COMPUTER VISION-BASED STRUCTURAL ASSESSMENT EXPLOITING LARGE VOLUMES OF IMAGES

by

Chul Min Yeum

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

Department of Civil Engineering

West Lafayette, Indiana

December 2016

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Shirley J. Dyke, Chair

Lyles School of Civil Engineering

Dr. Bedrich Benes

Department of Computer Graphics Technology

Dr. Juan P. Wachs

School of Industrial Engineering

Dr. Julio A. Ramirez

Lyles School of Civil Engineering

Dr. Santiago Pujol

Lyles School of Civil Engineering

Dr. Zygmunt Pizlo

Department of Psychological Sciences

Approved by:

Dr. Buck Doe

Head of the Departmental Graduate Program

This dissertation is dedicated to my wife, Seulki, who has been a constant source of support and encouragement during the challenges of graduate degree and life. I am truly thankful for having you in my life.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Dr. Dyke for the continuous support of my Ph.D. study and related research, for his patience, encouragement, and immense knowledge. Her guidance helped me in all the time of research and made this a thoughtful and rewarding journey. I could not have imagined having a better advisor and mentor for my Ph.D. studies.

Besides my advisor, I would like to thank the rest of my dissertation committee: Dr. Ramirez, Dr. Pujol, Dr. Benes, Dr. Pizlo, and Dr. Wachs, for their insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives.

Next, I wish to acknowledge the valuable data contributions from CrEEDD (Center for Earthquake Engineering and Disaster Data) at Purdue University (datacenterhub.org), the EUCentre (Pavia, Italy), the Instituto de Ingenieria of UNAM (Mexico), FEMA, and the EERI collections.

Finally, I acknowledge support for my studies from National Science Foundation under Grant No. NSF-CNS-1035748 and NSF-1608762, Small Business Innovative Research (SBIR) Program and the Engineering Research and Development Center – Construction Engineering Research Laboratory (ERDC-CERL) under Contract No. W9132T-12-C-0020, as well as the Indiana Joint Transportation Research Program SPR-4006.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Author: Yeum, Chul Min. Ph.D.
Institution: Purdue University
Degree Received: December 2016
Title: Computer Vision-Based Structural Assessment Exploiting Large Volumes of Images.
Major Professor: Shirley J. Dyke.

Visual assessment is a process to understand the state of a structure based on evaluations originating from visual information. Recent advances in computer vision to explore new sensors, sensing platforms and high-performance computing have shed light on the potential for vision-based visual assessment in civil engineering structures. The use of low-cost, high-resolution visual sensors in conjunction with mobile and aerial platforms can overcome spatial and temporal limitations typically associated with other forms of sensing in civil structures. Also, GPU-accelerated parallel computing offers unprecedented speed and performance, accelerating processing the collected visual data. However, despite the enormous endeavor in past research to implement such technologies, there are still many practical challenges to overcome to successfully apply these techniques in real world situations. A major challenge lies in dealing with a large volume of unordered and complex visual data, collected under uncontrolled circumstance (e.g. lighting, cluttered region, and variations in environmental conditions), while just a tiny fraction of them are useful for conducting actual assessment. Such difficulty induces an undesirable high rate of false-positive and false-negative errors, reducing the trustworthiness and efficiency of their implementation. To overcome the inherent challenges in using such images for visual assessment, high-level computer vision algorithms must be integrated with relevant prior knowledge and guidance, thus aiming to have similar performance with those of humans conducting visual assessment. Moreover, the techniques must be developed and validated in the realistic context of a large volume of real-world images, which is likely contain numerous practical challenges. In this dissertation, the use of computer vision algorithms is explored to address two promising applications of vision-based visual assessment in civil engineering: visual inspection, and visual data analysis for post-disaster evaluation. For both applications, techniques are developed here to enable reliable and efficient visual

assessment for civil structures and demonstrate them using a large volume of real-world images collected from actual structures. State-of-art computer vision techniques, such as structure-from-motion and convolutional neural network techniques, facilitate these tasks. This highly cross-disciplinary research is scalable and expandable to many other applications in vision-based visual assessment, and will serve to close the existing gaps between past research efforts and real-world implementations.

CHAPTER 1. INTRODUCTION

1.1 Motivation

Visual changes provide obvious warning signs that a structure's condition is deteriorating, and firm evidence of the consequences certain unexpected event(s) affecting the structure in the past. Thus, visual assessment is the primary form of structural evaluation to support decisions relating to their safety, maintenance, and repair. Typical applications of visual assessment in the field of civil engineering include inspection of damage in a structure, oversight of the construction process and equipment, post-hazard assessment for situational awareness or post-event reconnaissance, all of which are based strongly on visual information as the main source of information. Here, a "structure" refers to the large-scale facilities that constitute the built environment providing commodities and services essential to enable, sustain, or enhance societal living conditions such as roads, bridges, tunnels, or other constructed facilities.

Currently, human engineers should actively involve at least one or more steps of visual assessment, including the observation, data collection, analysis, decision making, and documentation. The human visual assessment does have certain limitations. First, human inspection is expensive and time-consuming. Civil structures are relatively large and are often in a harsh environment, introducing challenges in reaching and accessing the critical regions needed to view the structure. Expensive equipment or scaffolding is often required to reach important locations in certain structures, and thus sometimes these areas may be superficially examined or even neglected. Thus, inspections must be performed by personnel trained for such situations. Second, human inspection can be inconsistent, because human's attention and abilities vary from day to day, and the visual data is still manually analyzed and documented by humans. Relying on an engineer's subjective, qualitative, or empirical knowledge may result in false evaluations, followed by erroneous reports and documents. Third, human inspection is, in some cases, time-critical. In other words, there may be an immediate need for decision-making based on visual evaluation in some applications. For instance, it may be necessary to make decisions related to disaster response, to repair needs or closure of a structure based on damage existing in a critical

region, or to identify the needs for data collection during a reconnaissance trip. For time-critical needs, techniques that can sift through the information in a rapid and efficient manner should be developed.

There is a compelling need to offer rapid, efficient and inexpensive tools to support various applications for visual assessment. Ultimately, a breakthrough will be established by replicating the actions and abilities of the human performing visual assessment. To achieve this goal, the necessary steps will involve sensing, processing, and mobile platform, and through automation of these functions, the benefits will yield significant savings in time and cost. Fig. 1.1 indicates how the research community can exploit recent technologies to support, or eventually replace, these functions of the human using computer automation. For the sensing and platform, current visual sensing capabilities are quite remarkable, achieving fine temporal and spatial granularity. As imaging sensors have become smaller, cheaper and more powerful, images including RGB, infrared or thermal can be readily collected from smartphones or mobile cameras with almost no additional cost. Moreover, recently commercialized unmanned aerial vehicles (UAVs) have expanded sensor mobility from ground to sky (Bonnin-Pascual and Ortiz, 2014; New America, 2015), and a head-mounted optical displays such as Google Glass™ or action cameras have received significant attention for their ability to record and learn human actions. With the aid of modern parallel computations and GPUs, powerful computer vision methods and machine learning algorithms have been realized and established within computer science and engineering, and related disciplines. In some applications, these have nearly achieved human-level performance (Taigman et al., 2014). These methods have been considered for a broad range of applications, ranging from speech or text recognition to autonomous driving (Ciresan et al., 2012; Chen et al., 2014; Hannun et al., 2014). Such methods have the potential to transform the way computers recognize visual contents-of-interest in visual data, beyond just using the metadata recorded in a text form. These new opportunities are beginning to compete with the human-based visual assessment regarding cost and performance.

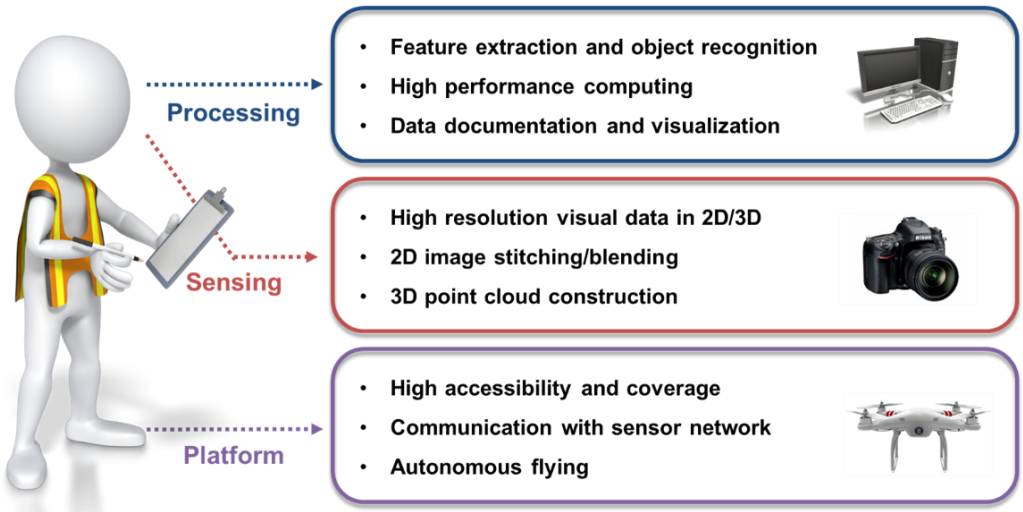


Figure 1.1 Recent technology that can replicate the functions of human for visual assessment.

