Vision-based Automated Visual Inspection of Large-scale Bridges*

Chul Min Yeum and Shirley J. Dyke

***Abstract*— Visual inspection of bridges is extensively used to identify faults such as crack, corrosion, or deformation. However, current methods conducted by human inspectors demand long inspection time to cover large and difficult to access structures, and highly rely on inspector's subjective or empirical knowledge. To address these limitations, a vision based visual inspection technique is proposed by automatically processing and analyzing images collected by aerial cameras. First, aerial cameras gather sequential images of exterior of bridges with high resolution. Next, the object areas, which are susceptible to damage such as a bolt hole or joint, are detected from all collected images. Then, images of object areas are matched and grouped across images so that damage near objects can be inspected from many different angles. Finally, damage initiated from objects, is detected by processing those images. All these processes are designed to be fully automated. The effectiveness of the propose technique is investigated and validated through a large beam with array of bolts and artificial cracks.**

I. INTRODUCTION

Visual inspection is the primary method used to evaluate bridge conditions. Most decisions relating to bridge maintenance are founded on assessments from visual inspections. However, current visual inspection conducted by human inspectors includes several problems. The study, conducted by the US 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 two years [1]. The study found that there is a great discrepancy of results in case 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 regarded as careless or out-of-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 of the bridge should be remotely accessed and collected automatically by standardized manners. In literature, many researchers have proposed remote access image acquisition system for capturing images of under or over bridges [2-7]. A visual monitoring system was proposed by controlling several cameras mounted on bridges for collecting images [2]. Based on these images, the scenes of bridges are periodically constructed to evaluate the evolution of changes like crack or corrosion. The other approach is related with developing access equipment for improving accessibility. U-BIROS (Ubiquitous Bridge Inspection Robot System) proposed robotic image acquisition system, which scans bridges using a robotic arm equipped with cameras [3]. 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 [4]. The vehicle is capable of vertical takeoff and landing, translation to horizontal movement and orienting a video camera, controlled by the operating personnel on the ground. Recently, a multi-rotor helicopter is commercially used to wirelessly take pictures or videos for inspection of structures such as pipeline, power lines or dam's [5,6].

Next, once a bunch of visual information are collected from bridges, robust inspection techniques should automatically perform visual inspection tasks in the manual. This vision based autonomous inspection is not a new concept and already widely used for civil, mechanical or aerospace structures [8-10]. In this study, crack damage is focused because it is a common types of damage occurred in civil structure and it is visually clear in appearance. A complete review of vision based crack detection technique was conducted in following references [8,9].

Based on the full development of such individual technology, the futuristic visual inspection is motivated and imagined as follow. An unmanned aerial vehicle (UAV) equipped with high resolution cameras arrives a target bridge for inspection. Following preliminary designed flying paths, the UAV collects and records images. The flying path is periodically updated based on the previous inspection records, for example, taking more images of damage areas, which are detected in previous flights. The UAV transmits collected images back to the base station. In the base station, by processing large volume of these images, damage on the structure is detected, localized and quantified automatically without helping human inspectors. The system automatically generates an inspection report to help that expert visual inspectors make decision whether the bridge is needed to have further inspection or maintenance. Based on reports and decisions, the system would continue updating to be more robust and smart inspection of damage and reduce false alarms or misdetections.

*Resrach supported by National Science Foundation.
Chul Min Yeum is with the School of Civil Engineering, Purdue University, IN, 47906 USA (email: chulminy@gmail.com)
Shirley J. Dyke is with the Schools of Mechanical Engineering and Civil Engineering, Purdue University, IN, 47906 USA (email: sdyke@purdue.edu)

(a) (b)

Figure 1. Images of a fatigue crack on a steel beam: Two images (a) and (b) are captured from the same scene of the crack on the steel beam, but from different viewpoints. However, the fatigue crack can be seen from image (a) but not image (b). Note that the fatigue crack in the images is started from the tip of the notch near the bolt, which is artificially induced to simulate initial discontinuity of the circumference of the bolt hole.

