COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING

Computer-Aided Civil and Infrastructure Engineering **30** (2015) 759–770

Vision-Based Automated Crack Detection for Bridge Inspection

Chul Min Yeum

The School of Civil Engineering, Purdue University, West Lafayette, IN, USA

&

Shirley J. Dyke*

The Schools of Mechanical Engineering and Civil Engineering, Purdue University, West Lafayette, IN, USA

Abstract: *Visual inspection of bridges is customarily used to identify and evaluate faults. However, current procedures followed by human inspectors demand long inspection times to examine large and difficult to access bridges. Also, highly relying on an inspector's subjective or empirical knowledge induces false evaluation. To address these limitations, a vision-based visual inspection technique is proposed by automatically processing and analyzing a large volume of collected images. Images used in this technique are captured without controlling angles and positions of cameras and no need for preliminary calibration. As a pilot study, cracks near bolts on a steel structure are identified from images. Using images from many different angles and prior knowledge of the typical appearance and characteristics of this class of faults, the proposed technique can successfully detect cracks near bolts.*

1 INTRODUCTION

Visual inspection is the primary method used to evaluate bridge condition. Most decisions relating to bridge maintenance are based on assessments from visual inspections. However, current visual inspection conducted by human inspectors has several limitations. The study, conducted by the U.S. Federal Highway Administration's Nondestructive Evaluation Validation Center (NDEVC) in 2001, investigated accuracy and reliability of routine and in-depth visual inspections, which are regularly scheduled inspection every 2 years (Phares et al., 2001). The study showed that there is a great discrepancy in the results when the same structure is inspected by several inspectors. Factors include accessibility, light intensity, lack of specialized knowledge, perception of maintenance, and visual acuity and color vision. Although some factors are attributed to carelessness or improper training of inspectors, most of them cannot be physically overcome by a human in the current visual inspection process.

To tackle this issue, initially, visual information from the bridge should be remotely accessed and collected automatically according to established standardized procedures. In the literature, many researchers have proposed remote access of image acquisition systems for capturing images under or over bridges. A visual monitoring system was proposed by controlling several cameras mounted on bridges to collect images (Jahanshahi et al., 2011). Using these images, the scenes of bridges are periodically constructed to evaluate the evolution of cracks or corrosion. Another approach is to develop equipment for improving accessibility to large bridges. U-BIROS (Ubiquitous Bridge Inspection Robot System) is a robotic image acquisition system, which scans bridges using a robotic arm equipped with cameras (Lee et al., 2011). This system is similar to an under-bridge inspection vehicle but replaces a bucket with cameras. The California Department of Transportation (Caltrans) bridge inspection project developed a wired aerial robotic platform for close inspection of bridges or other elevated highway structures (Moller, 2008; Miller, 2004). The vehicle is capable of vertical takeoff

*To whom correspondence should be addressed. E-mail: *sdyke@purdue.edu*.

DOI: 10.1111/mice.12141

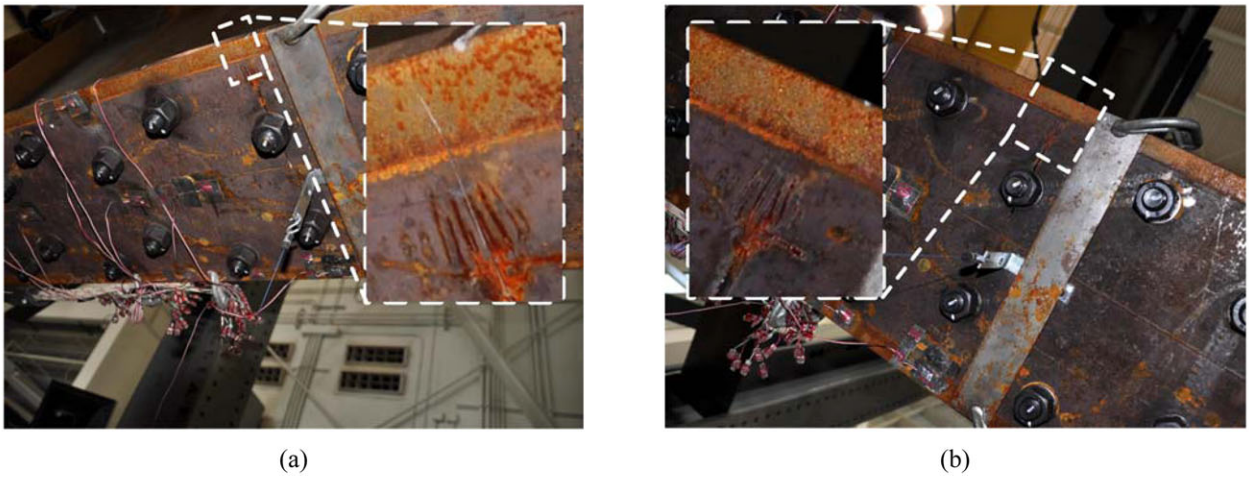

Fig. 1. Images of a fatigue crack on a steel beam, captured from different viewpoints. Note that the fatigue crack in the images is initiated from the tip of the artificial notch near the bolt, which is to simulate initial discontinuity of the circumference of the bolt hole.

and landing, translation to horizontal movement and orienting a video camera, all controlled by operating personnel on the ground. Recently, a multi-rotor helicopter became commercially available to wirelessly take pictures or videos for inspection of structures such as pipelines, power lines, or dams (Angel Aerial Survey, 2014; US Aerial Video, 2014).

Once a suitable collection of visual information is collected from a bridge, robust inspection techniques should automatically perform the visual inspection tasks set forth in the manual. This autonomous inspection is not a new concept and is widely developed and used for civil, mechanical, or aerospace structures (Abdel-Qader et al. 2003; Shen et al., 2013; Zhu et al., 2011; Malekian et al., 2012; O'Byrne et al., 2013, 2014; Jiang and Adeli, 2007). In this study, the focus is on cracks because it is a common type of damage in civil structures and is typically visually clear in appearance. A review of vision-based crack detection techniques is available in references Jahanshahi et al. (2009, 2013) and Torok et al. (2013).

