection			Twining Direction Recognition	
		Undetected	False	Recall (%)	False	Accuracy (%)
Single-locking	54	1	0	98.1	4	92.5
Double-locking	74	1	1	97.2	5	93.2
Multiple-locking	102	14	0	86.3	4	95.4

as

$$\text{Recall}_{bolt} = ((b-c)/a) \times 100\%, \tag{11}$$

The recall and accuracy of lockwire detection can be defined as

$$\text{Recall}_{lockwire} = ((n-k)/m) \times 100\% \tag{12}$$

$$\text{Accuracy}_{lockwire} = ((n-k-p)/(n-k)) \times 100\% \tag{13}$$

The ratio of lockwires with clockwise twining direction to those with counterclockwise twining direction in the test is 1:1, thus the accuracy as defined in Eq. (13) can be used to describe the performance of the algorithm.

C. OVERALL PERFORMANCE

Fig. 11 shows example images of the results, including the bolt detection and twining direction recognition result.

The Adaboost-based model introduced in Section 3.1 was created with pdollar Matlab toolbox [20] to detect hexagonal bolts. Table 3 shows some parameters used in the pdollar toolbox. The detector used a step size of 4 pixels, training was divided into 4 stages, and the number of decision trees trained at each stage was [32 128 512 2048]. The final detection model consisted of 2048 depth-2 trees. The cell size was set to 4 and the block size was set to 2. Considering that the bolt is hexagonal, the entire 360° area around the bolt is divided into 6 equal intervals. Each pixel casts a vote for the bin corresponding to its gradient orientation that is weighted by its gradient magnitude. The entire training process required approximately 7 to 9 minutes.

We tested the Adaboost-based model against the test set (101 images, 216 hexagonal bolts); the results are shown in Table 4. As can be seen from Table 3, there is 1 undetected hexagonal bolt with no errors. The detection recall of the model reaches 99.5%. The results suggest that our model can be used to accurately locate all hexagonal bolts in the images.

Afterwards, we tested the algorithm for detecting lockwires on the test set (101 images, 230 lockwires). Table 5 shows the result of the lockwire detection and twining direction recognition. The detection recalls for single-locking and double-locking images are 98.1% and 97.2%, respectively. Both the single-locking and double-locking images have 1 omitted bolt. All single-locking images were detected correctly, and only 1 double-locking image was detected incorrectly. The multiple-locking images have the worst lockwire detection results with 14 undetected hexagonal bolts with a recall of only 86.3%. There is one main reason why the detection recall from the multiple-locking images is the lowest. When a bolt is equipped with two lockwires and the two lockwires extend along the same direction, our algorithm may mistake the two lockwires for the same lockwire. Most bolts in the multiple-locking images are equipped with two lockwires, so there are more omissions in this kind of image.

Although the recall of lockwire detection for the three types of images is different, the accuracies of the twining direction recognition are nearly the same. From Table 5, there are 13 lockwires with incorrect twining direction in total, including 4 single-locking images, 5 double-locking images, and 4 multiple-locking images. The multiple-locking images have the highest recognition rate (95.4%). The difference in accuracy for single-locking images compared to double-locking images is only 0.7%. The accuracy for detecting single-locking and double-locking images reaches 92.5% and 93.2%, respectively.

Some results from our proposed method are shown in Fig. 12, which includes results for all three types of images. One can see that the background in some images is relatively simple, but it contains a lot of noise. Even so, the proposed method still provides high recognition accuracies.

D. ROBUSTNESS TESTS

Images captured in a practical environment may exhibit slight rotation or have low quality. To further test the robustness of the proposed method, various rotations were applied to the test images to simulate various industrial imaging situations. In addition, we tested the robustness of the proposed method with low-quality images.

