Original Article

SHM Structural Health Monitoring

Structural Health Monitoring
1–15

DOI: 10.1177/1475921718765419
journals.sagepub.com/home/shm
SAGE

Automated region-of-interest localization and classification for vision-based visual assessment of civil infrastructure

Chul Min Yeum[1], Jongseong Choi[2] and Shirley J Dyke[1,2]

Abstract

Complementary advances in computer vision and new sensing platforms have mobilized the research community to pursue automated methods for vision-based visual evaluation of our civil infrastructure. Spatial and temporal limitations typically associated with sensing in large-scale structures are being torn down through the use of low-cost aerial platforms with integrated high-resolution visual sensors. Despite the enormous efforts expended to implement such technology, practical real-world challenges still hinder the application of these methods. The large volumes of complex visual data, collected under uncontrolled circumstances (e.g. varied lighting, cluttered regions, occlusions, and variations in environmental conditions), impose a major challenge to such methods, especially when only a tiny fraction of them are used for conducting the actual assessment. Such difficulties induce undesirable high rates of false-positive and false-negative errors, reducing both trustworthiness and efficiency in the methods. To overcome these inherent challenges, a novel automated image localization and classification technique is developed to extract the regions of interest on each of the images, which contain the targeted region for inspection. Regions of interest are extracted here using structure-from-motion algorithm. Less useful regions of interest, such as those corrupted by occlusions, are then filtered effectively using a robust image classification technique, based on convolutional neural networks. Then, such highly relevant regions of interest are available for visual assessment. The capability of the technique is successfully demonstrated using a full-scale highway sign truss with welded connections.

Keywords

Visual assessment, image localization, structure from motion, damage detection, convolutional neural network

Introduction

Changes in the appearance of a structure often provide obvious warning signs that a structure's condition is deteriorating. Thus, visual evaluation, the process of understanding the condition of a structure based on information that originates from visual data, remains the predominant means of assessing infrastructure systems as they gradually degrade over their lifetime.

Currently, visual evaluation procedures require that human engineers are on-site and actively involved in at least one or more of the steps, including observation, data collection, data analysis, or decision making. For example, in bridges, routine inspection must be conducted every 24 months in the United States and requires a complete walk-around visual inspection of all components of structure to determine their physical and functional conditions. In fact, human observation from no greater than an arm-length distance is required with present guidelines for certain structures.[1] However, such human-oriented visual evaluation does have certain limitations. First, it is expensive and time-consuming. Civil structures are often large and complex and may be situated in a harsh environment. Thus, there are often challenges in reaching and accessing many critical regions for viewing the structure. Indirect

[1]Lyles School of Civil Engineering, Purdue University, West Lafayette, IN, USA
[2]School of Mechanical Engineering, Purdue University, West Lafayette, IN, USA

Corresponding author:
Chul Min Yeum, Lyles School of Civil Engineering, Purdue University, West Lafayette, IN 47906, USA.
Email: chulminy@gmail.com

costs are attributed to the need for restricting or closure of vehicle lanes. Direct costs can involve expensive equipment and the staff to operate this specialized equipment, such as construction lifts or scaffolding. Such equipment is often required to reach difficult-to-access regions of these structures, and when heights are involved, there is an associated risk to human inspectors.

Next, human-oriented visual evaluation is inconsistent and prone to errors in the analysis.[2] The data are still manually organized, analyzed, and recorded. There is potential for faulty evaluation due to the reliance on an engineer's subjective, qualitative, or empirical knowledge and the following manual tasks that may produce erroneous reports and documents. Harnessing the capability to streamline these procedures using computers and automation can improve both efficiency and accuracy.

Revolutionary new image–based sensors and sensing platforms offer the opportunity to perform more efficient and cost-effective visual inspection. Smaller, cheaper, and more powerful visual sensing capabilities will enable greater granularity in both time and space. Thus, vision sensors can readily be incorporated into a variety of sensing platforms to fulfill essential tasks. For example, commercial unmanned aerial vehicles (hereafter, UAVs) have expanded sensor mobility from the ground to the air and can rapidly collect a large volume of high-resolution images suitable for performing visual evaluation. With such availability, these new sensors and sensing systems are starting to compete with human visual inspection regarding cost and performance.

Unfortunately, the actual implementation of these technologies is extremely challenging and has thus been limited to date, especially for large scale of civil structures. The major barrier in realizing the goal of automated visual evaluation is that the set of images collected with autonomous platforms are filled with irrelevant content. The images are typically collected from uncontrolled viewpoints and positions. Unless they are preprocessed before applying the established vision-based damage detection algorithms (reviewed in Section "Literature review") on these large sets of images, a high number of undesirable false-positive and false-negative errors are likely to occur. This outcome would significantly reduce the reliability and usability of the techniques, which was meant to be improved through automation. Preprocessing techniques that localize relevant and favorable areas from a large set of images must be incorporated into the procedure to achieve robust and efficient evaluation.

In this study, region-of-interest (ROI) localization and classification techniques are developed and incorporated to enable realization of automated vision-based visual assessment. An ROI is the portion of an image that contains the region of the structure that is targeted for visual evaluation, which is denoted as the targeted region for inspection (TRI). Damage-sensitive components or areas of a structure would be assigned as the TRIs. The ROI on each image is first automatically localized based on the geometric relationship between the collected images and TRIs using structure-from-motion algorithm (SfM).[3] However, depending on the viewing angle, the ROIs in certain images may not be favorable for visual evaluation due to occlusions that impede the view of the TRI. To address this issue, here a robust image classification technique, using a convolutional neural network (CNN) algorithm, is further implemented to eliminate the useless ROIs. With such a binary occlusion classifier (BOC), only the valuable portions of certain images are retained for enabling efficient and reliable vision-based visual inspection. The technique developed herein is demonstrated through performing visual inspection of welded connections in a full-scale highway truss structure.

The major contribution of this study is to develop a technique to facilitate the real application of existing damage detection algorithms with large volumes of actual images collected using a commonly available camera and sensing platform, such as UAVs. Large numbers of images or video may readily be acquired for visual inspection purposes. However, despite the enormous potential of this technology, automated processing is just not possible at this time. Manual viewing, sorting, and analysis are costly and inconsistent and prone to human error. Past research has been successful in automating the analysis of individual images for damage detection (reviewed in Section "Literature review"). However, to apply these well-established methods, preselected and localized images are needed that are known to contain an ROI. Instead, the technique developed herein begins by searching a large set of images to extract preselected damage-sensitive areas of the structure. By identifying, extracting, and classifying these regions (ROIs) from many different viewpoints, the detectability of damage can be dramatically increased even if it is small, and false positives can be reduced by limiting and regularizing the search areas. This technique will be a key enabler in the automated visual evaluation, breaking down the existing barriers that have impeded the use of large volumes of complex images.

The ROI localization technique was initially introduced by the authors in Yeum et al.[3] Although the underlying concept remains similar, the technique discussed herein can outperform the previous approach regarding efficiency and robustness and is more suitable for evaluating complex and large-scale structures for several reasons. First, images are collected only around

the TRIs for visual evaluation. In the previous work, a large volume of images was collected, covering the entire target structure (not only the TRIs) for constructing a full three-dimensional (3D) point cloud model. Then, this model was transformed to an existing geometric model (e.g. 3D structural drawing). However, it is inefficient and perhaps impossible to cover an entire large structure with close range images and can produce significant errors in the ROI estimation if the as-built condition of the structure is different from its known geometric model. Alternately, the method developed here only requires collection of images in the vicinity of each TRI, which are automatically registered to the baseline model to obtain the necessary geometric information for ROI localization. Consequently, once the baseline model is initially constructed, manual operation is no longer required throughout the entire process. Additionally, we develop and implement a novel image classification technique to filter out less useful ROIs and facilitate more robust visual evaluation. This method improves the robustness of the automated technique by incorporating a BOC to classify favorable and useful ROIs from a mixture of useless ROIs including occluded and erroneously estimated ones. The capability of the technique is successfully demonstrated using a full-scale structure in an outdoor environment.