Futuristic visual inspection is motivated and imagined as follows. An unmanned aerial vehicle (UAV) equipped with high resolution cameras arrives at a candidate bridge for inspection. Following preliminary designed flying paths or navigating automatically, the UAV collects and records images. The flying path is periodically updated based on previous inspection records, for example, taking more images in damaged areas as detected in previous flights. The UAV transmits collected images to a base station. At the base station, processing takes place on the large volume of images, and damage on the structure is detected, localized, and quantified automatically without human inspectors. The system automatically generates an inspection report to help expert visual inspectors make decisions whether the tested bridge requires further inspection or prompt maintenance. By preserving and archiving such reports and decisions over the lifetime of the structure, the system would continue updating to be more robust and smarter in the inspection of damage, reducing false alarms or misdetections.

This study was begun by exploring the question: Given a large volume of images from the UAV, would it be feasible to detect damage in a realistic structure using currently available vision-based damage detection techniques? To answer this question, multiple photographs were taken of a steel beam having a real fatigue crack that initiated from one of bolt holes. Analysis of the data was performed using standard available methods found in the literature, such as edge detectors or morphological detectors (Jahanshahi et al., 2009). As a result, we identified two major issues that need to be addressed for vision-based automated inspection, which previous researchers may or may not have observed. First, searching for cracks over the entire area of an image generates many false-positive alarms and misdetections. In Figure 1a, there are many crack-like features such as structure boundaries, wires, or corrosion edges, causing either incorrect detection or a failure to detect real cracks due to its narrow width. However, in these cases detection of the real crack by human inspectors is not easy but is still possible. This detection is because they have prior knowledge about the crack's typical appearance and characteristics. In this case, the relevant information is that cracks on a steel structure have thin and shiny edges and are often initiated and propagated from bolt holes (Indiana Department of Transformation, 2013). These features draw their attention to bolts and nearby areas, facilitating crack detection. A second issue observed is that the crack may be visible or invisible depending on the viewpoint. Figure 1b shows identical scenes of a fatigue crack but from two different viewpoints. Comparing the white dotted boxes in both images, Figures 1a and b, the crack is hardly observed in the second image. Therefore we conclude that images of same scene from many different viewpoints are needed to detect the crack without knowing how and where it is created and propagated. Most previous research, of course, has unconsciously considered these two issues as they collect images through controlling the circumstances, such as camera positions or angles depending on appearance and location of cracks. However, in reality, it is hard to obtain sufficiently good images taken under the "best" conditions because the crack location, crack direction, and lighting direction cannot be known in advance, and also it is hard to precisely and continuously control camera positions and angles installed in the UAV.

In an attempt to consider the above two findings, we build the framework for the proposed technique. In this study, an automated crack detection technique is proposed using images collected under uncontrolled circumstances. Rather than searching for cracks on entire images, objects which have areas susceptible to cracks (bolts in this study) are first detected in the images. This step greatly increases detectability of cracks by narrowing down searching areas and scales in acquired images. Object detection and grouping techniques used in computer vision areas are implemented to extract, match, and group the same objects from many angles across images. Crack-like edges, which are similar in appearance to real cracks, are first detected from images of object areas using image processing techniques. Then based on prior knowledge of crack's typical appearance and characteristics, a decision is made whether crack-like edges are true cracks or not.

In this article, as one of visual inspection tasks, cracks occurring near bolts on a steel structure are detected. However, users can extend the proposed visual inspection framework to conduct other types of visual inspection. For example, suppose that corrosion or crack damage in gusset plates is damage of interest. Gusset plates become "objects" in this study and techniques suggested in Sections 2.1 to 2.3 can be applied to extract images of individual gusset plate from different angles. Users analyze images of all gusset plates in a test bridge by applying a corresponding crack detection technique.

The major contribution of the proposed technique is to propose a new approach to automated visual inspection using a large volume of images. Many previous researchers have focused on detecting cracks from a few images taken from set positions where cracks are visually clear. However, automatic image collection using aerial cameras or other equipment would not guarantee that favorable images would be obtained due to the uncertainty in crack's location and direction. Instead, the proposed technique begins by searching damage sensitive areas from a large pool of images. By detecting these areas from many different viewpoints, detectability of damage can be dramatically increased even if it is small, and false-positive alarms can be reduced by limiting searching areas. To the best knowledge of the authors, there is no literature developing vision-based visual inspection with this concept.

The remainder of this article is organized as follows. Section 2 starts from the brief overview of the proposed approach and provides technical details about image acquisition, object detection and grouping, and crack detection. Experimental descriptions and results are presented in Section 3. Section 4 includes summary and conclusion.

2 METHODOLOGY

The overview of the proposed technique is shown in Figure 2. First, in Figure 2a, images of the structure from many angles are collected using image acquisition equipment. This step may involve one or more possible ways of image collection such as aerial cameras or inspection robots. Second, in Figure 2b, target structural components called objects, which are susceptible to cracking, are detected and extracted from the images. The object patch indicates one such object and its nearby area where cracking is more likely to present. Third, in Figure 2c, common object patches (corresponding to the same object) across the collection of images are matched and grouped. Finally, in Figure 2d, the proposed crack detection technique diagnoses that a crack exists in the structural components. In this article, fatigue cracks initiated from bolt holes are chosen as target damage. Thus, the terms of "structure" and "object" in this section indicate the bridge and bolt, respectively. However, the proposed technique can be easily generalized to detect cracks from any structural components such as joints or welded areas and on any structure, not limited to bridges. The latest techniques in computer vision are implemented to increase the quality of object detection, object grouping, and crack detection.

2.1 Image acquisition

For image acquisition, the UAV flies under or over bridges by following a predetermined flying path, and cameras installed capture scenes of bridges