This study was begun by exploring the question: Given a large volume of images from UAV, would it be feasible in a realistic structure detect damage using currently available image processing techniques? To find out the answer, multiple photographs of a steel beam having a fatigue crack initiated from one of bolt holes were taken and analyzed as a pilot study. As a result, we identified that two major issues that need to be considered for vision based automated inspection, which previous researchers may or may not have observed. First, searching cracks over the entire area of an image generates many false positive alarms and misdetections. In Fig.1 (a), there are many crack-like features such as structure boundaries, wires, or corrosion edges, causing either incorrect damage detection or a failure to detect real cracks due to its narrow width. However, detection of the real crack by human inspectors is not easy but still is possible. This is because they have already historical information about the crack. The historical information in this case includes that cracks on the steel structure have thin and shiny edges and are initiated from bolt holes. These features draw their attention to the bolts and their nearby area, facilitating damage detection. A second characteristic that was identified is, the crack may be invisible depending on viewpoint. Fig.1 (b) shows the same scene of the fatigue crack from different viewpoints. Comparing the white boxes in both images, Figs.1 (a) and (b), the crack on the second image is hardly observed. This concludes that images of same scenes from many different viewpoints are needed to detect damage without knowing how and where the crack is created. Most previous research, of course, has unconsciously considered these two issues only when they collect images through controlling circumstance such 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, direction and lighting direction cannot be known in advance, and also it is hard to precisely control camera positions and angles installed in UAV.

An attempt to consider above two findings builds the framework of the proposed technique. In this study, an automated crack detection technique is proposed using images collected under uncontrolled circumstances. Rather than searching cracks on whole images, the object of which area is susceptible to cracks, the bolt in this study, are first detected from images. This gives great benefits to increase detectability of cracks by narrowing down searching areas and damage scales in images. Object detection and grouping techniques used in computer vision areas are implemented to extract, match and group same objects from many angles across images. Crack-like edges, which have similar appearance of real cracks, are first detected from images of object areas using image processing techniques, and then based on historical information of cracks a decision is made whether crack-like edges are true cracks or not.

The major contribution of the proposed technique is to propose a new way of automated visual inspection using a large volume of images. Many previous researchers focused on detecting cracks from a few images which have visually clear cracks. However, if automated image acquisition system is used like aerial cameras, those favorable images are hard to be selectively obtained because of uncertainty of crack’s location and direction. Instead, the proposed technique starts from 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 it is small as well as false-positive alarm can be reduce by limiting searching areas. To the best knowledge of authors, there is no literature developing vision based visual inspection with this concept.

The remainder of this paper is organized as follow. Section II starts from the brief overview of the proposed approach and enters technical details about image acquisition, object detection and grouping, and damage detection. Experimental description and results are presented in Section III. Section IV includes the conclusion and summary.

II. PROPOSED APPROACH

The overview of the proposed technique is shown in Fig. 2. First, in Fig. 2(a), images of a structure from many angles are collected using image acquisition equipment. This may include all possible ways of image collection such as aerial cameras or inspection robots. Second, in Fig. 2 (b), target structural components called objects which are susceptible to crack damage, are detected and extracted from images. The object patch indicates an object and its nearby area where crack damage is probably generated. Third, in Fig. 2 (c), same object patches across images are matched and grouped. Finally, in Fig 2. (d), the proposed crack detection technique diagnoses that crack exists at the structural components. In this paper, 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. Overall, recent techniques in the computer vision area are implemented to increase qualities of object detection, object grouping and crack detection.

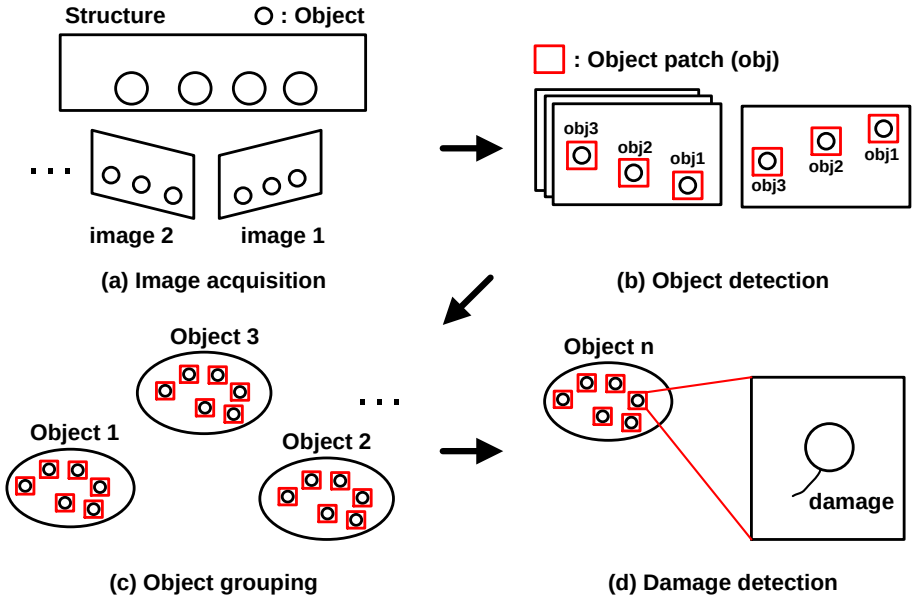


Figure 2. Overview of the proposed damage detection technique: (a) Acquisition of images from multiple angles of a structure, (b) Detection of object patches which may include damage, (c) Grouping of same objects across images, and (d) Detection of damage initiated from an object