The remainder of this article is organized as follows: Section "Literature review" begins with the review of current vision-based visual evaluation techniques. A brief overview of the proposed approach is presented in Section "Overview of the technique." In Section "Construction of a baseline model for the target structure," as a preliminary step, the process of building a baseline model is explained. Section "ROI localization and classification for visual inspection" provides the technical details behind how ROIs are localized and classified from a new set of collected images. In Section "Experimental verification," the proposed approach is demonstrated to be effective for visual inspection on welded connections using a full-scale highway sign truss. The summary and conclusions are given in Section "Conclusion."

Literature review

Vision-based autonomous visual inspection is not a new concept. Methods have been broadly developed and applied to civil, mechanical, or aerospace structures. In the past, many researchers have framed vision-based visual inspection techniques as the automatic implementation of the specific tasks outlined in the manual.[1] However, this series of steps is inefficient and, moreover, prone to error, when implemented without adjustment on a computer.

In civil infrastructure applications, the major tasks needed for visual inspection can be grouped based on the two most common materials used, concrete and steel. These two commonly used materials exhibit entirely different characteristics when it comes to damage. Defects in concrete typically manifest as a crack or delamination. Cracking often occurs during construction, settling, or operation. It can be the result of either a single factor or a combination of factors such as drying shrinkage, thermal contraction, restraint shortening, subgrade settlement, or applied load.[4] Thus, cracking in concrete is not necessarily a cause for concern but that decision should be left to an inspector's judgment. The appearance of a crack is mostly distinct with a lower intensity than the background and follows a straight or curved line with a relatively uniform width. Thus, intensity-based edge detection and segmentation approaches are widely used.[5,6] However, there are significant challenges to real-world implementation, including (1) similar appearance as that of other edges present, (2) connection of disjointed cracks detected, (3) scale estimation, and (4) image corruption due to environmental conditions, such as shadows or dirt. Various techniques have been proposed to overcome these challenges such as statistically learning to identify crack appearance for classification, quantification shadow removal, and connecting crack fragment.[7–10] Second, delamination, such as flaking or spalling, is another very different damage scenario that must be investigated with visual methods. Abrupt delamination damage, such as spalling or potholes, can pose risks to users. Such damage may also accelerate another mode of damage, such as corrosion of the steel rebar. Texture analysis and shape extraction techniques are used to extract potentially damaged areas in two-dimensional (2D),[11,12] and multi-view geometry is applied to obtain geometry information in 3D.[13,14]

Steel is a uniform solid material, and yet it is susceptible to environmental and operational conditions. Corrosion is a common source of damage in steel, causing material degradation or section loss. Corrosion appears as rust when there are uncoated, visible surfaces. Thus, color-based corrosion detection and texture-based corrosion have been widely studied.[15–17] Steel cracks, mainly fatigue cracks, are a second type of damage. They occur in areas of stress concentration and frequently originate from a flaw associated with a weld or material inconsistency. Detection of cracking in steel can be more difficult than in concrete because such cracks have thin, shiny edges and may even be invisible depending on lighting conditions and viewpoints. As with cracks in concrete, edge detection and segmentation techniques are used for detecting visibly clear cracks in a steel material. However, they do

require higher resolution images or must wait for the formation of larger crack sizes for ready detection.[18,19]

In this study, we do not focus on either developing a damage detection algorithm or improving these established techniques. Rather, our interest centers on providing ways to facilitate greater robustness in these existing techniques and thus enable their practical use. We aim to enable automated and efficient visual assessment when a large volume of images is automatically collected by UAVs. Clearly, when a large volume of images is collected from a structure that has many edges and lines, there will inevitably be a large number of false positives if every image is processed, regardless of the robustness of the method. Thus, we aim to remove the irrelevant regions from the image set prior to processing by defining ROIs in advance that are to be the focus of the inspection. The automated ROI localization and classification techniques developed in this study will overcome those challenges.

Overview of the technique

The technique is intended to achieve the visual assessment scenario proposed by the authors:[19] An UAV equipped with a high-resolution camera arrives at a candidate structure. Following a flying path designed a priori, the UAV automatically flies based on global positioning system (GPS), collecting and recording images near designated target areas from many viewpoints. Using those images, processing takes place on the large volume of images, and damage present in the structure is detected, localized, and quantified automatically without requiring the involvement of human inspectors. This information would provide evidence to facilitate better decision making related to repair and maintenance priorities for other structures. A key step that enables this scenario is to extract highly relevant and useful ROIs corresponding to the TRIs on the collection of complex images, supporting robust and efficient visual evaluation. This step is needed because regardless of the flying path designed a priori, each image will still include a large portion of irrelevant areas. This step is needed because regardless of the flying path designed a priori, each image will still include a large portion of irrelevant areas due to uncertainty in GPS data and occlusions. Thus, both irrelevant images and irrelevant portions of images must be filtered out in advance before implementing specific damage detection methods to avoid unwanted erroneous results.

The proposed approach first requires building a baseline 3D model of the target structure, in advance, using a large number of images (hereafter, baseline images) that cover the entire testing structure. This step is a one-time process, carried out at the beginning. Then, the resulting model will subsequently be utilized for visual inspection with newly collected images (hereafter, test images) during each inspection task over the years. During an inspection, a set of test images is collected around each TRI, and the ROIs on test images are localized by geometrically mapping the TRIs on the images. An image classification technique is further implemented to identify and extract the ROIs that are favorable for damage detection. Finally, a method to identify the damage type of interest is evaluated on only these highly relevant and visible ROIs. Each damage identification method would then be applied to each ROI found to be relevant.

In Figure 1(a), as a preliminary step, a baseline model is constructed using SfM from a large volume of images collected around a target structure. The baseline model consists of 3D points with descriptors and calibrated images.[20,21] The 3D points in this model and their descriptors are exploited to register the test images into this model. The TRI is defined by assigning new 3D point(s) in the baseline model through 2D point match on the calibrated images.[20]

For actual visual evaluation, the test images are first collected from near each TRI in the structure, as shown in Figure 1(b) and mentioned in the proposed visual evaluation scenario. The red blurred circles in the diagram indicate the selected TRIs for this structure. A large volume of the images is first collected from different viewpoints and locations around each TRI. Next, in Figure 1(c), the collected images are registered into the baseline model by matching the 3D points in the baseline model with the 2D features on the test images using their descriptors. Then, external (location and orientation) and internal camera parameters in each of test images are calibrated in the coordinate system of the baseline model. The geometric relationship between test images and each TRI in the baseline model is identified. Third, in Figure 1(d), each ROI is extracted by mapping of the TRI on to each of the test images. Since the size of the ROI is computed by geometrically projecting the TRI on the image, the scale (size of the TRI in the ROI) is identical, improving the performance of damage detection algorithms. However, the visibility of some portions of the ROIs may be hindered by occlusions due to the complex geometry of the structure. To filter out such undesirable occlusion ROIs, a robust image classification algorithm, based on CNN algorithms, is implemented in Figure 1(e). This BOC is used to distinguish between the nonoccluded and occluded ROIs and is trained using manually labeled ROIs obtained from the baseline model. Then, the trained BOC is applied to the ROIs extracted from test images. Finally, damage on the TRI is detected on the classified ROIs in Figure 1(f).