Despite the tremendous advances made in the relevant technologies, at this time, computer vision methods are rarely implemented for real-world visual assessment. This unfortunate situation is due to the fact that the techniques, when used in isolation, still have limitations for use in vison-based assessment. The methods are quite capable of replicating individual functions of the human. However, a major barrier to the use of these technologies for visual assessment is that a good understanding of how to implement them in a task-oriented and realistic manner is lacked, especially when it comes to a large volume of images. The technologies are certainly not “plug and play.” For example, many people are expecting that advancing UAV technologies will automatically yield a breakthrough in visual inspection for inaccessible and large civil structures. However, humans still need to spend significant time to collect, organize, and view all of the images or videos from the UAV. Thus, few significant gains in efficiency or cost have been achieved to date. A UAV must be able to be more selective in the acquisition and collection of photos to generate greater efficiency. Moreover, processing of the raw images without any preliminary filtering is computationally expensive and is also hindered by a large number of false-positive errors, which would impede actual implementation in the field. Further, as the cost of image collection decreases, people tend to collect more and more photos to inform decision making and scientific research. However, without computational methods that can enable useful automation, manual documentation and analysis are quite expensive. Thus, valuable visual data, collected at some cost, remain unused in many cases.

The methods in this dissertation are focused on breaking down these barriers to exploit the relevant technologies to their full potential, and to leverage existing computer vision techniques or those in development to be used more accurately and efficiently with large volumes of images. As an example of damage detection, existing techniques have continued to focus on acquiring an answer to the question "Can engineers detect damage on images?". Unfortunately, regardless of accuracy and reliability of the underlying techniques, the increasing complexity, and uncertainty in the collected large number of images remains a challenge for those methods. Alternatively, the techniques developed in this dissertation provide a means to detect the images that deserve to be processed based on the content and likelihood of meaningful information available on images, as extracted using modern computer vision technologies.

1.2 Objective and Contribution

In this study, the overarching research objective is to develop and demonstrate practical and feasible vision-based visual assessment techniques using a large volume of images. These methods will enable automated, reliable, and efficient evaluation of civil engineering structures. Beyond simply processing the images, the information available in the image(s) is extracted by leveraging modern computer vision techniques, including feature extraction, object recognition, and multi-view geometry. By incorporating these advanced technologies, the transformative techniques developed in this dissertation are capable of detection, localization, classification and evaluation of visual data, centered in the structural engineering domain for automated vision-based visual assessment. Two promising applications of vision-based visual assessment are considered in this dissertation: visual inspection for condition evaluation, and visual data analysis for post-disaster evaluation.

The key contribution of this research is to push the boundaries of computer vision into a new territory in such a way as to realize existing and future vision-based visual assessment techniques when it comes to a large volume of complex images collected from real-world structures. Previous researchers have developed and validated their techniques using a small quantity of images that were collected with the intention of using them for specific purposes or applications. However, in real circumstances, there is no guarantee

that one may be able to collect favorable images; in the real world, there is significant uncertainty in locations, viewpoints, or contents. Furthermore, there is no assurance that the methods will be able to handle large numbers of, complex, and unstructured images in such a way as to be tractable in the context of processing, analysis, and documentation. Instead, in a manner that is entirely different from the prior research, the techniques developed here utilize state-of-art computer vision techniques to fully extract and exploit all information available on images, including visual features, camera geometry, and contents so as to reduce the complexity coming from processing and analyzing a large volume of images. Thus, the techniques developed here will really enable a vision-based visual assessment to address real-world problems in a feasible way.

1.3 Scope of Work

In this dissertation, several important techniques are developed that, when incorporated into the procedures of existing or future vision-based visual assessment techniques, will enable for their successful implementations in real-world problems. The first half of this dissertation described the core techniques developed for enabling futuristic autonomous visual inspection in civil structures. In Chapter 2, a new approach for automating damage detection is proposed. Rather than searching for cracks on entire images, objects which have areas susceptible to cracks (bolts in this case) are first detected on all of the images. This strategy greatly increases detectability of cracks by narrowing down searching areas and damage scales in acquired images. In Chapter 3, an automated image localization technique is developed to extract regions of interest in each of the images before utilizing vision-based visual inspection techniques. Thus, this work goes beyond the specific approach developed in Chapter 2, in that this technique can extract regions-of-interest on images using the geometric relationship between the images and the target structure, which are independent of visual appearance. Analysis of such highly relevant and localized images would enable efficient and reliable visual inspection in the future. Both techniques are designed to be applicable for visual inspection of a large-scale civil engineering structure using a large number of images gathered from one or more UAVs. These core techniques can be adapted to consider other target objects and areas, and also integrated with existing damage detection techniques for various inspection tasks.

In the second half of this dissertation, a new field of research is addressed: the automated analysis, classification, and documentation of visual data collected from post-event reconnaissance. To systematically build the tools needed to automatically extract knowledge from images acquired from past disasters, the focus here is on developing a system to annotate (Chapter 4) and automatically classification and detection visual contents on images (Chapter 5). Here, “annotation” indicates that the visual contents on the images, which are to be documented in immediate or future usage, are identified in a manual or automated way. In Chapter 4, an advanced structured annotation method is proposed for describing semantic and descriptive contents on images originating from earthquake reconnaissance missions. The image annotation method enables conversion of the data into a searchable form, guided by various queries. Chapter 5 is dedicated to the development and demonstration of reconnaissance-driven computer vision methods capable of detection, classification, and evaluation of visual data for the automated analysis of large volumes of disaster reconnaissance images. Image classification and object detection are incorporated into the procedures to accurate extract the target contents of interest. As an illustration of the technique and its capabilities for visual assessment, collapse classification and spalling detection are demonstrated using a large volume of images gathered from past earthquake disasters.

The technologies developed in this dissertation are briefly summarized in Table 1.1. This table is a guide to navigating all research conducted.

Table 1.1 A brief summary of the techniques developed within this dissertation

	Vision-based Automated Crack Detection	**Autonomous Image Localization**	**Semantic Annotation of Earthquake Images**	**Collapse Image Classification**	**Spalling Detection and Localization on Images**
Motivating Questions	• How do engineers visually differentiate crack and scratch? • Is one image enough to detect damage? • Do engineers need to process the entire area of raw images for visual inspection?	• How do engineers handle a large volume of images collected from UAV for visual inspection? • How do engineers make existing vision-based visual inspection techniques feasible and practical?	• How to annotate visual information on images in a descriptive way? • How to retrieve annotated descriptive information?	• How do images support decision-making during a reconnoassiace mission? • How to automatically classify images-of-interest? • How to rapidly organize a large volume of images?	• How do images support decision-making during a reconnoassiace mission? • How to annotate visual contents on images collected from a disaster?
Purpose	Development of a futuristic vision-based crack detection using images gathered from UAVs	Localization of region-of-interest in images of a target structure	Retrievable annotation of descriptive information in earthquake images	Classification of collapse images among images gathered from reconnaissance mission	Detection and localization of spalling on images
Contribution	A new transformative approach for visual inspection using computer vision methods	Enabling the use of existing damage detection techniques in the context of a large volume of images collected from UAVs	Proposition of an advanced formalized and structured annotation method for earthquake images	Demonstration of the proposed method using an unprecedented number of real-world images	Demonstration of the proposed method using an unprecedented number of real-world images
Input Data	• Large volume of images collected from UAVs • Classifier for detecting target objects-of-interest	• Large volume of images gathered from UAVs • Locations of target areas in 3D • Baseline model in 3D	Images collected from earthquake events	• Large volume of images collected from building reconnaissance missions • Classifiers for image-of-interest	• Large volume of images collected from building reconnaissance missions • Classifiers for object-of-interest
Outcome	Detection of crack damage near selected target objects	Region-of-interest on images having the same scale of a target object from different view points	Retrievable descriptive information for earthquake images	Images-of-interest among all images collected at reconnaissance	Detection and localization of object-of-interest on images
Algorithm/ Technique	• Integral channel features • Multi-view geometry • Community detection • Edge detection	• Structure-from-motions • Multi-view geometry	• Web ontology language • Resource description framework (RDF)	Deep convolutional neural networks (CNN)	• Deep convolutional neural networks (CNN) • Selective searching algorithm
Feature	• Fully automated visual inspection • Design prior knowledge for damage detection	• Fully automated extraction of region-of-interest on images • Working on any sizes and shapes of region-of-inspection	• Earthquake image ontology • Triple-store annotation methodology	• Applicable to the classification of any images that have visual similarity. • Rapid classification of images	Automated estimation of size and location of objects-of-interest
Application	• Damage detection in large scale structures • Detection of target elements for visual inspection	• Visual inspection of large-scale structures in general • Visual condition monitoring	Archiving building reconnaissance images for scientific research	• Rapid disaster response • Automated documentation of building reconnaissance images for decision support	Automated documentation of building reconnaissance images for decision support
Chapter	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 5
Reference	Yeum and Dyke, 2015		Yeum, et al., 2017	Yeum, et al., 2016	

CHAPTER 2. VISION-BASED AUTOMATED VISUAL INSPECTION FOR LARGE-SCALE CIVIL INFRASTRUCTURE

2.1 Motivation

Visual inspection of the civil structure is customarily used to identify and evaluate faults. Most decisions relating to structural maintenance are based on assessments from visual inspections. However, a current visual inspection conducted by human inspectors has several limitations. The study, undertaken 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 (Moore et al., 2001). The study showed that there is a great discrepancy in the results when several inspectors inspect the same structure. Factors include accessibility, light intensity, lack of specialized knowledge, perception of maintenance, and visual acuity and color vision. Although some factors are attributed to carelessness or improper training of inspectors, most of them cannot be physically overcome by a human in the current visual inspection process.

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

operating personnel on the ground. Recently, a multi-rotor helicopter became commercially available to wirelessly take pictures or videos for inspection of structures such as pipelines, power lines or dams (Adams and Friedland, 2011).

Once a suitable collection of visual information is collected from a bridge, robust inspection techniques should automatically perform the visual inspection tasks outlined in the manual. Vision-based visual inspection is not a new concept and has been broadly developed and used for civil, mechanical or aerospace structures. In the last few decades, researchers have realized amazing improvements in vision-based structural evaluation for civil engineering applications. For civil engineering, the major tasks encompassed by visual inspection can be grouped according to the two most common materials used, concrete and steel, which exhibit entirely different characteristics when it comes to damage.