A. *Image Acquisition*

For image acquisition, UAV flies under or over bridges by following a predetermined flying path, called as autopilot, and cameras installed capture scenes of bridges consecutively. Some guidelines of image acquisition are suggested here for the best performance of the proposed technique, but not requirement: (1) Highly-focused and high resolution images are captured under a good lighting conditions. Thus, consecutive still images are recommended rather than video streaming. (2) An angle between the camera and bridge, called tilt or perspective angle, should not be large. Since objects on the bridge are commonly located together with a short distance, they are overlapped each other on images under a large perspective angle, making object and crack detection difficult. However, angle variations of the camera is necessary due to angle dependency of crack's appearance on images. (3) Distance between the UAV and bridge stays constant. This is because the number of scale images need to be searched can be reduce based on the approximated this distance and physical bolt size. It brings about increasing object detection rate or decreasing false positive errors with low computation time. (4) GPS data of each image is recorded and saved for roughly estimating crack locations.

B. *Object Detection*

Object detection is challenging because an object's position, pose, scale, lighting, and backgrounds vary relative to camera angles and positions. The key of object detection is the selection of robust features that can uniquely represent the object without affecting above variations. There are no perfect solution that has a high level of perception like human but, many researchers have improved detectability of the object such as face or pedestrian [11-14]. In this study, some modifications of famous object detection algorithm are made for our purpose [11,13].

In this study, an integral channel based sliding window technique is applied over multiple scales of images [11,13]. The sliding window technique is that a fixed rectangular window slides on 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 detail of HSV and LUV color formats can be seen in the following reference [15]. HSV and LUV spaces can separate colors from brightness, which is not robust under lighting variation. Thus, V component in HSV and L component in LUV, which are brightness terms, are ignored from channel images. In order to consider the object shape, a gradient magnitude and HOG are used. These gradient features are proven in many application for object detection [12,14]. Fig. 3 shows channel images of the bolt. Channel images in a grayscale from Figs. 3 (b) to (I) are computed from the original RGB image, Fig. 3 (a). The intensity of these image vary depending on position or color and this variation represents the unique feature of the object patch. For example, Figs. 3 (g)-(l) shows different edges of the bolt are highlighted depending on gradient direction.

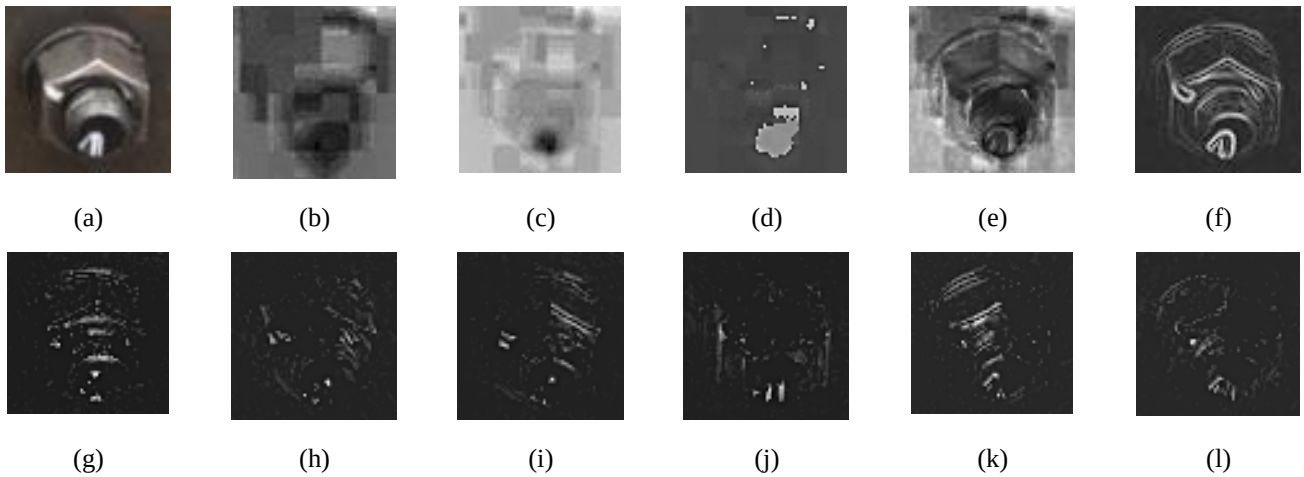


Figure 3. Example of channel images of a bolt: (a) original RGB image, (b, c) U and V components in LUV, (d, e) H and S components in HSV, (f) gradient, and (g-l) histogram of gradient with 6 orientations (0^o, 30^o, 60^o, 90^o, 120^o, 150^o)