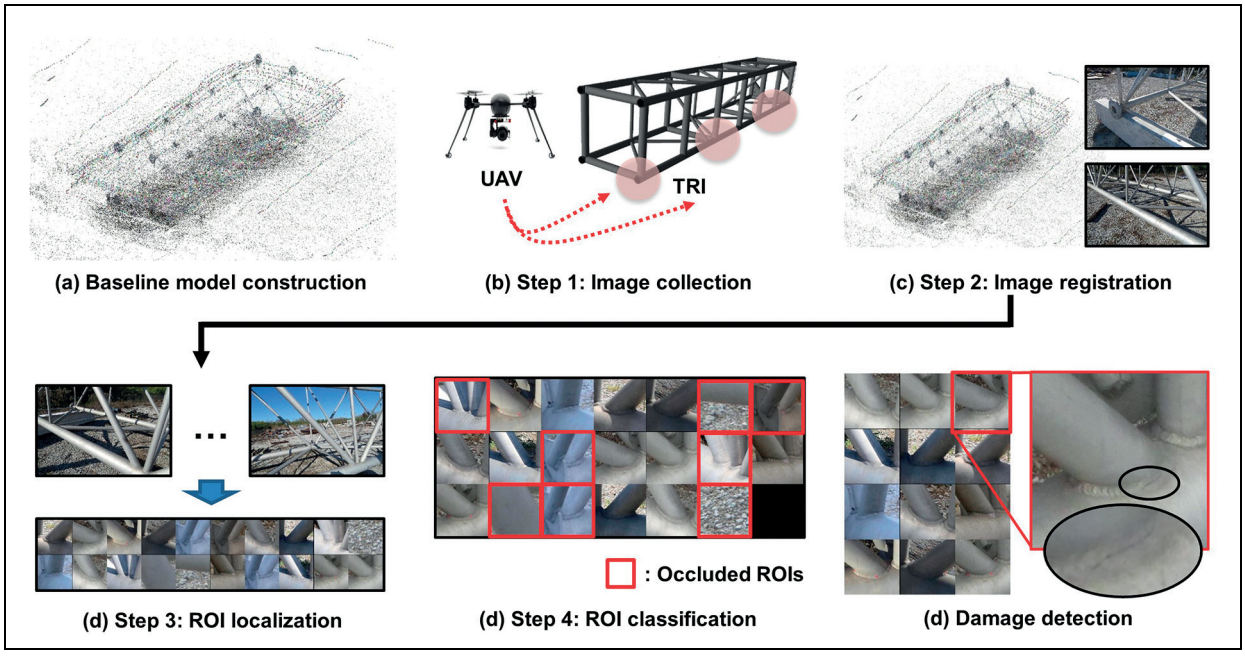

Figure 1. Overview of the technique developed: (a) construct a baseline model of target structure (preprocessing), (b) collect images around each of the targeted regions for inspection (TRIs; Step 1), (c) register the images into the baseline model (Step 2), (d) localize regions of interest (ROIs) on the images (Step 3), (e) classify the favorable ROIs (Step 4), and (f) detect (crack) damage on the ROIs (Step 5).

In this scenario, a large volume of test images collected from each of the TRIs turns into sets of ROIs corresponding to each of the TRIs. Users would apply the vision-based damage detection technique only to these filtered ROIs, which are preselected to be highly relevant and favorable for visual evaluation. We anticipate that any vision-based visual inspection technique can be facilitated by and integrated into the system developed here. Ultimately, this capability to localize and classify the ROIs will serve to strengthen the reliability of utilizing automated vision-based methods.

Construction of a baseline model for the target structure

In the last few years, SfM has led to major breakthroughs in performing 3D reconstruction of various types of physical scenes from images.[20,21] With this technique, a large number of uncalibrated images can be used to generate a scene of the structure. Compared to traditional photogrammetry methods, all calibration parameters and a point cloud are generated without the need for known control points or manual matching, enabling fully automated reconstruction. For civil engineering applications, SfM has been implemented for surveying applications and building information modeling.[22–24] Note that this baseline model construction is conducted only one time in the beginning and is subsequently used for damage detection on a set of test images collected from UAVs. Here, we will not be developing new SfM algorithms within this study. However, this study concentrates on the robust use of current SfM methods to establish a baseline model with high confidence.

SfM has become a well-established technology, and some popular commercial software packages have been built in conjunction with the development of UAV technology, such as Pix4D,[25] Photoscan,[26] or ContextCapture,[27] and noncommercial software, such as VisualSfM,[28,29] OpenMVG[30] (Moulon et al., 2012)[31], or Theia.[32] An image acquisition plan for successfully building such a model has been documented.[33] Regardless of the differences in functionalities and algorithms implemented in these tools, or their optimization for specific applications, in general, the input to SfM is a set of images, and the outputs are the associated projection matrices (constructed based on several internal and external camera parameters) and the 3D point cloud generated from the scene. Nearly all available commercial and noncommercial SfM software can export these outputs in a readable format.

Building a baseline model using SfM software is a process to define the TRI on the model and generate 3D points with descriptors for use in registering test images in the future. A large volume of baseline images with a variety of locations, orientations, and lighting

conditions is collected from the target structure. Each image produces many feature points and descriptors characterizing their appearance. The SfM software computes the 3D geometry of the scene by matching those features using their descriptors. The matched feature points across multiple images become 3D points when they are geometrically valid, and they are used for calibrating parameters in each image including a projection matrix and radial distortion coefficient(s).[20,21]

The output of the SfM is a projection matrix in each image (including radial distortion parameter(s)) and the 3D point cloud generated from the scene. The projection matrix of the image i is a 3×4 matrix, denoted as P_i, and the following relation holds

$$x_i = P_i X \quad (i = 1, 2, \ldots, n) \tag{1}$$

Here, n is the total number of participating images used for constructing the baseline model, X is arbitrary 3D points in the scene, and x_i is the corresponding point on the image i. These matrices are expressed in homogeneous coordinates.[20] Radial distortion should be corrected in advance so that all images follow the relationship in equation (1) based on the pinhole camera model. This relationship implies that *any 3D point in the model can be mapped to any image if its projection matrix is known. Thus, if the TRIs are* defined *using 3D points in this baseline model, and the projection matrices of the newly collected test images can be estimated, then the ROIs on the test images corresponding to the TRIs can be identified with the relationship in equation (1)*. The projection matrix of each test image can be computed by determining the match between 3D points in the baseline model and its 2D features, which will be illustrated in Section "Image registration."

To define the TRI, we use a virtual sphere that contains the corresponding TRI. This approach was developed in Yeum et al.[3] The virtual sphere is used for estimating bounding boxes for each TRI that appears in the images and tightly encloses the relevant portions of the image. Detailed theoretical derivations of the bounding box size and location have been established.[3] The virtual sphere is defined using only two parameters, a 3D center location and a diameter. In the past, these two parameters were obtained from the known geometric model, based on the assumption that the as-built structure has the same dimension. However, as mentioned in Section "Introduction," the technique developed herein eliminates the need to use an existing geometric model. This work focuses on the fact that this assumption may not be valid when it comes to large, complex structures. Instead, based on a triangulation relationship, new 3D points can be defined in the baseline model to compute the center of the virtual sphere and its diameter using more than two corresponding points on the baseline images. To match the same points across several images, temporary placement of a fiducial marker(s) on the structure is required. For example, in Section "Experimental verification," we use small red circular stickers, which are clearly visible on the images.

ROI localization and classification for visual inspection

Image acquisition

The way in which the test images are acquired will strongly influence their usefulness for successful completion of the subsequent steps in the technique and ultimately for damage detection. The conditions for image acquisition are outlined here to yield favorable test images for visual inspection. The most important guideline is to collect images in which the TRIs are clearly visible. Visibility of the TRIs on a subset of the test images is a vital prerequisite for successful implementation of the technique. Visibility of the TRIs on the test images is mainly affected by a couple of factors: working distance, motion blur, and occlusions. First, the working distance should be sufficiently close to guarantee sufficient resolution of the ROIs.[3] Although the ROI resolution increases as the working distance decreases, the UAVs should also not fly too close to the structure due to both the risk of collision and loss of the GPS signal. Areas can be magnified with an optical zoom of the camera, but this capability is also susceptible to motion blur. Thus, before flying the UAVs, the selection of the UAVs and the camera should take these issues into account. Several system capabilities should be carefully designed and considered, such as the parameters in the camera itself (e.g. sensor size, pixel resolution, or focal length), the variations in an angle between the TRIs and the camera, and the required resolution of the ROI for the damage to be detected.