Defects in concrete, similar to asphalt pavement, typically manifest as cracks or delaminations. First, cracking is a major mode of damage in concrete, and inevitably occurs at initiation or during operation. However, a crack can be the result of one or a combination of factors such as drying shrinkage, thermal contraction, restraint shortening, subgrade settlement, and applied load (Portland Cement, 2001). Thus, the occurrence of a crack in concrete is not necessarily a cause for concern but should be left to the judgment of the inspector. The appearance of a crack has a mostly clear low intensity than the background, and its pattern is a straight or curved line with a relatively uniform width. Thus, intensity-based edge detection and segmentation approaches are widely used (Abdel-Qader et al., 2003; Jahanshahi et al., 2009; Yamaguchi et al., 2010). However, the challenges include (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 (Miller, 2004; Jahanshahi et al., 2013; Zhang et al., 2014). Second, delamination, such as flaking or spalling, is another likely damage scenario that could be investigated with visual methods. Abrupt delamination damage like spalling or potholes, can pose damage to users as well as accelerate another mode of damage, such as corrosion on steel rebar. Texture analysis and shape extraction techniques are used to extract damage areas in 2D (German, 2012), and

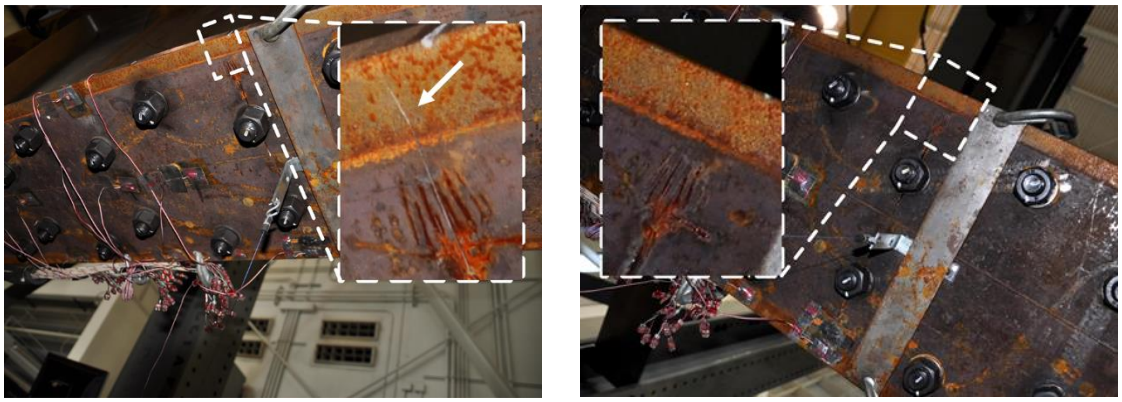
multi-view geometry is applied to obtain geometry information in 3D (Koch and Brilakis, 2011; Torok et al., 2014).

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. Corrosion appears as rust on uncoated, visible surfaces, and color-based corrosion detection and texture based corrosion have been widely studied (Lee et al., 2006; Chen et al., 2012; Bonnin-Pascual and Ortiz, 2014). Second, steel cracks, mainly fatigue cracks, occur at areas of stress concentration and frequently originate at a flaw associated with a weld or material inconsistency. Detection of cracks in steel can be harder than in concrete because the cracks have thin, shiny edges and may be invisible depending on lighting conditions and viewpoints. Similar to cracks in concrete, edge detection and segmentation techniques are used for detection of visibly obvious cracks, but they would require higher resolution images or large crack sizes for ready detection (Neogi et al., 2014).

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

This study was begun by exploring the question: Given a large volume of images from the UAV, would it be feasible to detect damage in a realistic structure using currently available vision-based damage detection techniques? To answer this question, many photographs were taken of a steel beam having a real fatigue crack that initiated from one of bolt holes. Analysis of the data was performed using standard available methods in

literature, such as edge detectors or morphological detectors (Jahanshahi et al., 2009). As a result, two major issues are identified that need to be addressed for vision-based automated inspection, which previous researchers may or may not have observed. First, searching for cracks over the entire area of an image generates many false-positive alarms and misdetections. In Fig. 2.1(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, in these cases, detection of the real crack by human inspectors is not easy but is still possible. This detection is because they have prior knowledge about the crack's typical appearance and characteristics. In this case, the relevant information is that cracks on a steel structure have thin and shiny edges and are often initiated and propagated from bolt holes (Indiana Department of Transportation, 2013). These features draw their attention to bolts and nearby areas, facilitating crack detection. A second issue observed is that the crack may be visible or invisible depending on the viewpoint. Fig. 2.1(b) shows same scenes of a fatigue crack but from two different viewpoints. Comparing the white dotted boxes in both images, Figs. 2.1(a) and (b), the crack is hardly observed in the second image. Therefore, it is concluded that images of the same scene from many different viewpoints are needed to detect the crack without knowing how and where it is created and propagated. Most previous research, of course, has unconsciously considered these two issues as they collect images through controlling the circumstances, such camera positions or angles depending on appearance and location of cracks. However, in reality, it is hard to obtain sufficiently good images taken under the "best" conditions because the crack location, crack direction, and lighting direction cannot be known in advance, and also it is hard to precisely and continuously control camera positions and angles installed in the UAV.



(a) (b)

Figure 2.1 Images of a fatigue crack on a steel beam captured from different viewpoints. Note that the fatigue crack in the images is initiated from the tip of the artificial notch near the bolt, which is to simulate initial discontinuity of the circumference of the bolt hole.

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

In this chapter, as one of visual inspection tasks, cracks occurring near bolts on a steel structure are detected. However, users can extend the proposed visual inspection framework to conduct other types of visual inspection. For example, suppose that corrosion or crack damage in gusset plates is damage of interest. Gusset plates become "objects" in this study and techniques suggested in chapter 2.2.1~2.2.3 can be applied to extract images of individual gusset plate from different angles. Users analyze images of all gusset plates in a test bridge by applying a corresponding crack detection technique.

The major contribution of the technique is to propose a new approach to automated visual inspection using a large volume of images. Many previous researchers have focused

on detecting cracks from a few images taken from set positions where cracks are visually clear. However, automatic image collection using aerial cameras or other equipment, would not guarantee that favorable images would be obtained due to the uncertainty in crack's location and direction. Instead, the proposed technique begins by searching damage sensitive areas from a large pool of images. By detecting these areas from many different viewpoints, detectability of damage can be dramatically increased even it is small, and false-positive alarms can be reduced by limiting searching areas.

The remainder of chapter 2 is organized as follow. Chapter 2.2 starts from the brief overview of the proposed approach and provides technical details about image acquisition, object detection and grouping, and crack detection. Experimental descriptions and results are presented in chapter 2.3. Chapter 2.4 includes a conclusion.

2.2 Methodology

The overview of the proposed technique is shown in Fig. 2.2. First, in Fig. 2.2(a), images of the structure from many angles are collected using image acquisition equipment. This step may involve one or more possible ways of image collection such as aerial cameras or inspection robots. Second, in Fig. 2.2(b), target structural components called objects, which are susceptible to cracking damage, are detected and extracted from the images. The object patch indicates one such object and its nearby area where cracking damage is more likely to present. Third, in Fig. 2.2(c), common object patches (corresponding to the same object) across the collection of images are matched and grouped. Finally, in Fig. 2.2(d), the proposed crack detection technique diagnoses that a crack exists in the structural components. In this study, fatigue cracks initiated from bolt holes are chosen as target damage. Thus, the terms of "structure" and "object" in this chapter indicate the bridge and bolt, respectively. However, the proposed technique can be easily generalized to detect cracks from any structural components such as joints or welded areas and on any structure, not limited to bridges. The latest techniques in computer vision are implemented to increase the quality of object detection, object grouping, and crack detection.

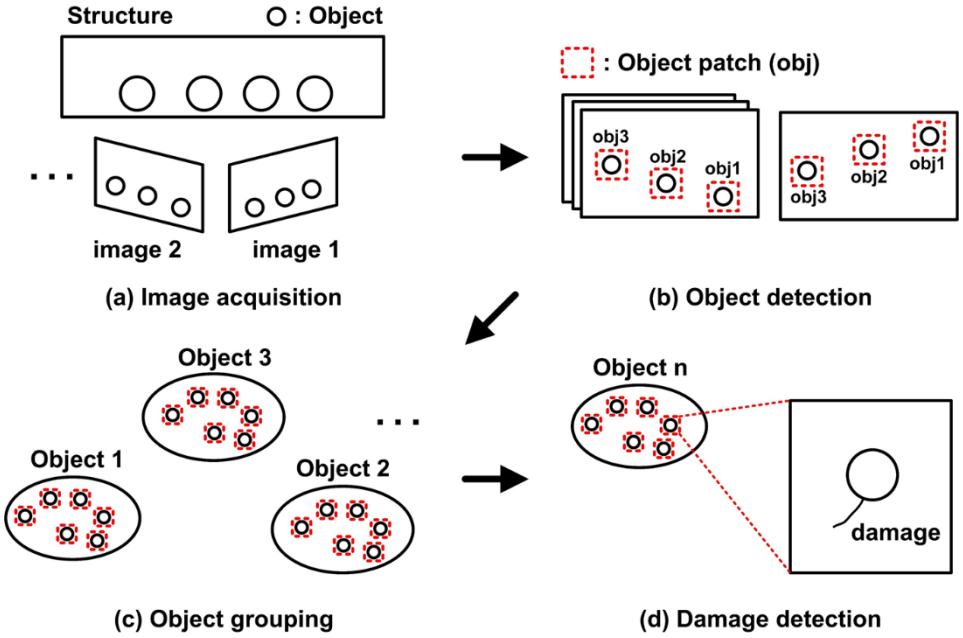


Figure 2.2 Overview showing the procedure for the proposed damage detection technique: (a) acquisition of images from multiple angles, (b) detection of object patches which may include damage, (c) grouping of same objects across images, and (d) detection of crack damage initiated from objects.