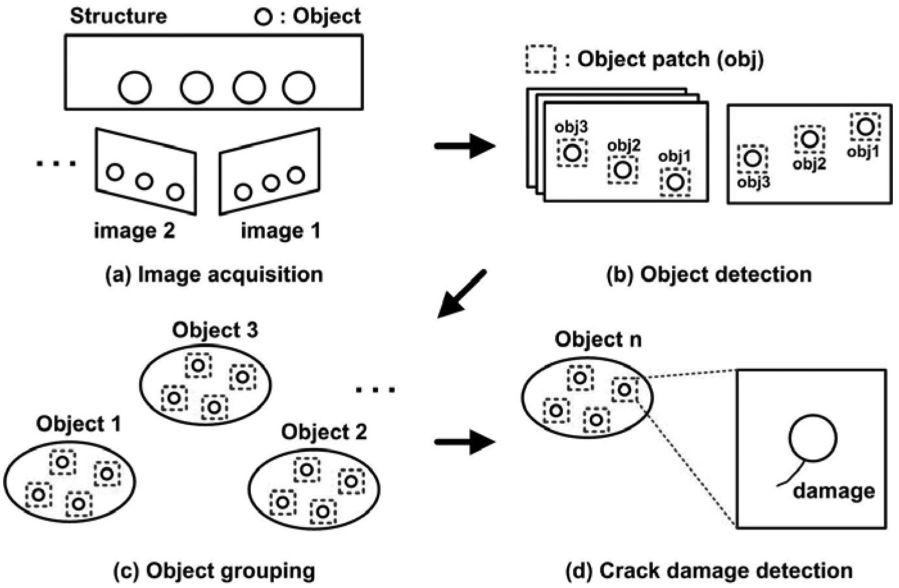

Fig. 2. Overview showing the procedure for the proposed damage detection technique: (a) acquisition of images from multiple angles, (b) detection of object patches which may include damage, (c) grouping of same objects across images, and (d) detection of crack damage initiated from objects.

consecutively. Some guidelines for image acquisition are suggested here for the best performance of the proposed technique, but are not required: (1) Images are not blurred and have sufficient resolution so that objects and cracks are clearly shown in images. (2) Angles between the camera and bridge, called tilt or perspective angle, should not be large. If interest objects on the bridge are presented within a short distance of each other, like bolts, their scenes may overlap on the images under large perspective angle, making object and crack detection difficult. However, angle variation of the camera is necessary due to the dependency of the crack's appearance on the angle of the images. (3) Distance between the UAV and bridge stays roughly constant. The number of scale images that need to be searched can be reduced by known approximate distances and physical bolt sizes (Jahanshahi et al., 2013). This results in increasing object detection rates or decreasing false-positive errors with low computation time, and (4) GPS data of each image are recorded and saved for approximated estimating crack locations.

2.2 Object detection

Object detection is challenging because an object's position, pose, scales, lighting, and background vary relative to camera angles and positions. The key of object detection is the selection of robust features that uniquely represent the object without affecting the above variations. There are no perfect solutions that achieve the high level of perception of a human, but many researchers have improved detectability of objects such as faces or pedestrians (Viola and Jones, 2001; Dalal and Triggs, 2005; Dollár et al., 2009; Felzenszwalb et al., 2010). In this study, modifications to established object detection algorithms are made for our purpose.

In this study, an integral channel-based sliding window technique is applied over multiple scales of images. The sliding window technique uses a fixed rectangular window that slides over the images to decide whether the window contains an object or not. To make this judgment, features are extracted from each window. Here, the channel image indicates linear and non-linear transformation of the original image to help discriminate the object from non-objects as a preliminary process of feature extraction. A total of 11 types of image channels are used in this study: H and S components in HSV color space, U and V components in LUV color space, gradient magnitude, and histogram of gradient with 6 orientations (HOG). The details of HSV and LUV color formats can be seen in reference Schanda (2007). HSV and LUV color spaces can separate colors from brightness which is not robust under lighting variation. Thus, the V component in HSV and the L component in LUV, which are brightness terms, are ignored from channel images. To consider the object shape, a gradient magnitude and HOG are used. These gradient features have been proven in many applications for object detection (Dalal and Triggs, 2005; Dollár et al., 2009; Felzenszwalb et al., 2010). Figure 3

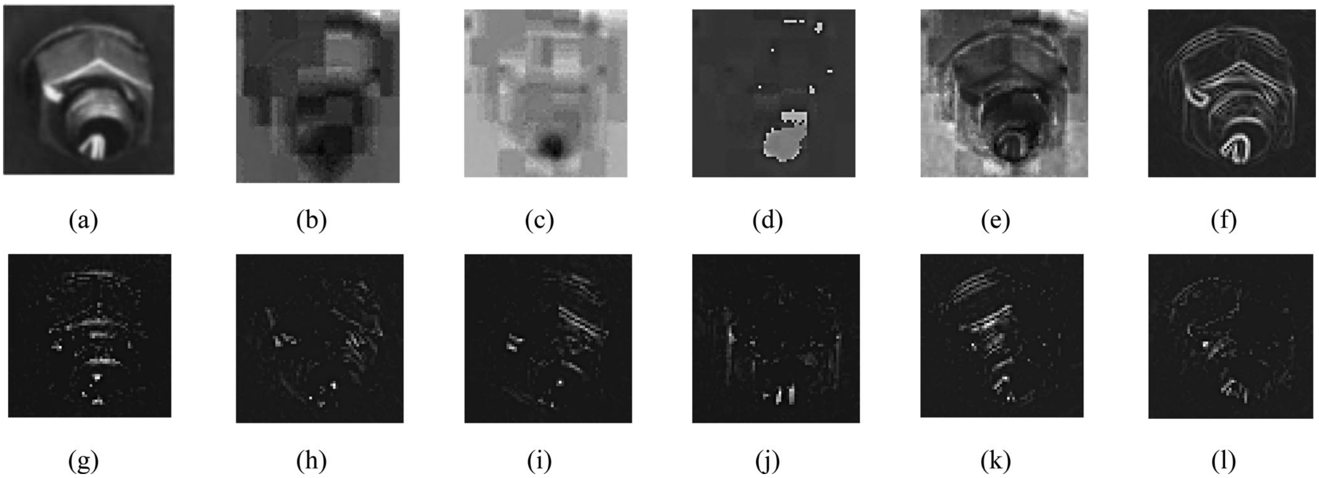

Fig. 3. Examples of channel images of a bolt: (a) original RGB (herein grayscale), (b, c) U and V components in LUV, (d, e) H and S components in HSV, (f) gradient, and (g–l) histogram of gradient with 6 orientations (0°, 30°, 60°, 90°, 120°, 150°).

shows channel images of the bolt. Channel images in a grayscale from Figures 3b to l are computed from the original RGB image (Figure 3a). The intensity of these images varies depending on positions or color and this variation represents the unique feature of the object patch. For example, Figures 3g–l show how different edges of the bolt are highlighted depending on gradient direction.

Using these channel images, features of each window are computed by summing over a local rectangular region using Haar-like wavelets (Viola and Jones, 2001). The integral image provides effective ways of calculating the local sum of channel images. For simplicity, 1, 2, 3, and 4 rectangular Haar-like feature windows are used, which was proposed by the original work of Viola and Jones (2001). Features are computed from all training positive (object patches) and negative (non-object patches) windows. Further modifications, not used in this study, are possible depending on complexity of object appearance and shape by increasing the number of possible rectangular or rotating window.