Next, motion blur is an effect that occurs when an object being recorded moves relative to the camera during the exposure time of the camera. Unwanted vibration of the UAV platform transmitted to the camera and/or a fast flying speed under a close working distance will likely produce blurry images. These problems can be alleviated by stabilizing sudden movements using a multi-axis gimbal and by configuring a fast shutter speed.[33] However, increasing the shutter speed may reduce the quality of the image. Thus, taking pictures with good weather (lighting) conditions is recommended to obtain high-quality images regardless of the shutter speed of the camera. Note that having some blur in a small portion of the image set is not critical in the actual implementation. They will be automatically rejected in the registration process (Section "Image registration")

because they have insufficient features for matching, and ROIs will not be generated in the subsequent process.

Occlusions represent an unavoidable corruption mode, frequently occurring in a structure with complex geometry. They occur when structural components impede the view of the TRIs at certain camera locations and angles. To mitigate this problem, either the angles between the camera direction and the TRIs should be small, or more images should be collected from a variety of viewpoints. Of course, including several angle variations in the image collection is also important for successful vision-based visual inspection.[19] Thus, the recommended strategy is to take several pictures with different angles at preset GPS locations programmed into the UAV path. Thus, all relevant regions are included in the image set regardless of possible GPS error. Further processing is then required to filter out occluded ROIs using an image classification technique implemented in this study (Section "ROI classification").

Image registration

Image registration is a process to estimate a full 6-degree-of-freedom (3 for the rotation and 3 for the position) camera pose (extrinsic matrix) and intrinsic camera parameters including focal lengths and principal points (intrinsic matrix), represented by a projection matrix.[35] Information about camera pose plays a crucial role in general scene understanding, such as where the images were taken or how the 3D scene geometrically relates to the images. Thus, this is, as one of the key problems in computer vision, used across a broad range of applications including robot pose estimation indoors where GPS is not available and augmented reality.[35–37] In the computer vision community, image registration is also referred to as camera calibration, camera resections, or image localization, depending on the application.

As mentioned in Section "Construction of a baseline model for the target structure," the projection matrix enables mapping of any 3D points in the scene to the corresponding 2D points on each image. The baseline model provides the 3D location of each TRI (precisely, the location of the center and the diameter of the virtual sphere encompassing the TRI) and 3D points with descriptors. By matching several 3D points in the baseline model with 2D features on the test images, each image can be registered in the baseline model. However, comparing all individual matches between massive 3D points and 2D features in each image is quite time-consuming, inefficient, and can produce incorrect results. Thus, this brute force approach becomes a computation bottleneck, as well as a significant source of error, when the size of the baseline model (the number of 3D points) is large.

In this study, we reduce the number of 3D points to find a rapid and accurate match with 2D features that appear in the test images. Because in practice the test images are mainly collected near a specific TRI to extract the corresponding ROI, the test images that clearly show the TRI may be registered using only a subset of 3D points. In Section "Overview of the technique," for the learning of the classifier, the ROIs of each TRI are prepared from the baseline images. The baseline images that include such valid ROIs are captured with an appropriate position and angles, producing useful 3D points that can support registering the test images. Not all baseline images generate valid ROIs corresponding to the TRI. Thus, the 3D points generated from "multiple" numbers of those images in the baseline model are only used for registering the test images. For example, in Section "Experimental verification," the baseline model is constructed using 700,000 3D points, which is a massive number of points. For Weld 1, 807 baseline images among 4630 produce valid (positive) ROIs. The 3D points of which participating 2D features are generated from the match between more than three of those images are used for the matching process. In this case, the 700,000 3D points are reduced to less than 100,000 for implementation, which dramatically increases the speed of the matching procedure.

Once we find the matches between the 3D points in the baseline model and the 2D features on each of the test images, the parameters in the projection matrix as well as a lens distortion coefficient are calibrated. First, the initial parameters for the intrinsic matrix are approximated from the EXIF data. The principal points, x and y, are initially set to half of the image size in width and height directions, respectively. The focal length can be approximated using the focal length and sensor width that correspond to the camera model (Moulon et al., 2012).[30] Next, based on the initial estimation of the intrinsic matrix, the camera pose (external parameters) is estimated using a Perspective-n-Point (PnP) algorithm.[38] However, PnP is prone to errors caused by outliers in the set of 2D–3D matches. RANdom SAmple Consensus (RANSAC) can be adopted to filter out spurious matching and estimate the camera pose in a way that is robust to outliers. Finally, all parameters for the projection matrix as well as a single lens distortion parameter are refined using bundle adjustment. The Levenberg–Marquardt (LM) algorithm has been used to minimize the reprojection errors to refine those parameters.[38]

ROI localization

Using the prior steps, the projection matrix of each test image is computed in the coordinate system of the baseline model. The virtual spheres encompassing the TRIs

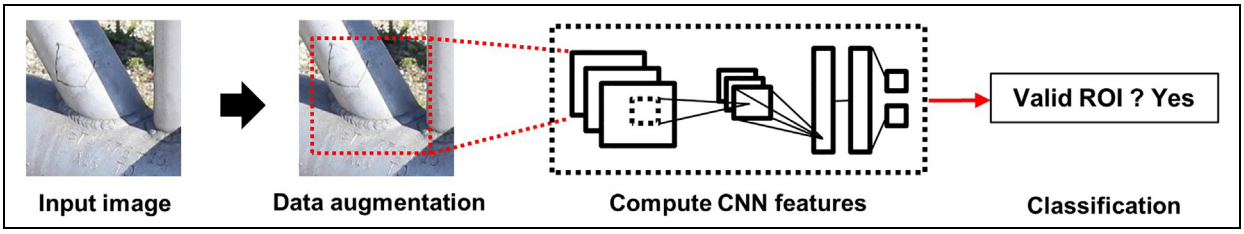

Figure 2. Overview of image classification using convolutional neural networks (CNNs) for binary classification.

can be mapped to each of the test images so that the corresponding ROIs on the test images are localized. The detailed theoretical derivations of the bounding box size and location on the test images are provided by Yeum et al.[3] At the end of this process, the ROIs in each test image that contain a complete view of the TRIs are obtained.

ROI classification

Although relevant ROIs on the test images are extracted using the prior steps, the registration process relies on an entirely geometric relationship. Thus, a portion of the ROIs extracted from the test images will inevitably contain occlusions and will be of limited use for automated damage detection. The performance and robustness of damage detection methods are dramatically reduced when there are unexpected changes in the visual appearance of the region of the structure in the images, for instance, if they have partial occlusions. Thus, a technique is required to eliminate ROIs that are of limited use due to occlusions to enable reliable analysis and confidence in the inspection results. To extend the analogy to human-based visual evaluation, while the previous steps are intended to mimic the action of reaching and gazing at the TRI, this process is intended to mimic the process of moving around to search for good viewpoints where the TRI is clear visible.