Using these channel images, features of each window are computed by summing over local rectangular region using Haar-like wavelets [13]. 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 the original work of Viola and Jones [13]. 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 used 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 error can be obtained. There are several boosting algorithms, 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 [16,17]. The detail of the gentle boost algorithm and sample codes can be found the following website and reference [16,18].

C. Object Grouping

Object grouping in this study is a process of matching two or more same object patches across images and divide them as groups of same object patches. If wrong matching does not exist, matched object patches are simply assigned as a same group. However, in reality, spurious matching does not allow such simple division. Moreover, the object matching, especially for the application of this study, is much more difficult than conventional matching problems because all bolts in image have almost similar appearance and are closely located, causing fail to generate their 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 select the closet object to an epipolar line after finding a fundamental matrix of a pair of images [19-21]. However, in this application, a

single use of these techniques produces large error for 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 most non-corresponding object patches, but generally not all of them. The center of the object patch does not indicates the geometrical center of the object and the epipolar line, which is computed from the center of the object patch, does not exactly pass on the corresponding object patch's center. Even worse in this study is more than one object patches are close to a certain epipolar line due to geometrically proximity of bolts on the structure.

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, is matching with one of object patches in the second image. The object patch in the second image is searched by satisfying that 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 this criteria, keypoints inside them are matched with ones 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 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 integrating matching technique.

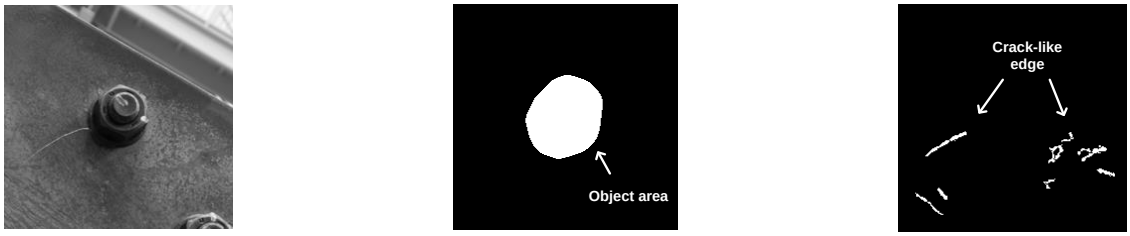
Despite of the use of the improved matching technique, not all object patches are correctly matched with same object across images. The final step is grouping object patches based on the matching between object patches. This structure is very similar to the community structure, which is widely used in data mining from large-scale data describing the topology of network such as social network. The object patches (= nodes) in a group (= community) have dense matching (= connection) internally but, some of nodes are shared with other nodes in other community 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 problem called, modularity maximization, is applied in this study [23]. 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 the pair of community to maximize modularity in network. The adjacent matrix A, which define the connection between nodes, is the only input to this system. The matrix A is defined as $A_{ij} = 1$ if nodes *i* and *j* are connected, otherwise 0. A_{ij} is the element at i^{th} row and j^{th} column. This matrix can easily be generated from the matching results. The technical details are provided in the paper [23].

D. Damage Detection

Based on groups of object patches that were found in the previous step, crack damage is achieved using the proposed crack detection technique. Before introducing the technique, the crack referred in this study should be first clearly defined. The crack has a very sharp edge and almost a straight line, and initiates from bolt holes on steel plates because the bolt hole inherently has initial discontinuity in its circumference. The detectable crack in images is the one can be detected by human vision. These "historical" information helps detecting the crack by filtering out non-crack edges. The proposed damage detection technique is applied to all object patches to decide whether cracks exist or not.