Based on these features, a robust classifier is designed to determine whether features at a certain window in a test image indicate an object or not. In this study, a boosting algorithm is implemented to produce a robust classifier. Boosting is a way of combining many weak classifiers to produce a strong classifier. By updating different weights of weak classifiers adaptively depending on misclassification errors, the optimum strong classifier which minimizes misclassification errors can be obtained. There are several boosting algorithms introduced in the literature, but in this study, the gentle boost algorithm, proposed by Friedman, is used because it is known as simple to implement, numerically robust, and experimentally proven for objection detection. The details of the gentle boost algorithm can be found in the following references (Friedman et al., 2000; Torralba et al., 2004).

2.3 Object grouping

Object grouping in this study is a process of matching two or more patches with the same object across the images, and dividing them into groups of same object patches. If incorrect matching does not occur, matched object patches are simply assigned as a same group. However, in reality, spurious matching does not allow such simple division. Moreover, object matching, especially for the applications in this study, is much more difficult than conventional matching problems because all bolts in an image have very similar appearance and are closely presented, causing failures in generating unique descriptors. To address these difficulties, robust matching and grouping algorithms are proposed by integrating conventional matching algorithms and introducing a community detection technique for grouping.

In general, object matching is accomplished by corresponding keypoints of object patches between images or selecting the closest object to an epipolar line after finding a fundamental matrix of a pair of images (Lowe, 2004; Snavely et al., 2008; Hartley and Zisserman, 2003). However, in our application, a single use of these techniques produces large errors in matching. First, object patches having similar appearance in each image cannot be uniquely described by keypoints inside, causing wrong keypoint matching across images. Second, epipolar constraint can help to remove the worst of the non-corresponding object patches, but not all of them. The center of the object patch does not indicate the geometrical center of the object, and thus the epipolar line, which is computed from the center of the object patch, does not exactly pass through the corresponding object patch's center. Moreover, due to the regional proximity of objects, more than one object is close to a certain epipolar line, making it difficult to determine a corresponding object (Hartley and Zisserman, 2003).

In this study, these two techniques are simply integrated for better performance on matching objects. Suppose that the object patch in the first image, called target object patch, matches with one of the object patches in the second image. The object patch in the second image is searched by ensuring that the distance between its center and the epipolar line computed from the target object patch is within a set threshold. If more than one object patches satisfy these criteria, keypoints inside of them are matched with those in the target object patch. The object patch having the maximum number of keypoints matched is selected as the corresponding patch of the target object patch. The integrated use of both keypoint matching and epipolar constraint highly improve matching performance. Correspondence of object patches in every pair of images are found using this integrated matching technique.

Despite the use of the proposed matching technique, not all object patches are correctly matched with the same object across images. The final step is grouping object patches based on the matching results between object patches. This structure is almost identical to the community structure, which is widely used in data mining from large-scale data describing the topology of network such as a social network. The object patches (nodes) in a group (community) have dense matching (connection) internally but, some nodes are shared with other nodes in other communities due to incorrect connections. From the standpoint of a network structure, the problem of grouping object patches is considered as community detection.

A very well-known community detection technique called modularity maximization is applied in this study. Modularity is defined as a measure of the quality of particular division of a network into community. The optimum community structure is obtained by iteratively amalgamating communities to maximize modularity in a network. The adjacent matrix A, which defines the connection between nodes, is the only input to this technique. The matrix A is defined as A_{ij} if nodes i and j are connected, otherwise 0, where A_{ij} is the element at ith row and jth column. This matrix can easily be generated from the matching results. The technical details are provided in Clauset et al. (2004).

2.4 Crack damage detection

The proposed crack detection technique is applied to all groups of object patches, which are found in the previous step, to decide whether cracks exist or not. Before introducing the technique, the type of crack referred to in this study should be clearly defined. The crack visually has a very sharp edge and an almost straight line, and initiates from bolt holes on steel plates because bolt holes inherently have initial discontinuity in circumference. A detectable crack in an image is one that can be detected by human vision. Such "preliminary" information helps to detect the crack of interest by filtering out non-crack edges.

The first step in the proposed approach is to remove the bolt areas from object patches. Each object patch found in the previous step includes both the bolt and its nearby area. The region of interest on each patch is the area connected to the bolt, but not the bolt itself. The benefits of removing the bolt area from the object patch is that crack-like edges are not falsely detected from the bolt area and also a threshold boundary for true crack edges can be made, which will be mentioned later in this section. Detection of the bolt on object patches is carried out by edge detection and binary morphology. The procedure is as follows: (1) A median filter is first applied to object patches to remove edges from real cracks or other textures, which are connected to the bolt. (2) Canny edge detector generates the edge image to detect boundaries of the bolt. Then, a binary edge map is obtained using a predetermined threshold (Canny, 1986). (3) Dilate operators using a disk structural element are applied to the edge map to fill gaps between edges and then, the convex hull of each connected entity is computed. Through this operation, several separated binary entities are generated from a bolt or background. (4) A true binary entity indicating the bolt is selected if it includes the center of the object patch. This criterion is reasonable because the object detector detects the bolt to be presented at the center of the window. (5) Finally, marginal pixels are added to the boundary of the detected entity so that it includes full appearance of the bolt. Example images are shown in Figures 4a and b. Figure 4a is the detected object patch from Sections 2.2 and 2.3. Following the above procedure, the area of the bolt in the object patch is detected in Figure 4b. Subsequent analysis is conducted on the area of the outside of the detected binary entity (object area).

The second step is detecting crack-like edges. In this study, the Hessian matrix-based vesselness measurement technique, called Frangi filter, is used to detect crack-like edges. The approach is based on the fact that a crack on steel has a thin and bright line, similar to the appearance of vessel in medical images. The

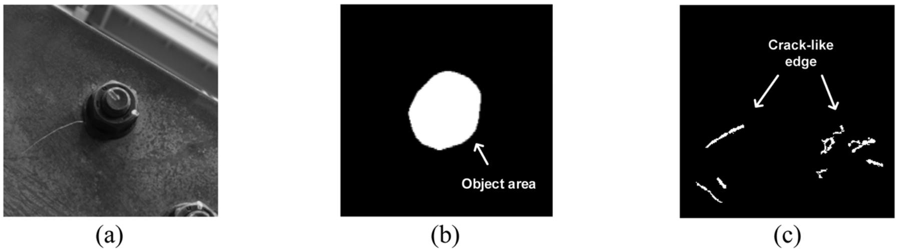

Fig. 4. Examples of bolt area and crack-like edge detection: (a) object patch including a bolt and its nearby area, (b) detection of a binary entity (white area) indicating bolt, and (c) crack-like edge detected from the outside of the object area using the Frangi filter.