1) ROBUSTNESS TO ROTATIONS

The original test images were rotated within −12° to 12° in 4° intervals. This rotation range is sufficient to cover most rotation cases in various imaging conditions. The test results

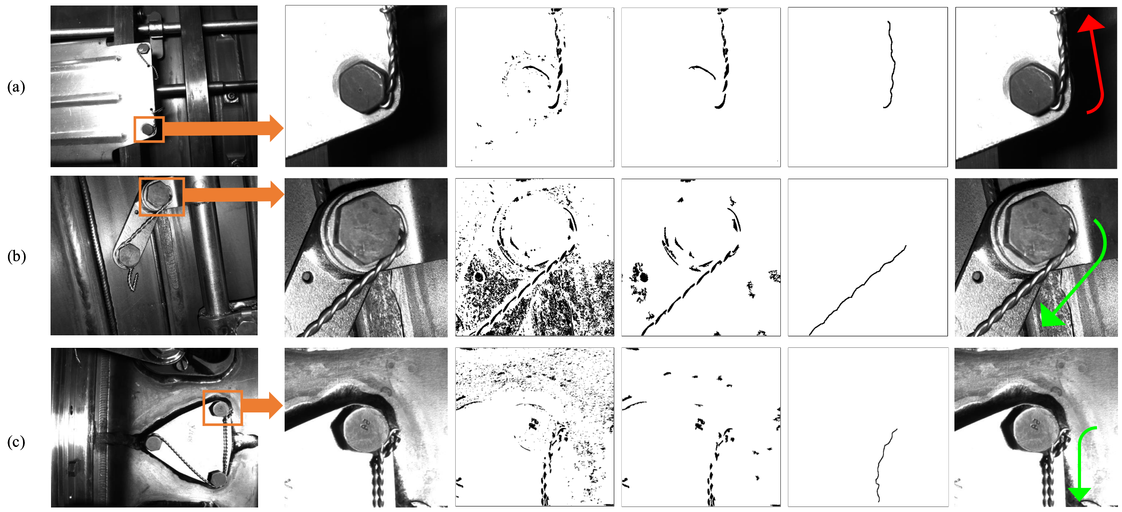

FIGURE 12. Some intermediate results of the proposed method on automatically identifying the twining direction of (a) single-locking lockwire, (b) double-locking lockwire, and (c) multiple-locking lockwire.

TABLE 6. Lockwire detection and twining direction recognition results (counterclockwise rotation).

Image Type	Hexagonal Bolt Recall (%)			Lockwire Recall (%)			Lockwire Accuracy (%)		
	+4 °	+8 °	+12 °	+4 °	+8 °	+12 °	+4 °	+8 °	+12 °
Single-locking	100	98.1	98.1	100	100	100	95.1	94.0	93.0
Double-locking	97.8	98.9	96.8	98.7	96.2	98.7	94.3	95.6	94.2
Multiple-locking	97.6	94.0	94.0	83.5	85.6	84.9	95.7	96.4	95.2

TABLE 7. Lockwire detection and twining direction recognition results (clockwise rotation).

Image Type	Hexagonal Bolt Recall (%)			Lockwire Recall (%)			Lockwire Accuracy (%)		
	-4 °	-8 °	-12 °	-4 °	-8 °	-12 °	-4 °	-8 °	-12 °
Single-locking	98.1	96.2	96.2	96.1	100	97.9	94.0	93.8	95.7
Double-locking	98.9	96.8	95.8	98.7	97.4	98.6	93.2	94.7	94.6
Multiple-locking	98.8	98.8	96.4	84.4	87.5	88.3	95.3	96.7	95.8