In this study, we utilize CNNs to filter out the unwanted, occluded images. In the last few years, CNNs have been exploited to advance several computer vision applications and have enabled the learning of high-level and deep features for image recognition using large-scale databases.[39,40] CNNs typically have one or more convolutional layers tied with weights and pooling layers to extract scale, translation, and rotation tolerant features and fully connected layers associated with these features to classify images or object categories. Conceptually, CNNs work by finding features that best describe the given images with a large number of convolutional filters. Successful work with CNNs resulted from training with a large number of images, including a large number of parameters, using "dropout" regularization, and exploiting graphics processing units (GPUs) for implementation. Several CNN architectures have been introduced in the literature, but their accuracy varies depending on how one configures the network architecture. Optimal network architectures and the configuration of input images and categories are still a topic of active research in different domains of applications. However, it is evident that CNNs provide exceptional performance in image classification.[41]

An overview of our approach to image classification using CNNs is provided in Figure 2. We implement two-class (binary) classification to train a complex discriminative boundary between nonoccluded (valid) ROIs (positive) and the rest (negative) which are corrupted by occlusions. For training this BOC, the ROIs are first extracted from the baseline images in the same way that they are extracted from the test images. Then, positive and negative ROIs are prepared by manually labeling them. For this step, it is critical to establish a clear rule for dividing positive and negative ROIs to maximize the visual discriminative boundary between them. The square center region on each ROI is cropped and resized to be slightly larger than the input size of the image for CNNs. Depending on the resolution of the original image, the ROIs may be resized or stretched, but typically, the input of the CNNs requires low resolution (e.g. 200–300 pixels on each side depending on algorithms).[40] Then, the resized ROIs are augmented to avoid overfitting with respect to translation, color, or light variations. Augmentation produces more training images without adding new original images. For example, in AlexNet, used in this study, all images are resized to be 256×256 pixels, and random square regions of 224×224 pixels are extracted.[40] Once the inputs for the training of the CNN are prepared, a large number of parameters at convolutional, pooling, and fully connected layers are automatically tuned so as to extract robust features by minimizing the error in estimating true labels of the training images. Stochastic gradient descent with a batch size of images is used to optimize the parameters in the network. At each epoch, a batch is assigned at each iteration using randomly

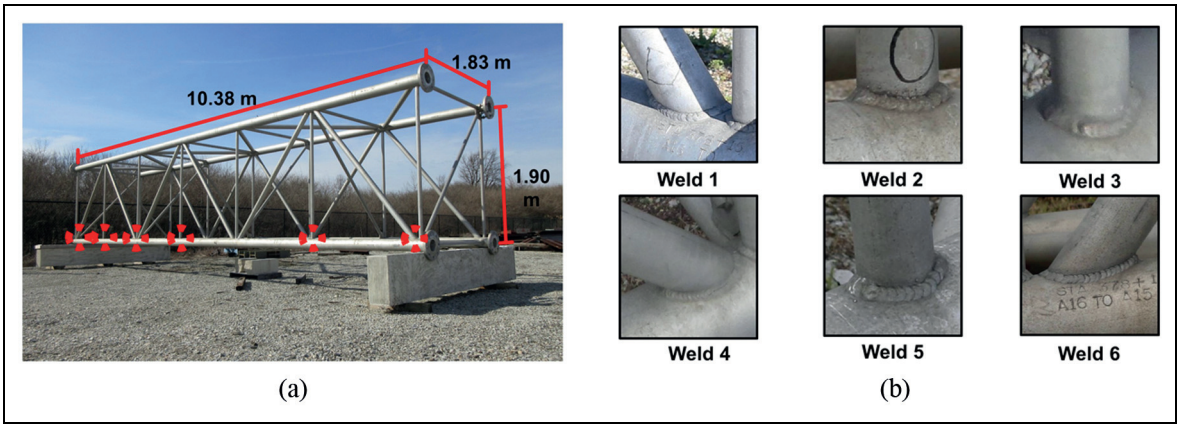

Figure 3. Description of a full-scale highway sign truss structure: (a) dimensions of the structure, and (b) six welded connections defined as the TRIs, which are marked as dotted red circles in (a).

ordered images after augmentation. Depending on the learning rates, number of images, network configuration, and hardware specification, this training process typically takes between a couple of hours to several weeks. The detailed implementation of our experimental study is discussed in Section "ROI localization and classification."

Experimental verification

Description of experiment

For demonstrating the method developed here, a full-scale highway sign truss structure (hereafter, the test structure) is used, as shown in Figure 3(a). This structure was originally built for supporting highway message signs and was taken out of service. Currently, it is located outside the Bowen Large-Scale Structural Engineering Laboratory at Purdue University.[42] The test structure is composed of six cubic segments. It has 4 main chords, 28 vertical braces, 24 diagonal braces, and 7 internal braces oriented diagonally to the main chord. All members are tubular sections. The diameters of the main chord, vertical brace, and diagonal (including internal) brace are 152.4, 63.5, and 76.2 mm, respectively. The braces are connected to the main chords using welds.

A total of six fillet welded connections between the vertical or diagonal braces and main chords, which are regarded as critical components, are selected as TRIs for the test structure. As shown in Figure 3(b), since tubular braces and tubular chords are connected, the shape of the weld is a warped circular line. The dotted red circles represent the TRIs. From right to left in Figure 3(a), these are denoted as Welds 1–6. Welds 2, 3, and 5 (denoted as Type 1) and Welds 1, 4, and 6 (denoted as Type 2) are created where the vertical and diagonal braces are attached to the main chord with diameters 63.5 and 101 mm, respectively.

Baseline model construction

To construct the baseline model for this verification, a total of 5321 baseline images were collected from the test structure over 5 months. Images were collected on 11 different days at different times of day and/or weather conditions. Sample baseline images are shown in Figure 4. Each column in this figure shows images captured on different days/times. Having such a large collection of images under various environmental conditions enables matching robust and unique features that are invariant to environment changes and have enhanced descriptors that normalize the variance resulting from varying lighting conditions. A total of two types of commercial cameras are used for image collection: Canon 6D with 24–105 mm (DSLR camera) and Canon PowerShot SX280 HS (compact camera).[43] The resolutions of each camera are 5472 × 3648 and 4000 × 2664 pixels, respectively. While the compact camera typically has an image sensor with a high cropping factor, producing low-quality images, it is widely used for commercial small- and mid-sized UAVs due to the light payload. For the collection of test images, we only use this compact camera to mimic image collection conditions similar to that of UAVs. No zoom or flash functions are used, and the focal length is fixed using a manual mode. We do not fly a UAV, although the process and hardware involved in the image acquisition is designed to be representative of the process needed for practical use of this technique using UAVs. Thus, it is important to note that these images are acquired from locations that are not predetermined. Furthermore, for successful construction of the baseline model using SfM, the image collection process is

Figure 4. Sample baseline images used for constructing a baseline model: Images in each column are collected on different days and under different lighting and weather conditions.

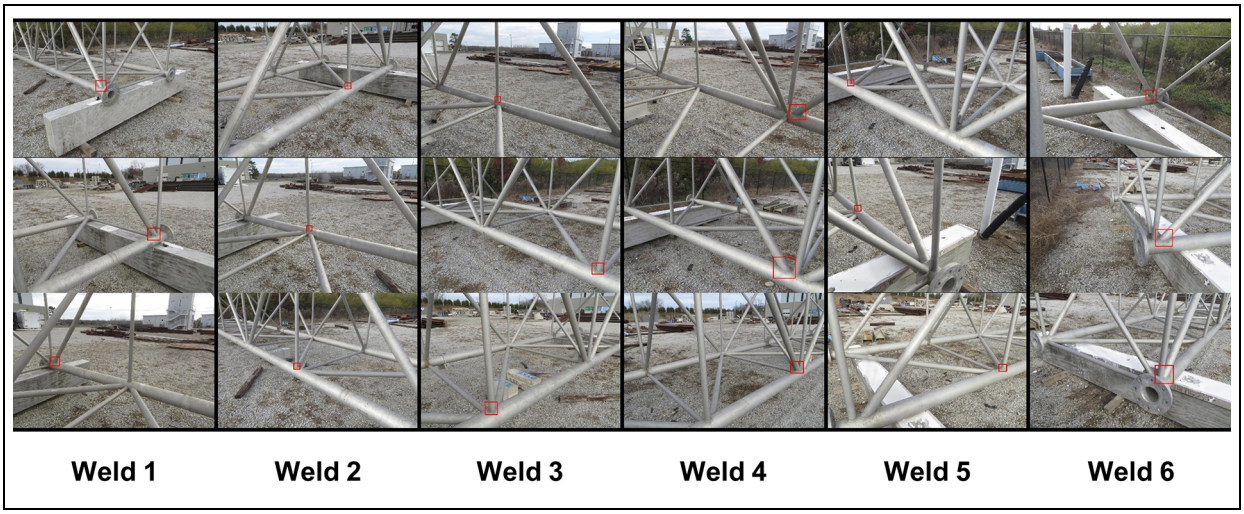

Figure 5. Bounding box (ROI) on each sample baseline image: Red rectangular boxes on the images in each column indicate Welds 1–6 (from left to right).

designed in such way that large overlaps exist between the images.