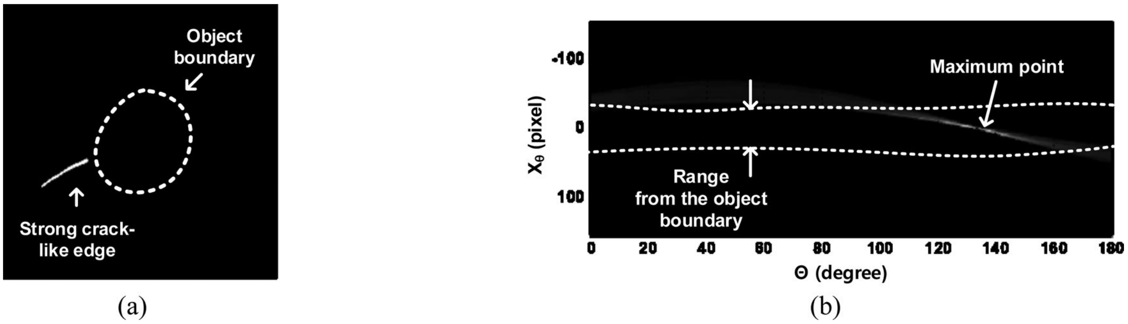

Fig. 5. Examples of crack detection using Radon transformation: (a) one strong crack-like edge with the object boundary and (b) Radon transformation of the strong crack-like edge and the range computed from the object boundary.

Hessian matrix-based edge detector does not produce double edges making good localization, and thus multi-scale crack detection is possible (Frangi et al., 1999). The brief outline of the Frangi filter is first, a Hessian matrix of the image is computed using a Gaussian derivative at multiple scales. Then, two eigenvalues (λ_1 and λ_2) of the Hessian matrix in each pixel of the image are derived ($|\lambda_1| > |\lambda_2|$). An ideal crack edge on image is $\lambda_2 \approx 0$ and $|\lambda_1| \gg |\lambda_2|$, and the sign of λ_1 is negative because crack is a bright edge. The strength of crack-like edges, V, is defined as

$$V = \begin{cases} 0 & \text{if } \lambda_1 > 0 \\ \exp(-\frac{R_B^2}{2\beta^2})(1-\exp(-\frac{S^2}{2c^2})) & \text{Otherwise} \end{cases}$$

where β and c are user-defined parameters, $R_B = \lambda_2/\lambda_1$, and $S = \sqrt{\lambda_1^2 + \lambda_2^2}$. For example, if the edge is close to the ideal crack edge, R_B goes to 0 and S becomes large, and thus V is close to 1, which is the maximum of V. The final edge map can be obtained by thresholding and removing edges connected to the border of the object patch. Figure 4c shows the result of crack-like edge detection from the outside of the bolt area in the object patch (Figure 4a). The technical details about the Frangi filter are provided in Frangi et al. (1999).

The remaining two steps are to detect real crack edges from non-crack edges using prior knowledge of crack's appearance because the Frangi filter detects several edges having similar appearance with cracks, as shown in Figure 4c. The third step is filtering out spurious non-crack edges using a region-based shape characteristic. All connected components in the binary edge image, obtained in the previous step, are identified using connected component labeling. Among these connected components, the crack-like edges are differentiated using their shape descriptors. For example, fat and short edges are not real crack edges. The shape descriptor used in this study is eccentricity, which is defined as the ratio of major to minor axes of a connected component and evaluate elongation of edges (Yang et al., 2008). A threshold, which represents the minimum eccentricity, is first set to filter out non-crack edges. Then, one strong edge line, which has the largest regional area, is detected, which is most likely a true crack. Figure 5a shows the detected edge line from Figure 4c using these two criteria.

As a final step, the detected crack-like edge, called strong crack-like edge, is evaluated as to whether it is the true crack or not. This decision is made based on the assumption that crack is initiated from the bolt hole, and has an almost straight line as mentioned in the previous section. Suppose that the axis perpendicular to the detected edge goes through the center of the object. When the object (or object boundary) is projected

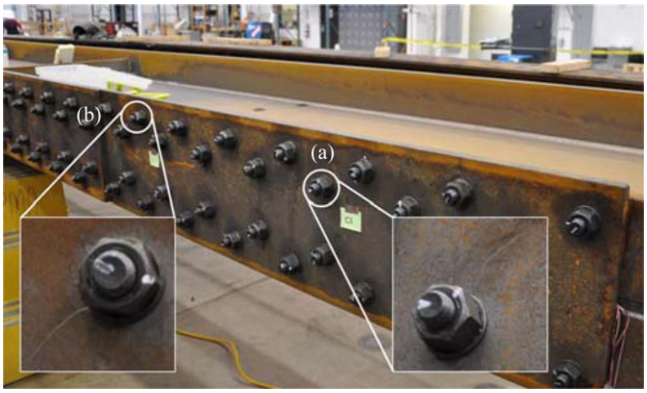

Fig. 6. A test steel beam including 68 bolts.

onto this axis, the range of the object projected on the axis can be computed. If the detected edge is the true crack, the line, which is drawn on the edge, will cross the axis within this range. This concept is successfully implemented using the Radon transform. The Radon transform in two dimensions is the integral projections of images along specified direction (MATLAB and Image Processing Toolbox, 2014; Deans, 1983). In this application, the maximum value of the Radon transformation image indicates the direction and position of the detected edge line. Figure 5a shows the detected crack-like edge and object boundary computed from Figure 4a. Radon transformation of the strong crack-like edge image is shown in Figure 5b. The dotted line is the range computed from projecting object boundary. The maximum value of the transformed image indicates the edge's direction and location on an angled axis, respectively. The axis angle of the maximum point in Figure 5b is around 140° and is perpendicular to the line of the edge. The true crack is determined if the maximum point is located inside of the range obtained by projecting object boundary onto this axis. This criterion may not perfectly classify all true cracks from crack-like edges because non-crack edges also exist in the similar position of true cracks. However, it can successfully remove most non-crack edges.