2.2.1 Image acquisition

For image acquisition, the UAV flies under or over civil structures by following a predetermined flying path, and cameras installed capture scenes of bridges consecutively. Some guidelines for image acquisition are suggested here for the best performance of the proposed technique but are not required: (1) Images are highly focused and have sufficient resolution so that objects and cracks are clearly shown in images. (2) Angles between the camera and bridge, called tilt or perspective angle, should not be large. Since objects on the bridge are commonly presented within a short distance of each other, their scenes may overlap on the images under a large perspective angle, making object and crack detection difficult. However, angle variation of the camera is necessary due to the dependency of the crack's appearance on the angle of the images. (3) The distance between the UAV and bridge stays roughly constant. The number of scales over which the images need to be searched can be reduced by known approximate distances and physical bolt sizes (Jahanshahi et al., 2013). The computation in detail is also shown in chapter 3.2.1. This results in increasing object detection rates or decreasing false-positive errors with low

computation time and (4) GPS data of each image is recorded and saved for approximated estimating crack locations.

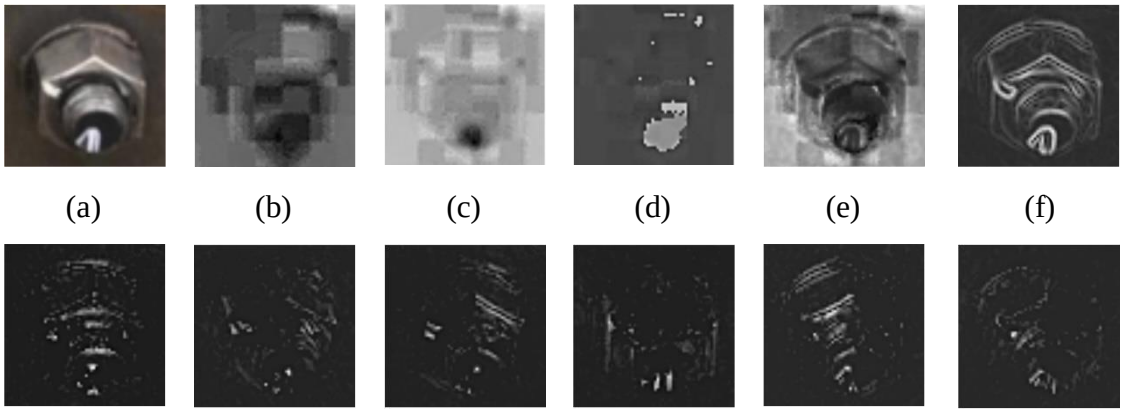
2.2.2 Object detection

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

In this study, an integral channel based sliding window technique is applied over multiple scales of images. The sliding window technique uses a fixed rectangular window that slides over the images to decide whether the window contains an object or not. To make this judgment, features are extracted from each window. Here, the channel image indicates the 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 a histogram of the gradient with six orientations (HOG). The details of HSV and LUV color formats can be seen in reference (Schanda, 2007). HSV and LUV color spaces can separate colors from brightness, which is not robust under lighting variation. Thus, the V component in HSV and the L component in LUV, which are brightness terms, are ignored from channel images. To consider the object shape, a gradient magnitude and HOG are used. These gradient features have been proven in many applications for object detection (Dalal and Triggs, 2005; Dallar et al., 2009; Felzenszwalb et al., 2010). Fig. 2.3 shows channel images of the bolt. Channel images in a grayscale from Figs. 2.3(b) to (I) are computed from the original RGB image, Fig. 2.3(a). The intensity of these images vary depending on positions or color, and this variation represents the unique feature of the object patch. For example, Figs.

2.3(g)-(l) shows how different edges of the bolt are highlighted depending on gradient direction.

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



(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k) (l)

Figure 2.3 Examples of channel images of a bolt: (a) original RGB (herein grayscale), (b, c) U and V components in LUV, (d, e) H and S components in HSV, (f) gradient, and (g-l) histogram of gradient with 6 orientations (0°, 30°, 60°, 90°, 120°, 150°).

Based on these features, a robust classifier is designed to determine whether features at a certain window in a test image indicate an object or not. In this study, a boosting algorithm is implemented to produce a robust classifier. Boosting is a way of combining many weak classifiers to produce a strong classifier. By updating different weights of weak classifiers adaptively depending on misclassification errors, the optimum strong classifier that minimizes misclassification errors can be obtained. There are several boosting algorithms introduced in the literature, but in this study, the gentle boost algorithm, proposed by Friedman, is used because it is known as simple to implement, numerically

robust and experimentally proven for objection detection. The details of the gentle boost algorithm and sample codes can be found the following references (Friedman et al., 2000; Torralba et al., 2004).

2.2.3 Object grouping

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

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

In this study, these two techniques are simply integrated for better performance on matching objects. Suppose that the object patch in the first image, called target object patch, matches with one of object patches in the second image. The object patch in the second

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

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

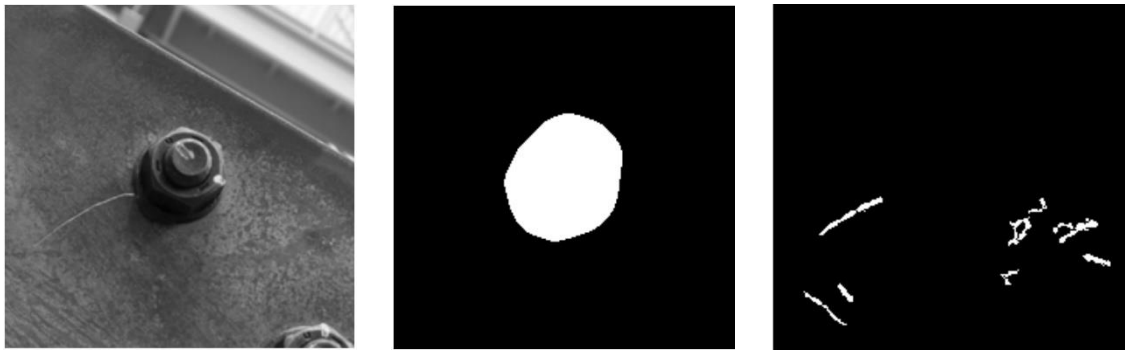
A very well-known community detection technique called modularity maximization is applied in this study (Clauset et al., 2004). Modularity is defined as a measure of the quality of particular division of a network into community. The optimum community structure is obtained by iteratively amalgamating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

2.2.4 Crack damage detection

The proposed crack detection technique is applied to all groups of object patches, which are found in the previous step, to decide whether cracks exist or not. Before introducing the technique, the type of crack referred to in this study should be clearly defined. The

crack visually has a very sharp edge and an almost straight line, and initiates from bolt holes on steel plates because bolt holes inherently have initial discontinuity in its circumference. A detectable crack in an image is one that can be detected by human vision. Such “preliminary” information helps to detect the crack of interest by filtering out non-crack edges.

The first step in the proposed approach is to remove the bolt areas from object patches. Each object patch found in the previous step includes both the bolt and its nearby area. The region of interest on each patch is the area connected to the bolt, but not the bolt itself. The benefits of removing the bolt area from the object patch is that crack-like edges are not falsely detected from the bolt area and also a threshold boundary for true crack edges can be made, which will be mentioned later in this chapter. Detection of the bolt on object patches is carried out by edge detection and binary morphology. The procedure is as follows: (1) A median filter is first applied to object patches to remove edges from real cracks or other textures, which are connected to the bolt. (2) Canny edge detector generates the edge image to detect boundaries of the bolt. Then, a binary edge map is obtained using a predetermined threshold. (3) Dilate operators using a disk structural element are applied to the edge map to fill gaps between edges and then, the convex hull of each connected entity is computed. Through this operation, several separated binary entities are generated from a bolt or background. (4) A true binary entity indicating the bolt is selected if it includes the center of the object patch. This criterion is reasonable because the object detector detects the bolt to be presented at the center of the window. (5) Finally, marginal pixels are added to the boundary of the detected entity so that it includes full appearance of the bolt. Example images are shown in Figs. 2.4(a) and (b). The Fig. 2.4(a) is the detected object patch from chapter 2.2.2 and 2.2.3. Following the above procedure, the area of the bolt in the object patch is detected in Fig. 2.4(b). Subsequent analysis is conducted on the area of the outside of the detected binary entity (object area).



(a) (b) (c)

Figure 2.4 Examples of bolt area and crack-like edge detection: (a) object patch including a bolt and its nearby area, (b) detection of a binary entity (white area) indicating bolt, and (c) crack-like edge detected from the outside of the object area using the Frangi filter.