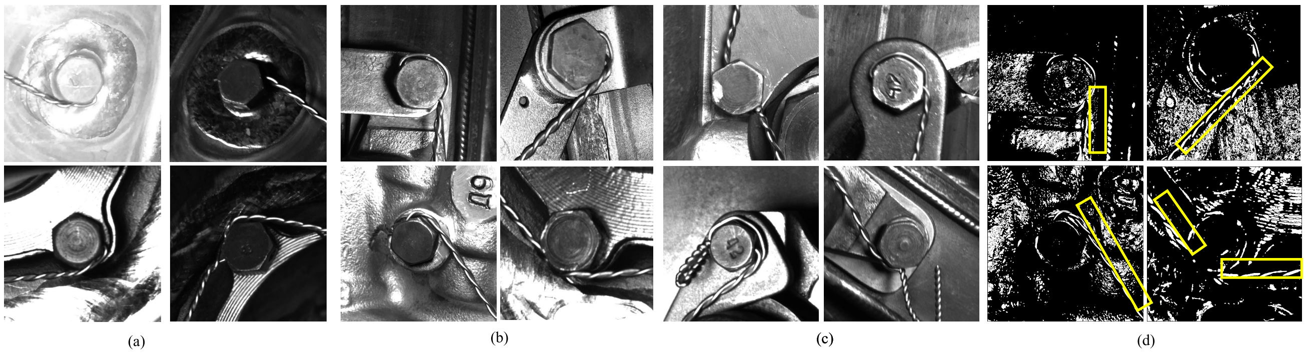

FIGURE 13. Some low-quality images: (a) images with uneven illumination; (b) low-quality images with significant noise; (c) low-quality images with low-quality hexagonal bolts; (d) segmentation results of (b). Images in the yellow rectangle are lockwires.

on the rotated images are shown in Table 6 and 7, where a positive number indicates counterclockwise rotation.

From Table 6 and 7, one can see that there is no significant decrease in accuracy as the angle of rotation increases. One can see that the lowest accuracy in the single-locking image is 93.0% within 12° rotation. For the other two types of images, the lowest accuracies are 93.2% and 95.2%. The results show that the accuracy after rotation is nearly equal to the accuracy before rotation. Therefore, our method is robust enough to rotation up to 12°.

2) ROBUSTNESS TO LOW-QUALITY IMAGES

We collected 123 low-quality images to further evaluate our proposed method. Fig. 13 shows some example images.

TABLE 8. Lockwire detection and twining direction recognition results of with low-quality images.

Image Type	Total Lockwires	Lockwire Detection		Twining Direction Recognition	
		Undetected	Recall (%)	False	Accuracy (%)
Poor Background	65	6	90.8	2	96.5
Low Quality Hexagonal Bolts	58	1	98.2	5	91.2

The low-quality images have either a poor background, such as uneven illumination and a lot of noise, or low-quality hexagonal bolts with uneven gray scale on the surface. The results are presented in Table 8. Although these images are of low quality, the proposed method still provides considerable accuracy.

V. CONCLUSION

We propose an automatic lockwire twining direction recognition (LTDR) method based on the machine learning and image processing in this paper. To the best of our knowledge, our method is the first designed for automatic LTDR tasks. Our experimental results show that our method provides high recall ratio and accuracy in lockwire detection and twining direction recognition. The results also show that our method is quite robust against low-quality images and diverse rotations. Moreover, our method provides precise determination of a bolt's position and lays a foundation for defect detection in bolts.

The proposed method only provides LTDR with limited illumination at this stage. Because bright spots on the surface of a lockwire may disappear in low light, it will be difficult to detect and identify the lockwire, resulting in decreased accuracy. In future research, we will improve the robustness of the proposed method such that it can be used in different illumination conditions.

REFERENCES

[1] J. Bickford, *Handbook of Bolts and Bolted Joints*. New York, NY, USA: Marcel Dekker, 1998, ch. 40, pp. 757–824.

[2] (2019). *Safety Wire*. Accessed: Jul. 16, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Safety_wire

[3] *Wire, Safety or Lock*, NASM, Chandler, AZ, USA, 1998.

[4] 2019. *Byron Gliding Club. The Use of Lock Wire—A Guide*. Accessed: Jul. 16, 2019. [Online]. Available: http://byrongliding.com/technical/lockwiring/

[5] *Requirements for Threaded Fastening Systems in Spaceflight Hardware*, NASA, Washington, DC, USA, 2012.

[6] D. P. Hess, "Threaded fastener secondary locking requirements," *J. Failure Anal. Prevention*, vol. 17, no. 4, pp. 724–730, 2017.

[7] G. Nan and J. Yao, "A real-time visual inspection method of fastening bolts in freight car operation," *Proc. SPIE*, vol. 9675, Oct. 2015, Art. no. 96752G.

[8] A. Rudawska, S. Cisz, and T. Warda, "Selected methods for locking screw joints, including the use of adhesives, used in the helicopter construction," *Technol. Eng.*, vol. 11, no. 2, pp. 26–31, 2014.

[9] D. P. Hess, "Threaded fastener locking with safety wire and cotter pins," *J. Failure Anal. Prevention*, vol. 18, no. 5, pp. 1216–1223, 2018.