The first step in the proposed approach is to remove the bolt area from an object patch. The object patch found in the previous step includes both a bolt and its nearby area. The region of interest on the image is the area connected to the bolt, but not the bolt itself. By removing the bolt area from the object patch, crack-like edges falsely detected from the bolt area can be avoided and a threshold for true crack edges is decided, which will be mentioned in the later of this section. Detection of the bolt on object patches is carried out by edge detection and binary morphology. The procedure is listed as follow: (1) A median filter is first applied to object patches to remove edges from real cracks. (2) Canny edge detector generates the edge image to detect boundaries of the bolt while removing the "weak" edges from the surface texture of the structure. A binary edge map is obtained using predetermined threshold. (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 object patch. It is reasonable criterion because the object detector detects the bolt to be located at the center of the window. (5) Finally, marginal pixels are added so that full appearance of the bolt are included in the binary entity. Example images are shown in Figs. 4 (a) and (b). The Fig. 4 (a) is the detected object patch in Sections II.B and C. Following above procedure, the area of the bolt in the object patch is detected in Fig. 4 (b). Subsequent processing is conducted on the area of the outside of the detected binary entity.



(a) (b) (c)

Figure. 4 Example 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 the object (bolt), and (c) Crack-like edge detected from the non-object area using Frangi filter. The Frangi filter cannot extract only real crack edge (s).

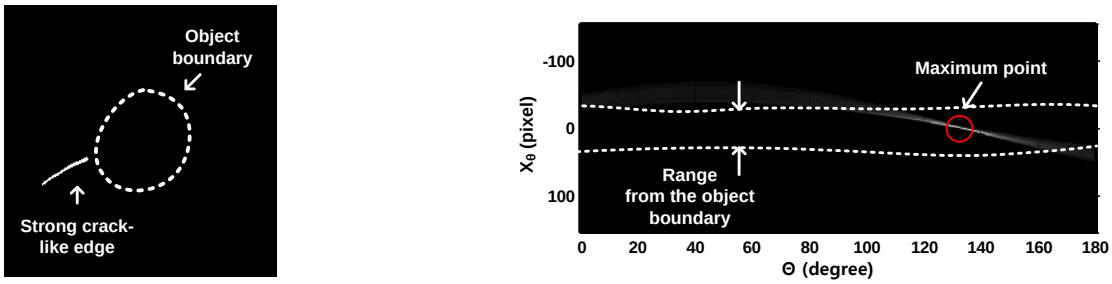
The second step is detecting crack-like edges. In this study, the Hessian matrix based vesselness measurement technique, called Frangi filter, is used to detect crack-like edges because a crack on steel which has a thin and bright line, is similar to the appearance of vessel in medical images, and the Hessian matrix based edge detector does not produce double edges making good localization, and multi-scale crack detection is possible [22]. The brief outline of the Frangi filter is first, a Hessian matrix of the image is computed using Gaussian derivative at multiple scales. Then, two eigenvalues, λ_1 and λ_2, of the Hessian matrix 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. The strength of crack-like edges, V, can be 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 becomes negative large and 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. Fig. 4 (c) shows the result of crack-like edge detection from the outside of the bolt area in the object patch, Fig. 4 (a). The technical details are provided in the paper [22].

The remaining two steps are to detect real crack edges from the non-crack edges using crack's historical information descriptors because the Frangi filter does not detect only the edge from crack, as shown in Fig. 4 (c). 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 or zigzag edges are not real crack edges. The shape descriptors used in this study are eccentricity, which is defined as the ratio of major and minor axes of a connected component and evaluate elongation of edges [28]. A threshold of the minimum eccentricity are set to filter out non-crack edges. One strong edge line, which has large regional area, is detected. Fig. 5 (a) shows the detected edge line from Fig. 4 (c) using this shape descriptor.

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 almost straight line 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 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 following the edge, will cross the axis within this range. This concept is successfully implemented using Radon transform. The Radon transform in two dimension is the integral projections of images along specified direction [24-25]. In the computer vision community, the Radon transform is used for line detection like Hough transform. The maximum value of the radon transformation image is the direction of line its position on angle axis. Fig. 5 (a) shows the detected crack-like edge and object boundary computed from Fig. 4 (a). Radon transformation of the strong crack-like edge image is shown in Fig. 5 (b). The dotted line is the range computed from projecting object boundary. The maximum value of the transformed image indicates edge's direction and location on rotating axis, respectively. The axis angle of the maximum point in Fig. 5 (b) is around 140 ^{o} 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.



(a) (b)

Figure 5. Example 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. The maximum point falling into this range indicates the strong crack-like edge is the real crack edge.