The second step is detecting crack-like edges. In this study, the Hessian matrix based vesselness measurement technique, called Frangi filter, is used to detect crack-like edges. The approach is based on the fact that a crack on steel has a thin and bright line, similar to the appearance of vessel in medical images. The Hessian matrix based edge detector does not produce double edges making good localization, and thus multi-scale crack detection is possible (Frangi et al., 1998). The brief outline of the Frangi filter is first, a Hessian matrix of the image is computed using a Gaussian derivative at multiple scales. Then, two eigenvalues, λ_1 and λ_2, of the Hessian matrix each pixel of the image are derived ($|\lambda_1| > |\lambda_2|$). An ideal crack edge on image is $\lambda_2 \approx 0$ and $|\lambda_1| \gg |\lambda_2|$, and the sign of λ_1 is negative because crack is a bright edge. The strength of crack-like edges, V, is defined as,

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

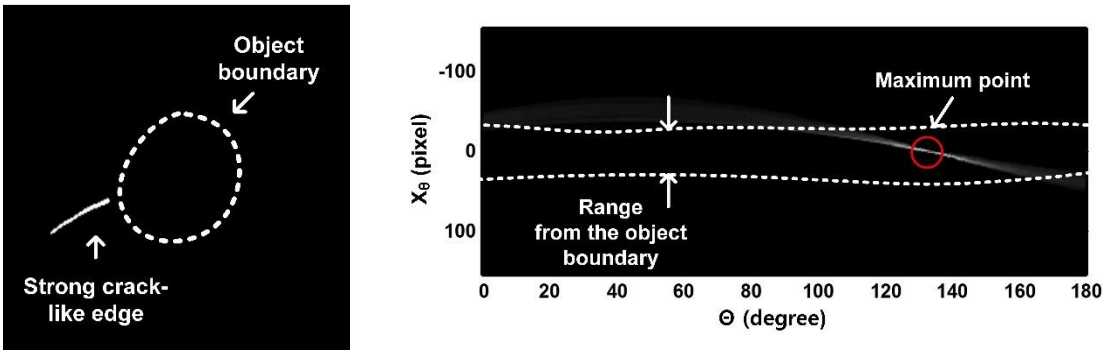
where β and c are user-defined parameters, $R_B = \lambda_2 / \lambda_1$, and $S = \sqrt{\lambda_1{}^2 + \lambda_2{}^2}$. For example, if the edge is close to the ideal crack edge, R_B goes to zero and S becomes large, and thus V is close to 1, which is the maximum of V. The final edge map can be obtained by thresholding and removing edges connected to the border of the object patch. Fig. 2.4(c) shows the result of crack-like edge detection from the outside of the bolt area in the object

patch, Fig. 2.4(a). The technical details about the Frangi filter are provided in (Frangi et al., 1999).

The remaining two steps are to detect real crack edges from non-crack edges using prior knowledge of crack's appearance because the Frangi filter detect several edges having similar appearance with cracks, as shown in Fig. 2.4(c). The third step is filtering out spurious non-crack edges using region-based shape characteristics. All connected components in the binary edge image, obtained in the previous step, are identified using connected component labeling. Among these connected components, the crack-like edges are differentiated using their shape descriptors. For example, fat and short edges or zigzag are not real crack edges. The shape descriptors used in this study is eccentricity, which is defined as the ratio of major to minor axes of a connected component and evaluate elongation of edges (Yang et al., 2008). A threshold, which represents the minimum eccentricity, is first set to filter out non-crack edges. Then, one strong edge line, which has the largest regional area, is detected. Fig. 2.5(a) shows the detected edge line from Fig. 2.4(c) using these two criteria.

As a final step, the detected crack-like edge, called strong crack-like edge, is evaluated as to whether it is the true crack or not. This decision is made based on the assumption that crack is initiated from the bolt hole, and has a relatively straight line as mentioned in the previous chapter. 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 on the edge, will cross the axis within this range. This concept is successfully implemented using the Radon transform. The Radon transform in two dimensions is the integral projections of images along specified direction (Deans, 2007). In the computer vision community, the Radon transform is used for line detection such as the Hough transform. In this application, the maximum value of the Radon transformation image indicates the direction and position of the detected edge line. Fig. 2.5(a) shows the detected crack-like edge and object boundary computed from Fig. 2.4(a). Radon transformation of the strong crack-like edge image is shown in Fig. 2.5(b). The dotted line is the range computed from projecting object boundary. The maximum value of the transformed image indicates the edge's direction and location on an angled

axis, respectively. The axis angle of the maximum point in Fig. 2.5(b) is around 140° and is perpendicular to the line of the edge. The true crack is determined if the maximum point is located inside of the range obtained by projecting object boundary onto this axis. This criterion may not perfectly classify all true cracks from crack-like edges because non-crack edges also exist in the similar position of true cracks. However, it can successfully remove most non-crack edges.



(a) (b)

Figure 2.5 Examples of crack detection using Radon transformation (a) one strong crack-like edge with the object boundary and (b) Radon transformation of the strong crack-like edge and the range computed from the object boundary.

2.3 Experiment Validation

2.3.1 Description of the experiment

A large scale, rusted I-beam having 68 bolts, as shown in Fig. 2.6, is used for validating the proposed technique. As shown in introduction, a real fatigue crack is almost visually similar to a sharp scratch. Instead of producing real fatigue cracks on the beam, two artificial scratches are made with an awl at locations A and B in Fig. 2.6. Test images of the beam are taken using a Nikon D90 camera with 18-105 mm lens, and zoom and flashing functions are not utilized. All 72 test images having 4,288 x 2,848 resolution are sequentially taken at roughly 2~3m working distance, which is between the camera and beam, and do not have much tilt angles due to the issues in chapter 2.2.2. For the object detection and grouping, downsized images with low resolution having 1,716 x 1,149 are used for fast computation. However, once the areas of object patches are detected, those are cropped from the full resolution images for accurate crack detection.

Five randomly selected test images among the 72 images are used for training. A square of 68 object patches are cropped from these images. 144 negative patches, which are three times of a number of positive patches, are randomly cropped from the non-object areas. 500 Haar-like features in each object patch are randomly selected with respect to positions, sizes, and types. Thus, 5,500 features can be generated in each object patch, given by the product of 11 channel transforms and 500 Haar-like features. A "strong" classifier is trained with 200 weak classifiers, which are enough to ensure convergence of classification.

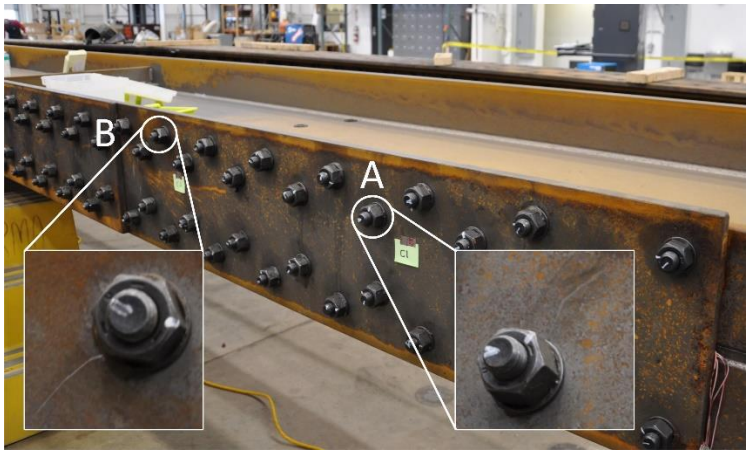


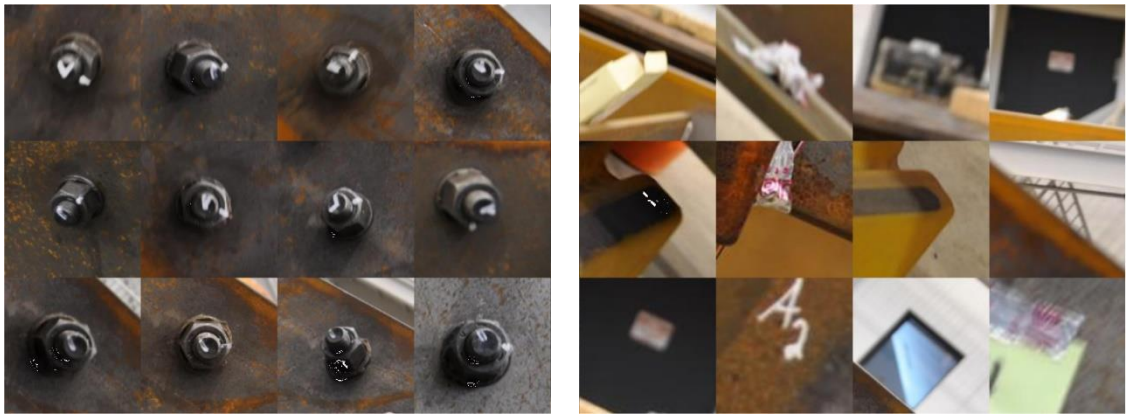
Figure 2.6 A test steel beam including 68 bolts.

For the sliding window technique, rough estimation of the size of the object patches in the images in advance can reduce computation time and false-positive detection. For example, suppose that the size of object patches in image is unbounded, all scales of test images are scanned, which is a very time consuming process. In this study, based on the physical bolt size and approximated working distance, the range of the bolt size on the downsized images is found to be 64~128 pixels. Consequently, the final scale is set to 2 with 2 1/5 scale steps, causing a total of 6 scales. For object detection, test images are downsized with these scales so that the different sizes of bolts in the images can be detected. All algorithms proposed in this study are implemented using MATLAB. VLFeat, which is an open source library of computer vision algorithms, implements the SIFT algorithm with MATLAB interfaces (Vedaldi and Fulkerson, 2010). Each image makes pairs with the next 5 images in the set of sequential images. The matching criterion of keypoint descriptors proposed by VLFeat, called VL_UBCMATCH, is used with a threshold of 2.5. If the

number of matches is less than 30, the fundamental matrix of this pair of images is not computed and this pair of images is not considered further during object matching. Using matched keypoints of an image pair, a fundamental matrix for this pair is computed using a normalized eight-point algorithm and RANSAC. If the number of remaining matches is less than 25, this pair is also not considered for object matching. The threshold of the epipolar constraint for matching is set to 181 pixels, which is the diagonal distance of the maximum size of object patches. For grouping object patches, if a group includes less 3 than object patches, this group and its object patches are removed and excluded from a list of object patches for damage detection. 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 to the high and low thresholds to products of 0.2 and 0.5 to the median value of the grayscale object patch, respectively. The size of the median filter for removing the crack edge for detecting object areas is a 5 x 5 square. A disk structure element having eight-pixel diameter is used to dilate images, and 10 pixels of the border margin are used. For the Frangi filter, parameters β and c are set to 0.5 and the maximum Hessian norm, respectively, presented in the original work and the threshold is 0.90 (Frangi et al., 1998). Scales of the filter is up to 3 pixels with 0.2 pixel step, which means the edges less than 6 pixels are more enhanced. For the shape descriptor, the minimum eccentricity is set to 8. Note that all parameters and thresholds are not automatically determined. Users should tune the parameters for different uses based on ones that authors suggested in the training step and by gaining experience. Values used are selected based on tests with an initial set of training images. However, this simple process is conducted just once, and then the selected values are used for future visual inspections without further tuning.