[10] H. de Oliveira, E. Grassi, P. Espindola, and C. de Araujo, "Smart lockwire: A shape memory alloy lockwire for improved reliability in bolted fixing in automotive and aeronautical applications," SAE Tech. Paper 2012-36-0244, 2012.

[11] J.-H. Chen, M.-C. Su, S.-C. Hsu, J.-C. Lu, and R. Cao, "A self organizing map optimization based image recognition and processing model for bridge crack inspection," *Autom. Construct.*, vol. 73, pp. 58–66, Jan. 2017.

[12] A. Garcia-Garcia, S. Orts-Escolano, M. Cazorla, and J. Garcia-Rodriguez, "Interactive 3D object recognition pipeline on mobile GPGPU computing platforms using low-cost RGB-D sensors," *J. Real-Time Image Process.*, vol. 14, no. 3, pp. 585–604, 2018.

[13] P. Zhao *et al.*, "Construction of mathematical model for the bending state of braided wire rope," in *Proc. Int. Conf. Machinery Mater. Inf. Technol. Appl.*, vol. 71. Paris, France: Atlantis Press, 2017, pp. 1422–1425.

[14] A. Sironi, V. Lepetit, and P. Fua, "Projection onto the manifold of elongated structures for accurate extraction," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 316–324.

[15] K. Wang *et al.*, "Natural scene text detection based on MSER," in *Proc. Int. Conf. Commun. Inf. Manage. Netw. Secur.*, vol. 65. Paris, France: Atlantis Press, 2018, pp. 92–95.

[16] E. Salahat and M. Qasaimeh, "Recent advances in features extraction and description algorithms: A comprehensive survey," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Mar. 2017, pp. 1059–1063.

[17] G. Nan and J. E. Yao, "A real-time visual inspection method of fastening bolts in freight car operation," *Proc. SPIE*, vol. 9675, Oct. 2015, Art. no. 96752G.

[18] L. Ramana, W. Choi, and Y.-J. Cha, "Automated vision-based loosened bolt detection using the cascade detector," *Sensors Instrum.*, vol. 5. Cham, Switzerland: Springer, 2017, pp. 23–28.

[19] S. Soni and S. Kaur, "To Propose an improvement in Zhang-Suen algorithm for image thinning in image processing," *Int. J. Sci. Technol. Eng.*, vol. 3, no. 1, p. 7481, 2016.

[20] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[21] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

[22] Y.-C. Cha, K. You, and W. Choi, "Vision-based detection of loosened bolts using the Hough transform and support vector machines," *Automat. Construct.*, vol. 71, pp. 181–188, Nov. 2016.

[23] C. Yan, H. Xie, S. Liu, J. Yin, Y. Zhang, and Q. Dai, "Effective uyghur language text detection in complex background images for traffic prompt identification," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 220–229, Jan. 2018.

JUNHUA SUN was born in Hubei, China, in 1975. He received the B.E. and M.E. degrees from the Beijing Institute of Machinery, China, in 1997 and 2003, respectively, and the Ph.D. degree in precision instrument and machinery from Beihang University, Beijing, China, in 2006.

He is currently a Professor with Beihang University. He is currently a Key Member of the Key Laboratory of Precision Opto-Mechatronics Technology of Ministry of Education. His research interests include machine vision, image processing and recognition, and machine learning.

LEWEI HUANG was born in Sichuan, China, in 1995. She received the B.E. degree from the Huazhong University of Science and Technology, Wu'han, China, in 2017. She is currently pursuing the M.E. degree with the Key Laboratory of Precision Opto-Mechatronics Technology of Ministry of Education, Beijing, China.

Her research interests include image recognition and machine learning.

JIE ZHANG was born in Jiangxi, China, in 1990. She received the Ph.D. degree in instrumentation science and technology from Beihang University, in June 2018.

She was a Visiting Researcher with the School of Informatics, University of Edinburgh, from 2016 to 2018. She is currently an Associate Professor with the Beijing Technology and Business University. Her research interests include 3D computer vision, covering 3D shape representation, 3D dynamic face recognition, and 3D vision measurement.