For the rapid SfM process, the size of each baseline image is reduced to one-quarter of the original one, which allows reducing the number of features to be matched. However, their ROIs are extracted from the original resolution images for visual assessment in the future. We use VisualSfM to compute the projection matrices and to construct the 3D point cloud. VisualSfM is a noncommercial, free software having a user-friendly graphical user interface (GUI).[28,29] This software has been widely used for many applications due to its high accuracy and speed. By incorporating a SiftGPU library, the computation speed is dramatically increased in feature detection and matching.[28] Other than the SfM process, the remainder of the processes, including image registration, ROI localization, and ROI classification, are implemented using MATLAB. Figure 1(a) shows a 3D point cloud model constructed for use as the baseline model, and the colored dots in various locations on the structure indicate the locations of the images. A total of 760,948 3D points are generated. The descriptor for each 3D point is computed by averaging the descriptors (scale-invariant feature transform (SIFT) descriptors) of participating 2D features on the images. Note that Figure 1(a) is only visualizing the 3D point cloud and camera poses, and no manual operations are used to directly manipulate these data in the 3D view.

Figure 6. Sample ROIs used for training a BOC: (a) positive (nonoccluded ROI) and (b) negative (occluded ROI).

To define the TRIs (a one-time process) in the baseline model, we manually attach red circular stickers at each weld, and these are clearly visible on the images. A total of four stickers are attached at the ends of both major and minor axes if the cross section of the welds is assumed to be an ellipse (precisely, they are warped ellipses). Among the sets of test images collected on 11 different days, only the final three image sets were captured from the test structure after attaching these stickers. In Figure 6, some sample ROIs show the stickers because they are localized from the images included in those sets. By matching the same sticker in a few different images (precisely, the center location of the sticker on the image), the 3D location of the corresponding sticker can be computed. Once the 3D locations of the two ends of the major axis for the TRIs are computed, the 3D center location and diameter (length of the major axis) of the weld can be computed. These two parameters can then be used for defining a virtual sphere (Yeum et al., 2017)[3]. Then, we apply an augmentation factor of 1.5 to the diameter of the virtual sphere to ensure we include the entire TRI when extracting each ROI.

Next, the ROIs are extracted from the baseline images to train the BOC for filtering out occluded ROIs. Note that the ROIs extracted from the baseline images are only used for training the classifier, and this step has nothing to do with the actual visual inspection. Also, to be useful for visual inspection, an ROI must be sufficiently visible in the image. In this study, the ROIs with a pixel size smaller than the diameter of the virtual sphere in the corresponding TRI (mm) are automatically removed. For example, if the minimum dimension (width or height) of an ROI corresponding to a Type 1 connection is smaller than 95.25 pixels (63.5 mm $\times$ 1.5 (augmentation factor)), then that ROI is rejected. Thus, both the ROIs localized from the baseline images and the test images for Type 1 connections are always larger than 95.25 pixels.

Knowing the projection matrix for each baseline image, and the 3D location of the TRIs in the baseline model, the ROIs of all six TRIs are localized from the baseline images. Sample images indicating the ROIs obtained are shown in Figure 5. The ROIs, marked with a red box on the baseline images in each column, indicate Welds 1–6 from left to right. Since the images are collected from various angles and positions, the ROIs extracted show various viewpoints of the TRIs, but their scale is almost identical. For example, when the image is captured close to the TRI, the size of the bounding box (ROI) is large.

A total of 4298 ROIs corresponding to all six welds are localized from the baseline images. We manually annotate these images to construct a dataset for training the BOC. Nonoccluded ROIs, denoted as positive, are defined as those in which the entire weld line on the ROI, that can be maximally viewed at the corresponding image location, is not interrupted by any object(s) in front. The remainder of the ROIs are annotated as negative for training. Some sample images are presented in Figure 6. Among 4298 ROIs, 945 ROIs are annotated as negative, and the rest are positive. All labeled ROIs are randomly divided into 2144 (50%), 1077 (25%), and 1077 (25%) images for training, validating, and testing. Note that the samples here are used to evaluate the performance of the trained BOC and are not actually used for extracting the ROIs on the test images.

In this study, we train a single binary classifier that is then applied to all welded connections. This approach is possible for inspection of this structure because the visual appearances of the welded connections are quite similar to each other (see Figure 6(a)) and considerably different than the occluded ones in Figure 6(b). However, if the appearance of the TRIs was visually dissimilar, and common visual features were not shared with each other, multiple classifiers would need to be trained individually for each type of TRI. In such a case, a large number of images are still needed to train the ROIs for each type of TRI to build an unbiased classifier.

We implement a popular ImageNet CNN model called AlexNet (TorontoNet in Caffe), framed in

MatConvNet library.[44] AlexNet exhibited superior implementation among CNNs in computer vision applications in the ImageNet image classification competition in 2012 and has been widely used for a benchmark test of a CNN model.[41] The network architecture has been presented in detail in the literature.[40] Although some improved architectures have been introduced since then, in this study, we use AlexNet based on its recognized performance as we implement it for this proof of concept for a simple, general CNN model.

As a preprocessing step, the raw images (ROIs) are first resized to 256 × 256 (pixel) to produce a set of regularized input images. The images are rescaled such that the shorter side of the image is set to 256 pixels and then the central 256 × 256 patch is cropped from the resulting image.[40] Once we resize the set of images, the mean red, green, blue values across the training set is subtracted from each pixel. This step will result in generating features that are unbiased with respect to color. For data augmentation, 227 × 227 patches are randomly cropped from each of the 256 × 256 images in each epoch. For further augmentation of the training set, the brightness of the patches is randomly varied to enable lighting invariance in the classification. The last 1000-way softmax layers in the original implementation from the ImageNet competition are modified to a logistic layer for binary classification. The layers are initialized as a Gaussian distribution with a zero mean and variance equal to 0.1. The hyper-parameters are the same as those used in AlexNet.[40] We trained our models using stochastic gradient descent with a batch size of 256 images, a momentum of 0.9, and a weight decay of 0.0005. The network is trained with/over 120 epochs, and the learning rate is logarithmically decreased from 10e−2 to 10e−6 during training. A workstation having a Xeon E5-2609 central processing unit (CPU) and a GPU, NVidia Tesla K40 with 12-GB video memory, is used for training and subsequent testing of the algorithm. The MatConvNet library installed in MATLAB 2016b is used for this study.[44] Training 2144 images, plus 1077 images for validation in each epoch, takes around 1 min.

In this study, ROI classification successfully attains a relatively high accuracy. We obtain rates of 89.73% (743/828 images) true positive (true classification of nonoccluded ROIs) and 91.83% (225/245 images) true negative, respectively. The precision is 97.37%, defined as the number of true positives over the total number of positives. Although these rates will vary slightly depending on the CNN architectures and their parameters, the overall performance of this approach is quite successful. These classification results imply that the trained BOC can successfully filter the occluded ROIs with high accuracy.