2.3.2 Description of the experiment

A total of 1,326 bolts are shown in all of the collected test images. Bolts having partial occlusion or distracted by foreign objects are removed before counting. The resulting object detector proposed in chapter 2 achieves a 98.7% detection rate (1,310 object patches) and a 6.8% false-positive detection (91 non-object patches). The proposed object detection technique successfully attains a high rate of true detection and minimize false-positive rate. Fig. 2.7 shows samples of detected bolts and false-positive detection.



(a) (b)

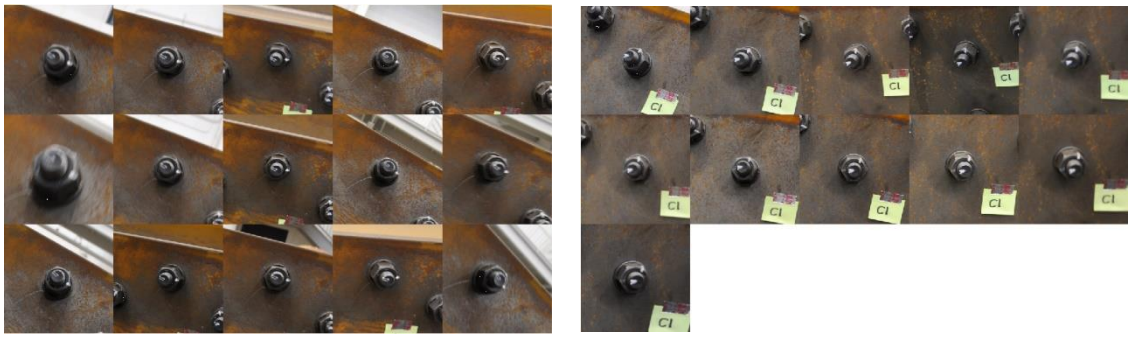
Figure 2.7 Examples of bolt detection results from test images: (a) true detection of bolts and (b) false-positive detection.

Table 2.1 Results of object detection and grouping

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

Table 2.1 shows the results of object detection and grouping. Based on the proposed matching and grouping techniques, 2,922 connections (= matching) between nodes (=object patches) are found and 77 communities (= groups) are detected. All 68 bolts are successfully grouped and 5 non-object groups are produced, which are visually similar to bolts. Here, 5 non-object groups come from false-positive detection in the object detection step. Since the non-object patches are consistently detected across images, these are finally classified as a group. To avoid this problem, these non-object patches are used for training as negative samples, not to be detected in future testing. Moreover, there are 4 communities overlapping, which means two communities indicate the same bolt. These discrepancies come from weak connections between sets of nodes in a same group, and could be overcome by increasing the number of pairs of each image for matching, but it would be computationally expensive. A total of 1,147 object patches from 77 communities are grouped and the remaining ones are removed due to a lack of nodes in their allocated group. Fig. 2.8 shows groups of object patches having a crack associated with locations A and B

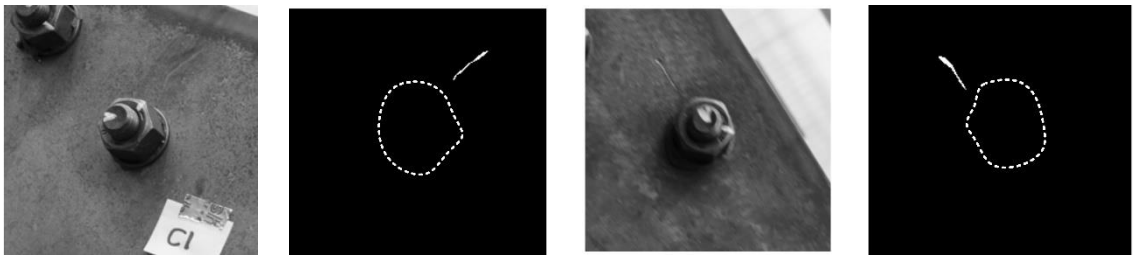
in Fig. 2.6. Detected object patches prove the necessity of images from different viewpoints. Only a couple of patches show clear cracks that are visually recognizable.



(a) (b)

Figure 2.8 Resulting groups of object patches at crack locations A and B in Fig. 2.6.

The proposed crack detection technique is applied to object patches in each group. There is a trade-off between true and false-positive detection depending on the threshold of the Frangi filter and Canny edge detector because these two parameters determine crack-edges and the decision boundary, respectively. However, regardless of these variations, cracks are constantly detected from at least one of the object patches in the groups showing locations A and B. Example results are shown in Fig. 2.9. Fig. 2.9(a) and Fig. 2.5(a) represent object patches and crack detection with object boundaries from locations A and B. The lines of true crack edges pass through the object areas. Fig. 2.9(b) is one of the false-positive detection results. The machine generated scratch next to the bolt, which is not induced by authors, is visually similar to a real crack. However, it is not a true crack and was originally present in the test structure. This is a limitation of the proposed technique, but based on visual information only, this scratch would be challenging to differentiate from a real crack, even by human inspectors.



(a) (b)

Figure 2.9 Examples of crack detection results: (a) true crack detection and (b) false-positive crack detection.

2.4 Conclusion

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

CHAPTER 3. AUTONOMOUS IMAGE LOCALIZATION FOR VISUAL INSPECTION OF CIVIL INFRASTRUCTURE

3.1 Background

This study was initiated by exploring the feasibility of several established *visual inspection techniques for use with a large volume of images automatically collected by sensing platforms like UAVs*, in terms of their potential for meeting the objectives of this futuristic vision-based visual inspection of civil infrastructure. Despite the dramatic advances made in vision sensors and UAVs, a significant gap is identified in their practical implementation at this time: a significant number of erroneous false-positives (Yeum and Dyke, 2015). Automatic image collection using UAVs provides no guarantee to selectively collect favorable images for visual inspection. The limitations are mainly due to insufficient accuracy of camera location (GPS) and rotational angle measurements. *Only a very small fraction of the images or region in each of the images will actually be relevant and useful for visual inspection purposes.* More importantly, due to large variations in the image scales, it is inevitable that existing techniques will be applied over multiple image scales (image pyramid) or a sliding window (Lindeberg, 1994; Viola and Jones, 2001; Dalal and Triggs, 2005), which are often computationally expensive and hindered by large-size files. In such circumstances, although existing damage detection techniques may not miss the presence of damage, a large quantity of false-positives will also be detected among the few true-positives detected. In other words, a high portion of the images found to contain damage will not actually contain damage. Of course, the goal of inspection is structure safety, and the detection of true damage far outweighs the possibility of false-positive errors in the visual inspection. However, *a high ratio of false-positive images means that human inspectors will need to spend valuable time looking through a large number of irrelevant images to find the true positives, leading to wasteful consumption of the very time that was meant to be saved by using automation*. Overall, the high likelihood of false-positives when these methods are used in isolation reduces the trustworthiness and efficiency of these methods. Alternative procedures, such as using pre-processing techniques for localizing relevant areas on images, need to be incorporated into the procedures.

In this study, an image localization technique is developed to automatically extract the regions of interest (ROIs) on each of the collected images named RILVI (ROI Image Localization for Visual Interrogation). The ROIs on the images are computed based on the 3D geometry of the images with respect to targeted regions of interest (TRIs) in the structure. The geometry is computed using a structure-from-motion (SfM) technique. Vulnerable components or areas in a test structure to be examined would be assigned as the TRIs. The ROIs are then automatically cropped and scaled before applying vision-based visual inspection techniques, ensuring that efficient analysis and reliable outcomes are facilitated.

An important contribution of the proposed technique is to develop a technique to facilitate real application of existing damage detection techniques on large volumes of actual images collected from UAVs. In spite of the variety of automated vision-based visual inspection techniques proposed in civil engineering, unfortunately, real implementations using UAV images are quite limited to date due to large variations in scale, pose, lighting conditions, and background at the time of image collection. Rather than processing entire images, RILVI focuses on and localizes ROIs, which will then be used for visual inspection. Since the ROIs on images are estimated based on geometric relationship, not by object recognition, any shape and size of components and regions may be assigned as the TRIs. Moreover, extracting a small set of ROIs can greatly reduce the rate of false-positive errors that are likely to be generated from the analysis of irrelevant areas. Moreover, from a different perspective, as the transition is made to automated visual inspection, this technique will also serve to assist human-based visual inspection. By avoiding the unnecessary processing of immense volumes of images, RILVI provides a confined number of valuable localized ROIs, allowing intensive visual inspection without the distractions generated by irrelevant areas on images.

The remainder of this paper is organized as follows. Chapter 3.2 begins with the overview of the proposed method and discusses detailed steps including image acquisition, projection matrix estimation, 3D coordinate transformation, and the localization of ROIs on images. In chapter 3.3, the capability of the RILVI technique is demonstrated using a full-scale highway sign truss for the application of weld inspection. Chapter 3.4 includes the summary and conclusions.

3.2 System Overview

The overview of RILVI is presented in Fig. 3.1. The objective of the technique developed here is to automatically extract ROIs from the large volume of original images collected by UAVs. The ROIs are the portions of each of the images that contain the TRIs. The TRIs can be anywhere in the structure, but they would likely be assigned to encompass the failure-sensitive areas set forth in the inspection manual. The locations of the TRIs are easily obtained from a geometric model (e.g., drawings or 3D model) or from actual measurements from a physical structure in advance.