3 EXPERIMENTAL VALIDATION

3.1 Description of the experiment

A large scale, rusted I-beam having 68 bolts, as shown in Figure 6, is used for validating the proposed technique. As shown in introduction, a real fatigue crack is almost visually similar to a sharp scratch. Instead of producing real fatigue cracks on the beam, two artificial scratches are made with an awl at location (a) and location (b) in Figure 6. Test images of the beam are taken using a Nikon D90 camera with 18–105 mm lens, and zoom and flashing functions are not utilized. All 72 test images having 4,288 × 2,848 resolution are sequentially taken at roughly 2~3 m working distance, which is between the camera and beam, and do not have much tilt angles due to the issues in Section 2.2. For the object detection and grouping, downsized images with low resolution having 1,716 × 1,149 are used for fast computation. However, once the areas of object patches are detected, those are cropped from the full resolution images for accurate crack detection.

Five randomly selected test images among the 72 images are used for training. A square of 68 object patches are cropped from these images. One hundred forty-four negative patches, which are three times of a number of positive patches, are randomly cropped from the non-object areas. Five hundred Haar-like features in each object patch are randomly selected with respect to positions, sizes, and types. Thus, 5,500 features can be generated in each object patch, given by the product of 11 channel transforms and 500 Haar-like features. A "strong" classifier is trained with 200 weak classifiers, which are enough to ensure convergence of classification.

For the sliding window technique, rough estimation of the size of the object patches in the images in advance can reduce computation time and false-positive detection. For example, suppose that the size of object patches in image is unbounded, all scales of test images are scanned, which is a very time-consuming process. In this study, based on the physical bolt size and approximated working distance, the range of the bolt size on the downsized images is found to be 64~128 pixels. Consequently, the final scale is set to from 1 to 2 with $2^{1/5}$ scale steps, causing a total of 6 scales. For object detection, test images are downsized with these scales so that the different sizes of bolts in the images can be detected.

All algorithms proposed in this study are implemented using MATLAB. VLFeat, which is an open source library of computer vision algorithms, implements the SIFT algorithm with MATLAB interfaces (Vedaldi and Fulkerson, 2010). Each image makes pairs with the next 5 images in the set of sequential images. The matching criterion of keypoint descriptors proposed by VLFeat, called VL_UBCMATCH, is used with a threshold of 2.5. If the number of matches is less than 30, the fundamental matrix of this pair of images is not computed and this pair of images is not considered further during object matching. Using matched keypoints of an image pair, a fundamental matrix for this pair is computed using normalized 8 point algorithm and RANSAC. The function "estimateFundamentalMatrix" in the computer vision system toolbox in MATLAB implements this algorithm (MATLAB and Statistic Toolbox, 2012). If the number of remaining

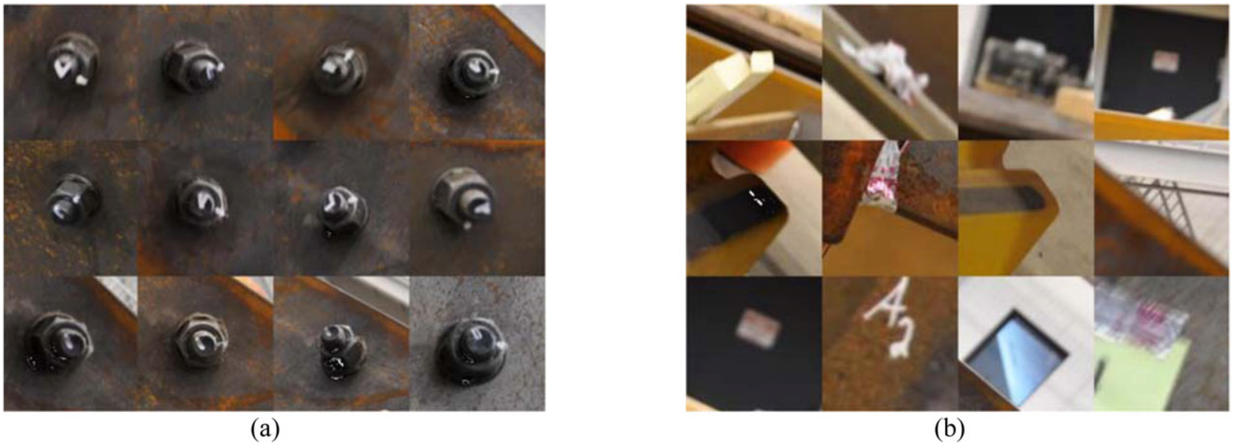

Fig. 7. Examples of bolt detection results from test images: (a) true detection of bolts and (b) false-positive detection.

matches is less than 25, this pair is also not considered for object matching. The threshold of the epipolar constraint for matching is set to 181 pixels, which is the diagonal distance of the maximum size of object patches. For grouping object patches, if a group includes less than three object patches, this group and its object patches are removed and excluded from a list of object patches. All object patches cropped from original test images are resized as 240 × 240 pixel. As a rule of thumb, a threshold for Canny edge detection is set to the high and low thresholds to products of 0.2 and 0.5 to the median value of the grayscale object patch, respectively. The size of the median filter for removing the crack edge for detecting object areas is a 5 × 5 square. A disk structure element having 8 pixel diameter is used to dilate images, and 10 pixels of the border margin are used. For the Frangi filter, parameters β and c are set to 0.5 and the maximum Hessian norm, respectively, presented in the original work and the threshold is 0.90 (Frangi et al., 1999). Scales of the filter are up to 3 pixels with 0.2 pixel step, which means the edges less than 6 pixels are more enhanced. For the shape descriptor, the minimum eccentricity is set to 8. Note that all parameters and thresholds are not automatically determined. Users should tune the parameters for different uses based on ones that authors suggested in the training step and by gaining experience. We chose values used based on tests with an initial set of training images. However, this simple process is conducted just once, and then the selected values are used for future visual inspections without further tuning.

3.2 Experimental results

A total of 1,326 bolts are shown in all test images. Bolts having partial occlusion or distracted by foreign objects are removed before counting. The resulting object detector proposed in Section 2 achieves a 98.7% detection rate (1,310 object patches) and a 6.8% false-positive detection (91 non-object patches). The proposed object detection technique successfully attains a high rate of true detection and minimizes false-positive rate. Figure 7 shows samples of detected bolts and false-positive detection.