III. EXPERIMENTAL VALIDATION

A. Description of the Experiment

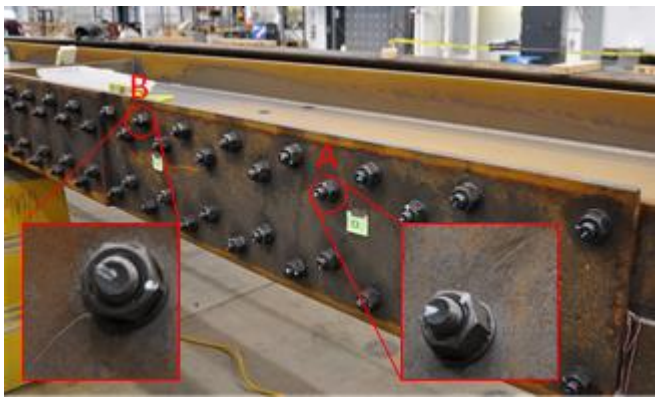


Figure 6. A test steel beam including 68 bolts: Two artificial cracks (locations A and B) are induced, which have similar appearance to a fatigue crack on the steel structure.

An I-beam having 68 bolts, as shown in Fig. 6, is used for validating the proposed technique. As shown in introduction, the real fatigue crack is almost visually similar to a sharp scratch. Instead of producing real fatigue cracks on the beam, two artificial crack are made with an awl at locations A and B in Fig 6. Test images of the beam are taken using a Nikon D90 camera with 18-105mm lens, and zoom and flashing functions are not utilized. All 72 test images having 4288 x 2848 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 II.B. For the object detection and grouping, downsized images with low resolution having 1716 x 1149 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.

Randomly selecting five test images among 72 images, these are used for training. A square of 68 object patches are cropped from these images to allow the object to be located at the center of each object patch. 144 negative patches, which are three times of a number of positive patches, are automatically cropped from the non-object areas. 500 Haar-like features in each object patch are randomly selected with respect to position, size, and type. Thus, 5500 features can be generated in each object patches, given by the product of 11 channel transform and 500 Haar-like feature. A "strong" classifier is trained with 200 weak classifiers, which is enough to ensure convergence.

For sliding window technique, estimation of size of object patches in 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, by considering the physical bolt size and rough working distance, the range of the bolt size on the downsize images is founded as 64 ~ 128 pixels. Consequently, the final scale is set to 2 and $2^{1/5}$ scale steps, causing a total of 6 scales. The images are downsized with these scales so that the different sizes of bolts in the images can be detected.

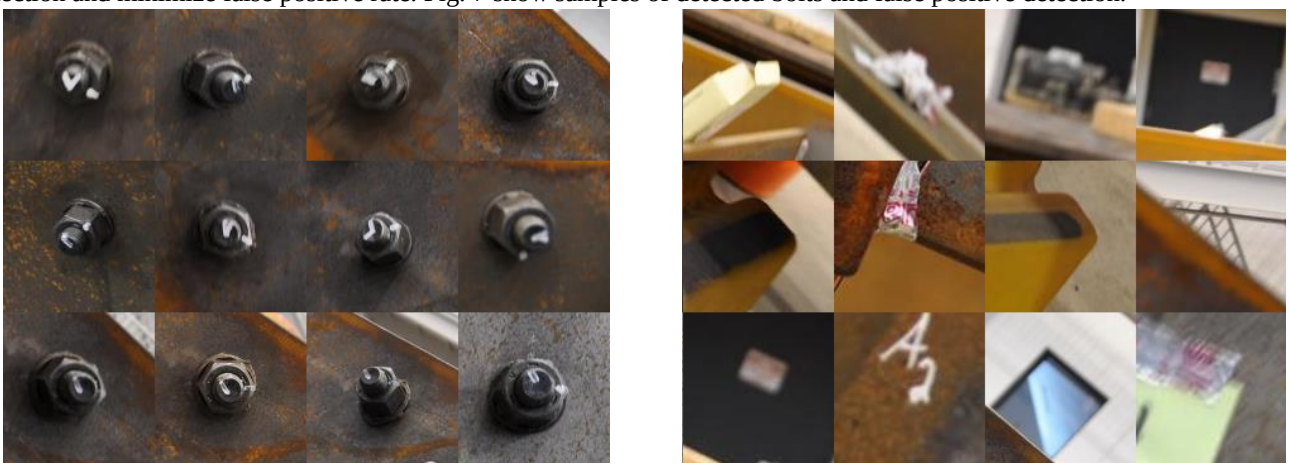
All algorithms proposed in this study are implemented using MATLAB. VLFeat, which is an open source library of computer vision algorithms, implements the SIFT algorithm with MATLAB interfaces [27]. Each image makes pairs with next 5 images in the set of sequential images. The matching criterion of keypoint 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 in object matching. Using matched keypoints of an image pair, a fundamental matrix for this pair is computed using a normalized 8 point algorithm and RANSAC. A function of 'estimateFundamentalMatrix' in computer vision system toolbox in MATLAB implements this algorithm [26]. The number of remaining matches is less than 25, this pair is also not considered in object matching. The threshold of the epipolar constraint for matching is set to 181 pixel, which is the diagonal distance of maximum size of object patches. For grouping object patches, if a group includes less 3 object patches, this group and its object patches are removed from consideration.