First, in Fig. 3.1(a), images of the structure are collected from many viewpoints and positions using UAVs. Several variables should be considered to be sure to collect suitable images for visual inspection, including flight configuration (e.g. flight speed or flight path), camera setup (e.g. shutter speed or focal length), and/or some guidelines that will be discussed later for successful implementation of the proposed technique. Second, in Fig.3.1(b), a projection matrix (camera matrix) of each image is estimated using SfM. The projection matrix includes information about camera location and orientation in 3D coordinates (extrinsic orientation) and internal camera parameters (camera calibration matrix), describing the mapping of 3D points in the world to 2D points in each image (Hartley and Zisserman, 2004; Snavely et al., 2008; Szeliski, 2010). In Fig. 3.1(c), this step is performed to transform the coordinate system of the constructed SfM model (defined as the SfM coordinate) to the coordinate system which is used to define the coordinate information of the TRIs (defined as the TRI coordinate). Then, the projection matrix and TRI are represented in the same coordinates. Lastly, in Fig.3.1(d), the ROIs are extracted from the original images using the geometric relationship between the cameras (image) and the TRIs. The overall process here is designed to be fully automated without the need for any human intervention during the image collection and processing.

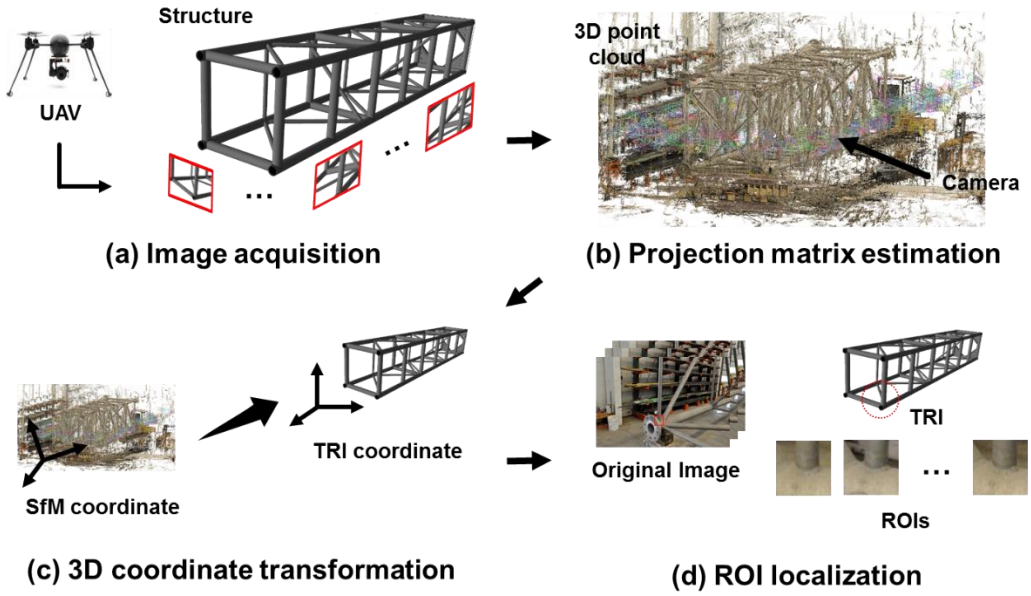


Figure 3.1 Overview of the technique developed (RILVI): (a) acquisition of images from multiple viewpoints and positions, (b) estimation of a projection matrix in each image using a structure-from-motion (SfM) technique, (c) 3D coordinate transformation for alignment and scale matching, and (d) extraction of the regions of interest (ROIs) on all images collected in (a).

3.2.1 Image acquisition

The manner in which the images are acquired will strongly influence their usefulness for the subsequent steps in the technique, and ultimately for damage detection. Herein a simple model of a camera is considered, and the conditions to yield favorable images are outlined. During image acquisition, two major guidelines should be followed for successful implementation of the technique developed here.

First, the most important guideline is to collect images with good visibility of the TRIs. Visibility of the TRIs on a subset of the images is a vital prerequisite for automated vision-based visual inspection. Visibility of the TRIs on UAV images is mainly affected by a couple of factors: working distance, motion blur, and occlusions. The maximum allowable working distance is computed to guarantee sufficient size of the ROIs. The ROI size decreases as the working distance increases. Fig. 3.2 and Eq. (3.1) explain how to determine the maximum allowable working distance for designing UAV flight paths under given camera model parameters, and the physical size and minimum resolution of ROIs:

$$WD = \frac{FL \cdot SR}{SS \cdot TP} \cdot \frac{TS \cdot \sin(\beta - \alpha)}{\sin(\pi - \beta)} + \frac{TS \cdot \sin\alpha}{2} \tag{3.1}$$

where *WD* (mm) is the maximum allowable working distance, called the distance threshold, *FL* (mm) is a focal length, *SR* (pixel) is the resolution of the whole image, *TS* (mm) is the physical size of the TRI on the TRI plane, which is a/the major plane of the TRI in the 3D space, *SS* (mm) is the camera (image) sensor size, *TP* (pixel) is the minimum size of the ROIs on images required for visual inspection, α (radian) is the angle between the image and the TRI plane, and β (radian) is the angle between the image plane and the ray from the ROIs boundary on the image. *TP* is often computed using a required accuracy represented by pixels per physical length (e.g. 1 pixel/1 mm) and thus a TRI with a large physical size will require high resolution on the image. Since α and β are randomly determined based on the physical geometry and locations of the ROIs, *WD* is computed by simulating various combinations of α and β. An example calculation for *WD* will be illustrated in chapter 3.3. Note that *WD* is understood as a threshold for the distance between the TRIs and cameras to ensure valid and useful ROIs will be obtained. For a collection of images using UAVs, a larger *WD* is always preferred for safe flight by increasing *FL* and/or *SR* or decreasing α and/or *SS* if sufficient *TP* is guaranteed. However, since a large *FL* will result in a small field of view, the images may not have enough overlap with each other, degrading the SfM outcome. Also, a large *SS* has good low-light performance and sharpness to describe fine details, but *SS* typically increases with *SR*. Thus, both high *SR* and large *SS* are recommended (e.g. full frame sensor with high resolution). Note that other environmental conditions (e.g. motion blur, low light conditions) may degrade the quality of the images (may require higher *TP*), and α and β cannot be known before flying/acquisition. Therefore, *WD* is not a precise value that users need strictly follow. However, enforcing this threshold will greatly help to plan a flight path that facilitates collection of more useful images.

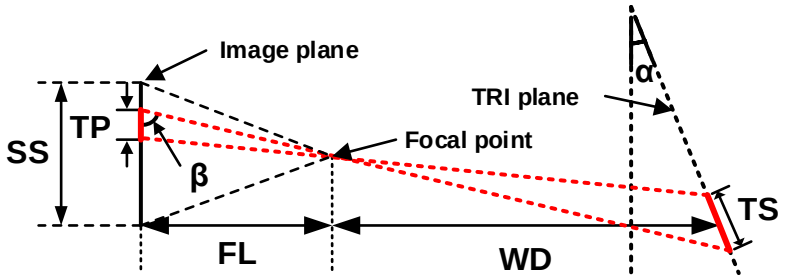


Figure 3.2 Estimation of the maximum allowable working distance in a pinhole camera model.

Motion blur is an effect that occurs when an object being recorded moves 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. However, these problems can be alleviated by stabilizing sudden movements using a multi-axis gimbal, and configuring a fast shutter speed (New America, 2015). Taking pictures under good weather (lighting) conditions is also recommended to increase the shutter speed of the camera.

Occlusion is an unavoidable corruption mode frequently occurring in a structure having complex geometry. Structural components will 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 different viewpoints. Including several angle variations in image collection are also important for successful vision-based visual inspection (Yeum and Dyke, 2015).

Next, image collection guidelines are also delineated here for the SfM procedure to ensure that the projection matrix in each image is accurately estimated. Projection matrices are computed using corresponding feature points shared across collected images. Sharing many features between images can reduce erroneous estimation. As a rule of thumb, at least 60% overlap is highly recommended between neighbor images (Kraus, 2011). However, configurations yielding such a high overlap cannot easily be attained under a close working distance, with a long time interval between images, or at fast flying speeds because such conditions produce large scene changes between images. Collecting a large number of high-resolution images with a fine time interval is often recommended but, adversely, it is computationally expensive. Thus, users need to balance these parameters to

achieve accurate and efficient implementation of SfM. Additionally, capturing both the target structure and stationary background in the images may be helpful to increase the overlap between images as the background is usually far away from the camera location, producing slow scene changes between images. Thus, the overlaps present within the sequence of the images are increased (Ozden et al., 2010).

3.2.2 Estimation of the projection matrices

The next step in the technique is to estimate a projection matrix of each camera. The projection matrix describes the mapping of a pinhole camera from 3D points in the world to 2D points in an image. The projection matrix includes intrinsic orientation including focal length or principle points, and extrinsic orientation including rotation angles and location in 3D. SfM automatically computes the 3D geometry of the scene (point cloud), projection matrix, and lens distortion parameters. The major advantage of SfM is that all these parameters and data are generated without the need to augment the structure with networks of targets with known (or even unknown) 3D positions, which are generally required in conventional photogrammetry-based survey methods. Once users collect a set of good photos by following appropriate guidelines, projection matrices can be estimated without preliminary measurements or calibration (Hartley and Zisserman, 2004; Snavely et al., 2008; Szeliski, 2010).