ROI localization and classification

For demonstrating the actual visual evaluation scenario, test images are collected from each of the TRIs using the compact camera mentioned in Section "Baseline model construction." The number of images collected from all six of the TRIs is listed in Table 1. First, SIFT features and descriptors are extracted from each test image at the original resolution. Each image of size 4000 × 2664 produces around 6000–12,000 SIFT features. Second, these features are matched with 3D points in the baseline model to register each of the test images. Using the strategy of reducing the 3D point cloud for matching, introduced in Section "ROI localization," the number of 3D points (here, 700,000) used for each TRI is reduced to around 1/7 on average. Instead of exhaustive searching for the closeness of the descriptors, we used a GPU-implemented k-nearest search algorithm in MATLAB to rapidly find the best pairings.[45] The criterion to accept matches is the same as the one implemented in vl_ubcmatch in the VLFeat library: "A descriptor D1 is matched to a descriptor D2 only if the distance between D1 and D2 multiplied by 1.5 is not greater than the distance of D1 to all other descriptors."[46]

Based on the pairing between the 3D points in the baseline model and the 2D features in each of the test images, the projection matrices and one lens distortion parameter are calibrated. For this purpose, the function "solvepnpransac" in OpenCV is implemented through the "mexopencv" library, which is a development kit of MATLAB MEX functions for the OpenCV library.[38,47] This function is used to estimate the extrinsic camera matrix for each image based on the initial assumption for the intrinsic matrix using the RANSAC scheme. Then, a bundle adjustment is used to refine all parameters for the projection matrix (intrinsic and extrinsic matrices), as well as the lens distortion parameter in each image, by minimizing reprojection errors of inlier sets obtained from previous RANSAC process using LM algorithm.

Based on the projection matrix associated with each test image, the ROI localization and classification outcomes are shown in Figure 7. In each double row, 30 random localized ROIs are shown for Welds 1–6 from top to bottom. The actual number of localized ROIs is listed in Table 1. Because any ROIs having insufficient resolution and not including the ROIs are rejected, the number of ROIs identified in these results is fewer than the total number of test images. Next, the trained BOC is applied to the localized ROIs to filter out the occluded ROIs that are not useful for visual evaluation. Among the 30 ROIs shown in Figure 7, the ROIs that are classified as negative (with a maximum of 10 shown) are added at the end of the list and marked

Table 1. Results of the ROI localization and classification.

	Weld 1	Weld 2	Weld 3	Weld 4	Weld 5	Weld 6
No. of images	119	77	88	84	60	55
No. of localized ROIs	104	51	54	70	45	47
No. of classified ROIs (positive/negative)	69/35	49/2	48/6	47/23	44/1	33/14
Precision (%)	92.75	100	97.92	85.11	100	90.91

ROI: region of interest.

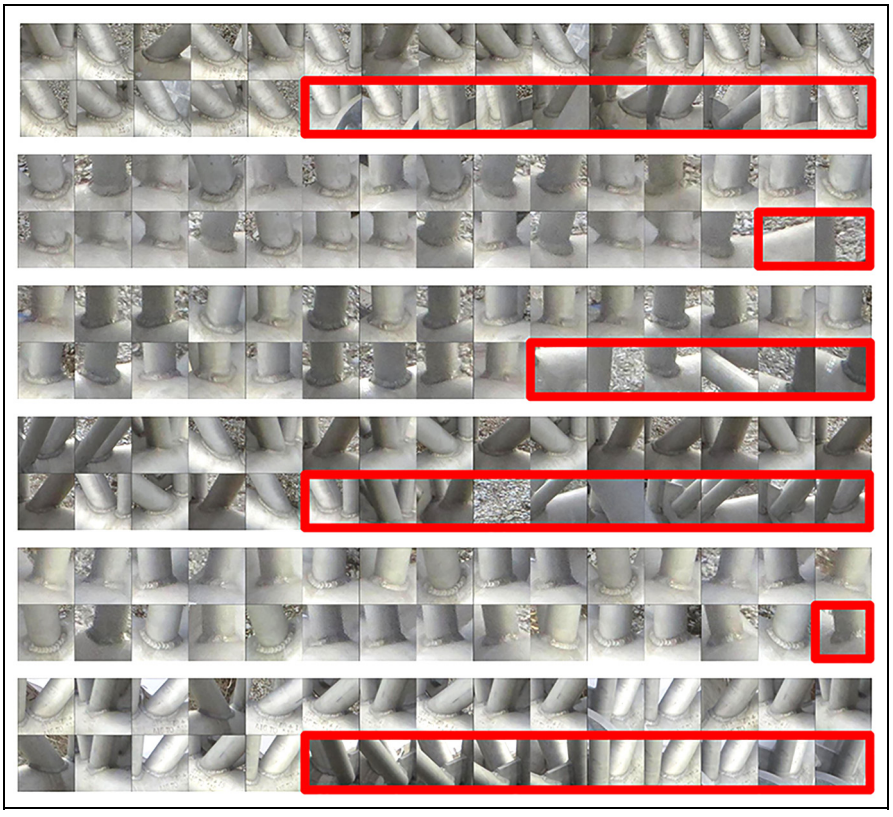

Figure 7. Examples of ROIs that have been localized and classified from the set of test images: Each set of 30 localized ROIs corresponds to Welds 1–6 (from top to bottom). A maximum of 10 negatively classified ROIs are positioned at the end of the set (marked with a red box). The rest are classified as positive.

with a red box. The rest of them are classified as positive. The total number of ROIs that is classified as positive and negative is listed in Table 1. Also, after manually annotating the classified ROIs extracted from the test images, the precision in classification is computed. Overall, the performance of this classifier is exceptionally successful based on its high precision. Such high precision, which is the number of true positives over the number of positives, implies that when reliable damage detection methods are applied to these ROIs, they are unlikely to produce false-positive errors because the majority of the positive ROIs are true-positive (nonoccluded) ones.

Note that the BOC is originally designed and trained to filter out occluded ROIs based on the assumption that the baseline images are precisely calibrated in the baseline model. However, in reality, test images are occasionally incorrectly registered due to spurious 3D–2D matches, which also produced incorrect ROIs. This outcome is demonstrated, for instance, with the two negative ROIs shown for Weld 2 in Figure 7. Although this type of incorrect ROIs is not explicitly considered in the training process, the BOC trained here is able to successfully classify these images as negatives (and filter them out) because the common visual features for positive ROIs are quite different from the ones in these

ROIs. In this particular scenario, erroneous registration results in ROIs that are similar to typical occluded ROIs.

Conclusion

This study presents and verifies an automated technique for ROI localization and classification that will directly enable robust, vision-based, visual structural assessment. This technique overcomes a major practical barrier in the use of large volumes of complex images, such as those collected with UAVs for assessing structural condition. A key technical achievement in this study is to make the best use of the collected images to (1) efficiently localize ROIs by computing the 3D geometric relationship between the TRIs and the images using SfM and (2) obtain the most useful ROIs by learning their 2D unique visual patterns using CNN algorithms to implement a BOC.

Our technique automatically extracts highly regularized, relevant, and useful ROIs from the images for visual evaluation. The technique is more efficient, simpler, and robust than previous approaches. The capabilities of this technique are demonstrated using a full-scale highway sign truss with complex geometry that is placed outdoors. A total of six welded connections are assigned as TRIs, and 340 ROIs containing scaled images of the TRIs are localized from 520 test images. Favorable and nonoccluded ROIs are successfully classified (to facilitate eliminating these from use in the inspection process) with 94.13% precision on average. Ultimately, we expect that this technique will play an enabling role in the development of methods to automatically assess various large-scale civil structures using the images collected from UAVs.

Acknowledgements

We thank technicians at the Herrick Laboratory and Bowen Laboratory at Purdue University for preparing the experiment and Prof. Robert Connor at Purdue University for acquiring the truss used in this study.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was partially supported by the National Science Foundation under NSF grant no. CMMI-1645047.