All object patches cropped from original test images are resized as 240 x 240 pixel. As a rule of thumb, a threshold for Canny edge detection is set the high and low thresholds to products of 0.2 and 0.5 with the median value of the grayscale object patch, respectively. The size of the median filtering is a 5 x 5 square. A disk structure element having 8 diameter pixel is used to dilate image. 10 pixels of the border margin are used. For Frangi filter, parameters β and c are set to 0.5 and the maximum Hessian norm, presented in the original work and the threshold is 0.90 [22]. Scales of the filter is up to 3 pixels with 0.2 pixel step, which means the edges within 6 pixels are more enhanced. For the shape descriptor, the minimum eccentricity is set to 8.

B. Experimental Results

A total of 1326 bolts are shown in all test images. The bolts having partial occlusion or distracting objects are removed in counting. The resulting object detector proposed in Section II achieves a 98.7% detection rate (1310 object patches) and a 6.8% false positive detection (91 non-object patches). The proposed object detection technique attains high rate of true detection and minimize false positive rate. Fig. 7 show samples of detected bolts and false positive detection.



(a) (b)

Figure 7 Example of bolt detection results from test images : (a) True bolt detection and (b) False positive detection

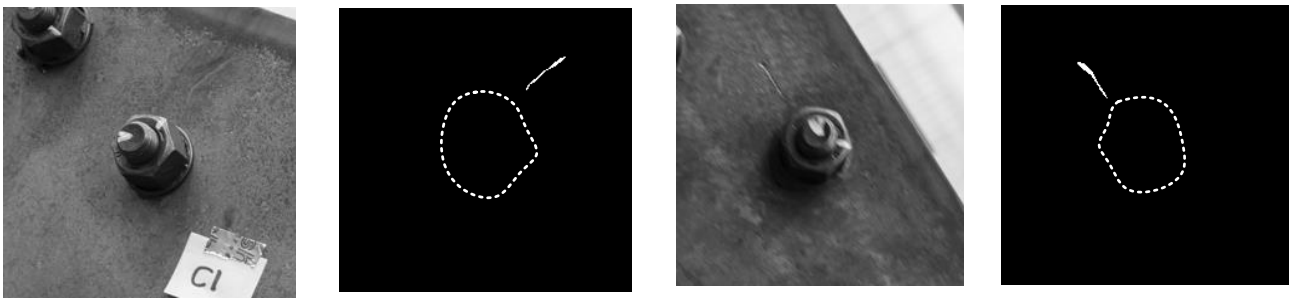
Based on the proposed matching and grouping techniques, 2922 connections (= matching) between nodes (=object patches) are found and 77 communities (= group) are detected. All 68 bolts are successfully grouped and 5 non-object groups are produced, which are visually similar to bolts. 4 communities are overlapped, which means two communities indicate the same bolt. This error comes from weak connections between sets of nodes in a group. This can be overcome by increasing the number of pairs of each image for matching but it is computationally expensive. A total of 1147 object patches are grouped and the rest ones are removed due to lack of nodes in their allocated group. Fig. 8 shows groups of object patches having a crack of which locations are A and B in Fig. 6, respectively.

(a) (b)

Figure 8. Groups of object patches having a crack: (a) and (b) are groups of object patches showing locations A and B in

Fig. 6, respectively.

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 variation, cracks are constantly detected from at least one of object patches in the groups showing the locations A and B. Example results are shown in Fig. 9. Fig. 9 (a) and Fig 5 (a) represent object patches and crack detection with object boundary from location A and B. The line of the true crack passes through the object areas. Fig. 9 (b) is one of false positive detection results. The scratch next to the bolt is visually similar to real crack but it was originally located in the test structure. However, this scratch is hard to be differentiated from real cracks, even human.



(a) (b)

Figure 9 Example of crack detection results: (a) True crack detection and (b) False positive crack detection.