Table 1
Results of object detection and grouping

Object detection	*Object grouping*
# of true object patches: 1,326	# of matching: 2,922
# of true-positive detection: 1,310	# of object groups: 68
# of true-negative detection: 16	# of object groups: 72 (with 4 overlaps)
# of false-positive detection: 91	# of non-object groups: 5

Table 1 shows the results of object detection and grouping. Based on the proposed matching and grouping techniques, 2,922 connections (matching) between nodes (objects) are found and 77 communities (groups) are detected. All 68 bolts are successfully grouped and 5 non-object groups are produced, which are visually similar to bolts. Here, 5 non-object groups come from false-positive detection in the object detection step. Non-object patches are also classified as groups because they are consistently detected across images. To avoid this problem, these non-object patches are used for training as negative samples, not to be detected in future testing. Moreover, there are 4 communities overlapping, which means two

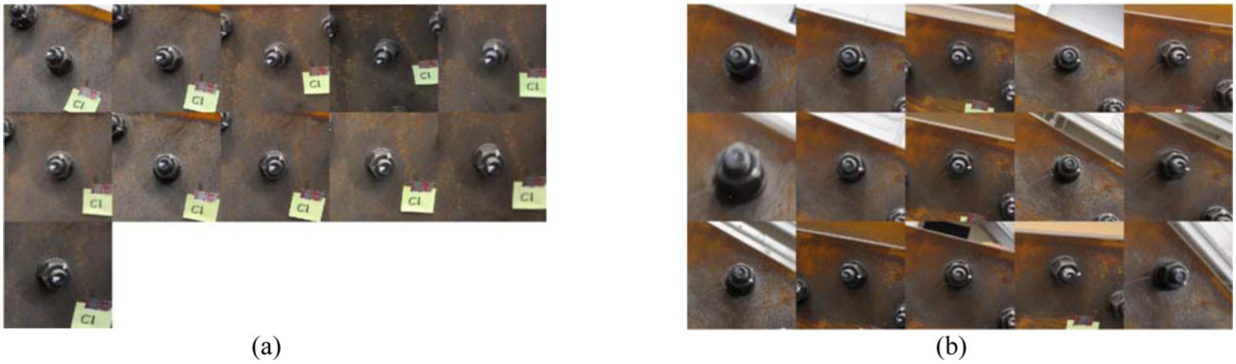

Fig. 8. Resulting groups of object patches at crack location (a) and location (b) in Figure 6.

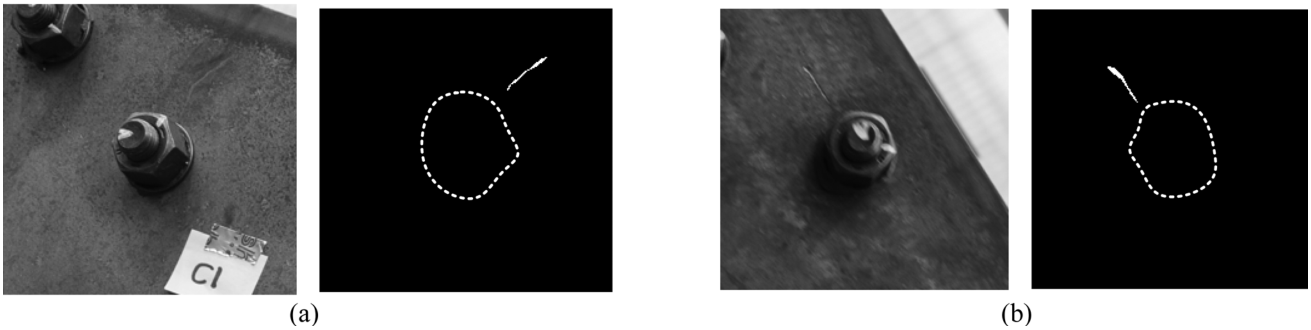

Fig. 9. Examples of crack detection results: (a) true crack detection and (b) false-positive crack detection.

communities indicate the same bolt. These discrepancies come from weak connections between sets of nodes in a same group, and could be overcome by increasing the number of pairs of each image for matching, but it would be computationally expensive. A total of 1,147 object patches from 77 communities are grouped and the remaining ones are removed due to a lack of nodes in their allocated group. Figure 8 shows groups of object patches having a crack associated with location (a) and location (b) in Figure 6. Detected object patches prove the necessity of images from different viewpoints. Only a couple of patches show clear cracks that are visually recognizable.

The proposed crack detection technique is applied to object patches in each group. There is a trade-off between true and false-positive detection depending on the threshold of the Frangi filter and Canny edge detector because these two parameters determine crack-edges and the decision boundary, respectively. However, regardless of these variations, cracks are constantly detected from at least one of the object patches in the groups showing location (a) and location (b). Example results are shown in Figure 9. Figures 9a and 5a represent object patches and crack detection with object boundaries from location (a) and location (b). The lines of true crack edges pass through the object areas. Figure 9b is one of the false-positive detection results. The machine generated scratch next to the bolt, which is not induced by authors, is visually similar to a real crack. However, it is not a true crack and was originally present in the test structure. This is a limitation of the proposed technique, but based on visual information only, this scratch would be challenging to differentiate from a real crack, even by human inspectors.

4 CONCLUSIONS

In this study, a vision-based crack detection technique is developed for automated inspection of large scale bridge structures only using images. Such images are collected from an aerial camera without advanced knowledge of the crack locations or special control of camera positions or angles. The study focuses on processing these images to identify the presence of damage on structure. The key idea is extracting images of damage sensitive areas from different angles so as to increase detectability of damage and decrease false-positive errors. To achieve this goal, object detection and grouping techniques, which are used in the area of computer vision, are implemented to extract images of possible damage regions. Using these images, the proposed damage detection technique can successfully detect cracks regardless of the small size or the possibility of their not being clearly visible depending on the viewpoint of the images. The effectiveness of the proposed technique is successfully demonstrated using images of a large scale, rusty, steel beam with cracks using a handheld camera instead of one mounted on a UAV. In the future, the proposed technique will be validated using images of real civil structures collected from a UAV.

ACKNOWLEDGMENTS

The authors acknowledge support from National Science Foundation under Grant No. NSF-CNS-1035748. The authors would like to thank Dr. Robert J. Connor and Matt Hebdon at Purdue for providing the test structure and invaluable comments for this study.