References

1. Indiana Department of Transportation. Bridge inspection manual, 2013, http://www.in.gov/dot/div/contracts/standards/bridge/inspector_manual/INDIANA%20BRIDGE%20INSPECTION%20MANUAL.pdf
2. Washer G, Nasrollahi M, Applebury C, et al. Proposed guideline for reliability-based bridge inspection practices. NCHRP report 782, 2014. Washington, DC: Transportation Research Board.
3. Yeum CM, Choi J and Dyke SJ. Autonomous image localization for visual inspection of civil infrastructure. *Smart Mater Struct* 2017; 26(3): 035051.
4. Portland Cement Association. Concrete slab surface defects: causes, prevention, repair, 2001, https://www.oboa.on.ca/events/2009/sessions/files/Slab%20Surface%20Prevention%20Repair.pdf
5. Abdel-Qader I, Abudayyeh O and Kelly ME. Analysis of edge-detection techniques for crack identification in bridges. *J Comput Civil Eng* 2003; 17(4): 255–263.
6. Yamaguchi T and Hashimoto S. Fast crack detection method for large-size concrete surface images using percolation-based image processing. *Mach Vision Appl* 2010; 21(5): 797–809.
7. Zhang W, Zhang Z, Qi D, et al. Automatic crack detection and classification method for subway tunnel safety monitoring. *Sensors* 2014; 14(10): 19307–19328.
8. Zou Q, Cao Y, Li Q, et al. CrackTree: automatic crack detection from pavement images. *Pattern Recogn Lett* 2012; 33(3): 227–238.
9. Subirats P, Dumoulin J, Legeay V, et al. Automation of pavement surface crack detection with a matched filtering to define the mother wavelet function used. In: *Proceedings of the* IEEE *conference on signal processing* 14th European, Florence, 4–8 September 2006, pp. 1–5. New York: IEEE.
10. Yang Y and Nagarajaiah S. Dynamic imaging: real-time detection of local structural damage with blind separation of low-rank background and sparse innovation. *J Struct Eng* 2015; 142(2): 04015144.
11. German S, Brilakis I and DesRoches R. Rapid entropy-based detection and properties measurement of concrete spalling with machine vision for post-earthquake safety assessments. *Adv Eng Inform* 2012; 26(4): 846–858.
12. Yeum CM, Dyke SJ, Ramirez J, et al. Big visual data analytics for damage classification in civil engineering. In: *Proceedings of the international conference on smart infrastructure and construction* (ICSIC), Cambridge, 27–29 June 2016.
13. Koch C and Brilakis I. Pothole detection in asphalt pavement images. *Adv Eng Inform* 2011; 25(3): 507–515.
14. Torok MM, Golparvar-Fard M and Kochersberger KB. Image-based automated 3D crack detection for post-disaster building assessment. *J Comput Civil Eng* 2013; 28(5): 334.
15. Lee S, Chang LM and Skibniewski M. Automated recognition of surface defects using digital color image processing. *Automat Constr* 2006; 15(4): 540–549.

16. Chen PH, Shen HK, Lei CY, et al. Support-vector-machine-based method for automated steel bridge rust assessment. *Automat Constr* 2012; 23: 9–19.
17. Bonnin-Pascual F, Ortiz A and Aliofkhazraei DM. Corrosion detection for automated visual inspection. *Dev Corros Prot* 2014; 25: 619–632.
18. Neogi N, Mohanta DK and Dutta PK. Review of vision-based steel surface inspection systems. *EURASIP J Image Video Process* 2014; 2014(1): 50.
19. Yeum CM and Dyke SJ. Vision-based automated crack detection for bridge inspection. *Comput-Aided Civ Inf* 2015; 30(10): 759–770.
20. Hartley R and Zisserman A. *Multiple view geometry in computer vision*. 2nd ed. Cambridge: Cambridge University Press, 2003.
21. Snavely N, Seitz SM and Szeliski R. Modeling the world from internet photo collections. *Int J Comput Vision* 2008; 80(2): 189–210.
22. Turner D, Lucieer A and Watson C. An automated technique for generating georectified mosaics from ultra-high resolution unmanned aerial vehicle (UAV) imagery, based on structure from motion (SfM) point clouds. *Remote Sens* 2012; 4(5): 1392–1410.
23. Mancini F, Dubbini M, Gattelli M, et al. Using unmanned aerial vehicles (UAV) for high-resolution reconstruction of topography: the structure from motion approach on coastal environments. *Remote Sens* 2013; 5(12): 6880–6898.
24. Golparvar-Fard M, Peña-Mora F and Savarese S. Automated progress monitoring using unordered daily construction photographs and IFC-based building information models. *J Comput Civil Eng* 2012; 29(1): 04014025.
25. Pix4D. Drone mapping software, https://pix4d.com (accessed 30 April 2017).
26. Agisoft. PhotoScan, http://www.agisoft.com/ (accessed 30 April 2017).
27. Bentley. Infrastructure and engineering software and solutions, https://bentley.com (accessed 1 May 2017).
28. Wu C. Towards linear-time incremental structure from motion. In: Proceedings of the IEEE *conf*erence on 3DTV, Seattle, WA, 29 June–1 July 2013, pp. 127–34. New York: IEEE.
29. VisualSFM. A visual structure from motion system, http://ccwu.me/vsfm (accessed 30 April 2017).
30. OpenMVG. OpenMVG, https://github.com/openMVG/openMVG (accessed 31 May 2017).
31. Moulon P, Monasse P and Marlet R. Global fusion of relative motions for robust, accurate and scalable structure from motion. In: *Proceedings of the IEEE international conference on computer vision*, Sydney, NSW, Australia, 1–8 December 2013, pp. 3248–3255. NewYork: IEEE.
32. Sweeney C. *Theia multiview geometry library: tutorial & reference*. Santa Barbara, CA: University of California, Santa Barbara, 2015.
33. Pix4D. Designing the image acquisition plan, https://support.pix4d.com/hc/en-us/articles/202557409 (accessed 30 September 2017).
34. World of Drone. Drone and aerial observation, http://drones.newamerica.org/primer/ (accessed 30 April 2017).
35. Lim H, Sinha SN, Cohen MF, et al. Real-time image-based 6-DOF localization in large-scale environments. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, Providence, RI, 16–21 June 2012, pp. 1043–1050. New York: IEEE.
36. Santana LV, Brandao AS, Sarcinelli-Filho M, et al. A trajectory tracking and 3D positioning controller for the AR.Drone quadrotor. In: *Proceedings of the IEEE conference on unmanned aircraft systems (ICUAS)*, Orlando, FL, 27–30 May 2014, pp. 756–767. New York: IEEE.
37. Sattler T, Leibe B and Kobbelt L. Fast image-based localization using direct 2D-to-3D matching. In: *Proceedings of the* IEEE *conference on computer vision* (ICCV), Barcelona, 6–13 November 2011, pp. 667–674. New York: IEEE.
38. Open Source Computer Vision Library (OpenCV), http://opencv.org/ (accessed 30 April 2017).
39. LeCun Y, Boser B, Denker JS, et al. Backpropagation applied to handwritten zip code recognition. *Neural Comput* 1989; 1(4): 541–551.
40. Krizhevsky A, Sutskever I and Hinton GE. ImageNet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*, 2012, pp. 1097–1105.
41. Russakovsky O, Deng J, Su H, et al. ImageNet large scale visual recognition challenge. *Int J Comput Vision* 2015; 115(3): 211–252.
42. Bowen Laboratory, Lyles School of Civil Engineering, Purdue University,https://engineering.purdue.edu/CE/Bowen (accessed 30 April 2017).
43. Canon. Canon EOS digital SLR cameras & PowerShot digital cameras. Canon online store, https://shop.usa.canon.com/shop/en/catalog/cameras (accessed 30 April 2017).
44. MatConvNet. MatConvNet: CNNs for MATLAB, http://www.vlfeat.org/matconvnet/ (accessed 5 March 2017).
45. MATLAB. *Version 7.10.0 (R2016b)*. Natick, MA: MathWorks, 2016.
46. VLFeat. Computer vision algorithms library, http://www.vlfeat.org (accessed 30 April 2017).
47. GitHub. Collection and a development kit of MATLAB MEX functions for OpenCV library, https://github.com/kyamagu/mexopencv (accessed 30 April 2017).