IV. CONCLUSION

In this study, a vision based damage detection technique is developed for automated inspection of large scale bridge structures only using images. Such images are collected from an aerial camera without special control of their position or angle. 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 error. 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 region. Using these images, the proposed damage detection technique can successfully detect damage regardless of the small size damage or it not being clearly visible depending on the viewpoint of images. The effectiveness of the proposed technique is demonstrated using images collected from a steel beam with crack damages.

ACKNOWLEDGEMENT

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

[1] Phares, Brent M., et al. "Reliability of visual bridge inspection." Public Roads 64.5 (2001).
[2] Jahanshahi, Mohammad R., Sami F. Masri, and Gaurav S. Sukhatme. "Multi-image stitching and scene reconstruction for evaluating defect evolution in structures." Structural Health Monitoring 10.6 (2011): 643-657.
[3] Lee, Byeong Ju, et al. "Intelligent bridge inspection using remote controlled robot and image processing technique." International Association for Automation and Robotics in Construction proceedings (2011): 1426-1431
[4] Moller, Paul S. CALTRANS Bridge Inspection Aerial Robot Final Report. No. CA 08-0182. 2008.
[5] http://angelaerialsurvey.com/ (date last viewed 05/14/14).
[6] http://www.usaerialvideo.com/ (date last viewed 05/14/14).
[7] Miller, Jonathan. Robotic Systems for Inspection and Surveillance of Civil Structures. Diss. The University of Vermont, 2004.
[8] Jahanshahi, Mohammad R., et al. "A survey and evaluation of promising approaches for automatic image-based defect detection of bridge structures." Structure and Infrastructure Engineering 5.6 (2009): 455-486.
[9] Torok, Matthew M., Mani Golparvar-Fard, and Kevin B. Kochersberger. "Image-Based Automated 3D Crack Detection for Post-Disaster Building Assessment." Journal of Computing in Civil Engineering (2013).
[10] Abdel-Qader, Ikhlas, Osama Abudayyeh, and Michael E. Kelly. "Analysis of edge-detection techniques for crack identification in bridges." Journal of Computing in Civil Engineering 17.4 (2003): 255-263.
[11] Dollár, Piotr, et al. "Integral Channel Features." BMVC. Vol. 2. No. 3. 2009.
[12] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.
[13] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Vol. 1. IEEE, 2001.

[14] Felzenszwalb, Pedro F., et al. "Object detection with discriminatively trained part-based models." Pattern Analysis and Machine Intelligence, IEEE Transactions on 32.9 (2010): 1627-1645.
[15] http://en.wikipedia.org/wiki/Color_model/ (date last viewed 05/14/14).
[16] Torralba, Antonio, Kevin P. Murphy, and William T. Freeman. "Sharing features: efficient boosting procedures for multiclass object detection." Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. Vol. 2. IEEE, 2004.
[17] Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. "Special invited paper. additive logistic regression: A statistical view of boosting." Annals of statistics (2000): 337-374.
[18] http://people.csail.mit.edu/torralba/shortCourseRLOC/boosting/boosting.html (date last viewed 05/14/14)
[19] Lowe, David G. "Distinctive image features from scale-invariant keypoints." International journal of computer vision 60.2 (2004): 91-110.
[20] Snavely, Noah, Steven M. Seitz, and Richard Szeliski. "Modeling the world from internet photo collections." International Journal of Computer Vision 80.2 (2008): 189-210.
[21] Hartley, Richard, and Andrew Zisserman. Multiple view geometry in computer vision. Cambridge university press, 2003.
[22] Frangi, Alejandro F., et al. "Model-based quantitation of 3-D magnetic resonance angiographic images." Medical Imaging, IEEE Transactions on 18.10 (1999): 946-956.
[23] Clauset, Aaron, Mark EJ Newman, and Cristopher Moore. "Finding community structure in very large networks." Physical review E 70.6 (2004): 066111.
[24] MATLAB and Image Processing Toolbox 2014a, The MathWorks, Inc., Natick, Massachusetts, United States.
[25] http://en.wikipedia.org/wiki/Radon_transform (date last viewed 05/14/14)
[26] MATLAB and Statistics Toolbox Release 2012b, The MathWorks, Inc., Natick, Massachusetts, United States.
[27] Vedaldi, Andrea, and Brian Fulkerson. "VLFeat: An open and portable library of computer vision algorithms." Proceedings of the international conference on Multimedia. ACM, 2010.
[28] Yang, Mingqiang, Kidiyo Kpalma, and Joseph Ronsin. "A survey of shape feature extraction techniques." Pattern recognition (2008): 43-90.