REFERENCES

Abdel-Qader, I., Abudayyeh, O. & Kelly, M. E. (2003), Analysis of edge-detection techniques for crack identification in bridges, *Journal of Computing in Civil Engineering*, **17**(4), 255–63.

Angel Aerial Survey (2014), Available at: http://angelaerialsurvey.com/, accessed May 14, 2014.

Canny, J. (1986), A computational approach to edge detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**(6), 679–98.

Clauset, A., Newman, M. E. & Moore, C. (2004), Finding community structure in very large networks, *Physical Review E*, **70**(6), 066111.

Dalal, N. & Triggs, B. (2005), Histograms of oriented gradients for human detection, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, June 20-25, 2005, Vol. 1, pp. 886–93.

Deans, S. R. (1983), *The Radon Transform and Some of Its Applications*, John Wiley & Sons, New York.

Dollár, P., Tu, Z., Perona, P. & Belongie, S. (2009), Integral channel features, in *British Machine Vision Conference (BMVC)*, London, England, September 5-8, 2009.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D. & Ramanan, D. (2010), Object detection with discriminatively trained part-based models, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**(9), 1627–45.

Frangi, A. F., Niessen, W. J., Hoogeveen, R. M., Van Walsum, T. & Viergever, M. A. (1999), Model-based quantitation of 3-D magnetic resonance angiographic images, *IEEE Transactions on Medical Imaging*, **18**(10), 946–56.

Friedman, J., Hastie, T. & Tibshirani, R. (2000), Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors), *The Annals of Statistics*, **28**(2), 337–407.

Hartley, R. & Zisserman, A. (2003), *Multiple View Geometry in Computer Vision*, 2nd edn., Cambridge University Press, New York.

Indiana Department of Transportation (2013), *Bridge Inspection Manual*, Indianapolis, IN.

Jahanshahi, M. R., Kelly, J. S., Masri, S. F. & Sukhatme, G. S. (2009), A survey and evaluation of promising approaches for automatic image-based defect detection of bridge structures, *Structure and Infrastructure Engineering*, **5**(6), 455–86.

Jahanshahi, M. R., Masri, S. F. & Sukhatme, G. S. (2011), Multi-image stitching and scene reconstruction for evaluating defect evolution in structures, *Structural Health Monitoring*, **10**(6), 643–57, DOI: 10.1177/1475921710395809.

Jahanshahi, M. R., Sami, F. M., Curtis, W. P. & Gaurav, S. S. (2013), An innovative methodology for detection and quantification of cracks through incorporation of depth perception, *Machine Vision and Applications*, **24**(2), 227–41.

Jiang, X. & Adeli, H. (2007), Pseudospectra, MUSIC, and dynamic wavelet neural network for damage detection of highrise buildings, *International Journal for Numerical Methods in Engineering*, **71**(5), 606–29.

Lee, B. J., Shin, D. H., Seo, J. W., Jung, J. D. & Lee, J. Y. (2011), Intelligent bridge inspection using remote controlled robot and image processing technique, in *Proceedings of International Association for Automation and Robotics in Construction*, Seoul, Korea, June 29–July 2, 1426–31.

Lowe, D. G. (2004), Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, **60**(2), 91–110.

Malekian, V., Amirfattahi, R., Rezaeian, M., Aghaei, A. & Rahimi, P. (2012), Automatic detection and localization of surface cracks in continuously cast hot steel slabs using digital image analysis techniques, *International Journal of ISSI*, **9**(1), 30–40.

MATLAB and Image Processing Toolbox (2014), The MathWorks, Inc., Natick, MA.

MATLAB and Statistics Toolbox Release (2012), The MathWorks, Inc., Natick, MA.

Miller, J. (2004), Robotic systems for inspection and surveillance of civil structures. Doctoral dissertation, The University of Vermont.

Moller, P. S. (2008), *CALTRANS Bridge Inspection Aerial Robot*, Final Report, CA 08–0182.

O'Byrne, M., Schoefs, F., Ghosh, B. & Pakrashi, V. (2013), Texture analysis based damage detection of ageing infrastructural elements, *Computer-Aided Civil and Infrastructure Engineering*, **28**(3), 162–77.

O'Byrne, M., Schoefs, F., Pakrashi, V. & Ghosh, B. (2014), Regionally enhanced multi-phase segmentation technique for damaged surfaces, *Computer-Aided Civil and Infrastructure Engineering*, **29**(9), 644–58.

Phares, B. M., Rolander, D. D., Graybeal, B. A. & Washer, G. A. (2001), Reliability of visual bridge inspection, *Public Roads*, **64**(5).

Schanda, J. (2007), CIE colorimetry, in *Colorimetry: Understanding the CIE System*, John Wiley & Sons Inc., Hoboken, NJ.

Shen, H. K., Po-Han, C. & Luh-Maan, C. (2013), Automated steel bridge coating rust defect recognition method based on color and texture feature, *Automation in Construction*, **31**, 338–56.

Snavely, N., Seitz, S. M. & Szeliski, R. (2008), Modeling the world from internet photo collections, *International Journal of Computer Vision*, **80**(2), 189–210.

Torok, M. M., Golparvar-Fard, M. & Kochersberger, K. B. (2013), Image-based automated 3D crack detection for post-disaster building assessment, *Journal of Computing in Civil Engineering*, **28**(5), A4014004.

Torralba, A., Murphy, K. P. & Freeman, W. T. (2004), Sharing features: efficient boosting procedures for multiclass object detection, in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington DC, June 27–July 2 2004, Vol. 2, II-762–69.

US Aerial Video (2014), Available at: http://www.usaerialvideo.com/, accessed May 14, 2014.

Vedaldi, A. & Fulkerson, B. (2010), VLFeat: an open and portable library of computer vision algorithms, in *Proceedings of the International Conference on Multimedia*, New York, NY, October, 1469–72.

Viola, P. & Jones, M. (2001), Rapid object detection using a boosted cascade of simple features, in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Kauai, HI, 2001, Vol. 1, 511–18.

Yang, M., Kpalma, K. & Ronsin, J. (2008), A survey of shape feature extraction techniques, in P.-Y. Yin (ed.), *Pattern Recognition Techniques, Technology and Applications*, InTech, pp. 43–90, DOI: 10.5772/6237.

Zhu, Z., German, S. & Brilakis, I. (2011), Visual retrieval of concrete crack properties for automated post-earthquake structural safety evaluation, *Automation in Construction*, **20**(7), 874–83.