SfM has had a broad array of applications ranging from image-based 3D modeling to aerial mapping applications. In civil engineering, SfM is often implemented for survey applications using UAVs (Westoby et al., 2012), but it has also been explored for applications related to geometric damage detection based on 3D reconstruction (Torok et al., 2014), crack quantification (Jahanshahi et al., 2013) or tunnel lining image stitching (Chaiyasarn et al., 2016). Recently, SfM has become a well-established technology and some popular commercial software packages exist, such as Pix4D (Pix4D, 2016), Photoscan (Agisoft, 2016), or ContextCapture (Bentley, 2016), and non-commercial software, such as VisualSfM (Wu, 2013), OpenMVG (Moulon et al., 2012). Regardless of the differences in functionalities in these tools, and 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 by several internal and external camera

parameters) and the 3D point cloud. Almost all commercial and non-commercial software can export these outputs in a readable format. In our study, the projection matrices are only used to analyze the geometric relationships between cameras (images) and the TRIs.

The projection matrix is a 3x4 matrix, denoted as P, and the following relation holds

$$x_i = P_i^S X^S \quad (i = 1, 2, ..., n) \tag{3.2}$$

These matrices are expressed in homogeneous coordinates (Hartley and Zisserman, 2004). X^S is a point in the 3D SfM coordinate system (a 4-dimensional vector) and the superscript s indicates the SfM coordinate, which is an arbitrary coordinate assigned by the software if the coordinate is not designated. P_i^S is the projection matrix of the ith image among a total of n images. x_i is a point on image i and is typically defined as pixel values (a 3-dimensional vector). Lens distortion should be corrected in advance so that all images hold the relationship in Eq. (3.2) based on the pin-hole camera model. Simply, Eq. (3.2) represents the fact that any arbitrary 3D point can be mapped to each image provided the projection matrices are defined in identical coordinate systems.

3.2.3 Transformation of the coordinate system

When no GPS data is used for initialization and no coordinates are assigned to the images, projection matrices defined in the SfM coordinate system do not have scale, position, and orientation information. The SfM coordinates should be transformed into the known coordinate system where the locations of the TRIs are defined (the TRI coordinates) to use Eq. (3.2). For this transformation, use of the same 3D points is required, with both defined in the different coordinate systems of the SfM and TRI.

In typical scenarios, fiducial markers are installed on the structure with known locations in the TRI coordinate system, such as corners or midpoints of elements. The fiducial markers are designed to have clear visibility on the images for accurate localization. Having fiducial markers visible on two or more images produces their 3D locations in the SfM coordinate using triangulation (Hartley and Zisserman, 2004). Users can manually identify and relate the points of these fiducial markers on images or use coded fiducial markers for automated matching and correspondence, as done in the software of Photomodeler or iWitness) (iWitness, 2016; Photomodeler, 2016). However, in the case of

coded target applications, the target size should be large enough so that the code (identity) can be recognized using image processing algorithms. If the locations of the TRIs are defined as GPS coordinates (e.g. WGS84), initialization of the GPS data stored in the acquired images would be helpful, but additional calibration using known 3D locations are still required due to current limitations in GPS accuracy (Pix4D, 2016).

Once the 3D points in SfM and TRI coordinates correspond, a 3D coordinate transformation (scaling, translation, and rotation) is performed using Horn's method, which is a closed-form solution of the absolute orientation (Horn, 1987). The transformation matrix, denoted as M (a 4 x 4 matrix) in the homogeneous coordinate system, is computed, which can map a point in the SfM coordinate system to a point in the TRI coordinate system. The following relation holds

$$X^T = MX^S$$

where X^T and X^S are any corresponding 3D points (a 4-dimensional vector) defined in TRI and SfM coordinates, respectively. For the 3D coordinate transformation, a total of seven unknown parameters, including three degrees-of-freedom for both translation and rotation and one degree-of-freedom for scaling, can be estimated using more than three corresponding points. As a rule of thumb, those points are evenly located over the structure and visible in many images to minimize estimation errors in the transformation (Hartley and Zisserman, 2004). The projection matrix is also transformed as follows:

$$P_i^T = P_i^S M^{-1}$$

where P^T is a projection matrix defined in the TRI coordinate system after transformation from the SfM coordinate system. This relation can be derived from the fact that the corresponding 3D points in the TRI and SfM coordinate are projected to the same image points, represented as $P^T X^T = P^S X^S$.

3.2.4 Localization of the ROIs on images

Using the prior steps, the projection matrices and the TRI locations are now represented in the identical TRI coordinate system. 3D points in the TRI coordinate system and their projected points on the images are associated by the relationship in Eq. (3.2) (although the superscript S is replaced with a T after the coordinate transformation). With this, users can define any shape and location for the TRIs using known 3D points. These can range from

a line (e.g. plate joints, weld connections), surface (e.g. steel plate for corrosion inspection, concrete deck for cracking inspection), to a space or 3D object (e.g. bearing, component). A simple geometry like a line or rectangular flat surface requires only a few 3D points to describe its geometry (e.g. two 3D points define a 3D line). However, it would be a challenge to describe the complex geometry of more intricate TRIs such as a complex polygon, a curved surface or a circular line, which may require thousands of 3D points or even a mathematical representation in 3D. For example, as our target TRIs in chapter 3.3, a weld connection between a cylindrical brace and the main chord forms a circular line in 3D space. It is hard to define the exact shape of the weld (welding line between two tubular components) using 3D points, and it would be time-consuming to manually assign all those points on the weld. The localization of images in this study is not intended for extracting the exact shape or boundary of the TRIs. Rather, our goal is to find bounding boxes on images where entire regions of TRIs are tightly enclosed for further inspection, and those bounding boxes become ROIs in this study.

In this study, a generalized and straightforward solution is designed by implementing a virtual sphere. As shown in Fig. 3.3, a virtual sphere is used for estimating the bounding boxes of the TRIs on the image, which tightly encloses them. A virtual sphere defined at each of the TRIs is projected to the images, and find the location and size of the bounding box. The benefits of this virtual sphere in the definition of the TRIs are that only two parameters, a 3D center location and a radius, are required to define each of the TRIs, regardless of the actual complexity of their geometry. For example, when straight or curved lines are assigned as TRIs, its 3D center location and the radius of the virtual sphere that encloses them are easily obtained. Ultimately, this virtual sphere can apply to any arbitrary shape (region) for the TRIs. The idea of the virtual sphere is developed here for ready implementation to a wide variety of inspection problems. However, if more precise 3D point information of the TRIs is known, the ROIs on the images can be directly computed from Eq. (3.2) without using this virtual sphere.

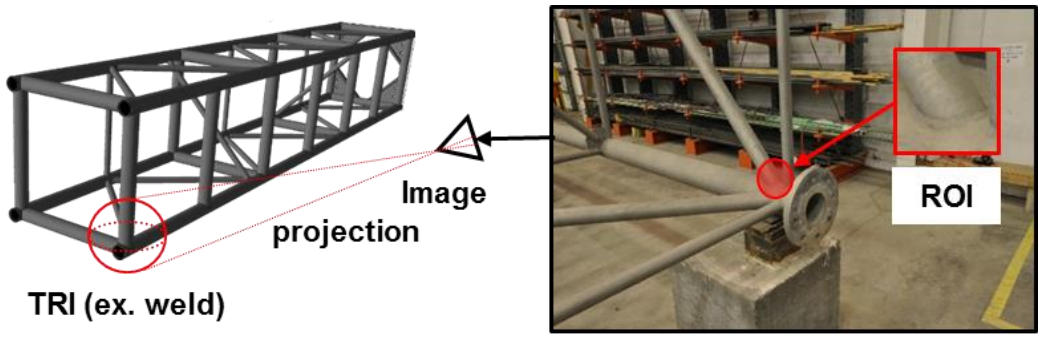


Figure 3.3 Projection of a virtual sphere to an image.

The size and location of the bounding box can be computed based on knowledge of the projection matrices and the TRI locations in the TRI coordinate system (hereafter, the TRI location indicates the 3D center location and radius of its virtual sphere). The bounding box is the smallest rectangle containing the ROI of the virtual sphere of the corresponding TRI on the images. The detailed derivation of the bounding box size and location is provided below.

In this study, a sphere projection model is introduced to compute the size of the bounding box that tightly encompasses each of the ROIs on each of the image (Guan et al., 2015; Sun et al., 2016). This model is originally developed for calibrating cameras when a spherical fiducial target is used for correspondence between images. Here the goal is to compute a projected area on an image corresponding to a virtual sphere defined in the TRI coordinate system, when all camera model parameters including interior and exterior parameters and lens distortion parameters are preliminarily obtained using SfM.

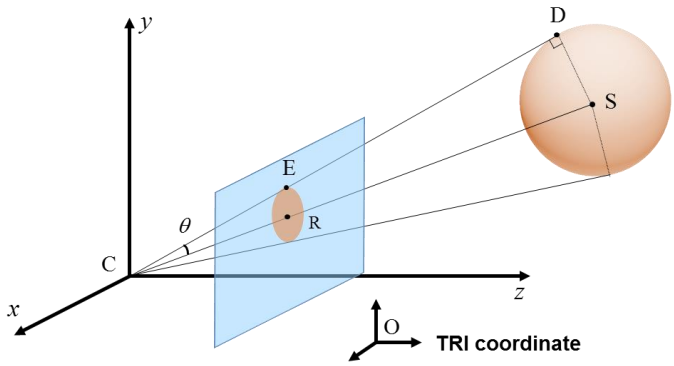


Figure 3.4 Geometric relationship between a sphere in the TRI coordinate system and a projection area in the image coordinate system.

Fig. 3.4 describes the sphere projection model. Suppose that the virtual sphere, assigned as a TRI is located at the center S, having a radius of the length of the line segment,

$\overline{\mathrm{SD}}$. All points in Fig. 3.4 are defined in the TRI coordinate system. The projection matrix P, obtained using SfM, is decomposed into a 3 x 3 camera calibration matrix (K), a 3 x 3 rotation matrix (R), a 3 x 3 identity matrix (I), and a 3 x 1 vector of the camera center, (C), shown in Eq. (3.3). The tilde on C indicates an inhomogeneous 3-vector representation. Each of matrices (or vectors) are obtained from the output of SfM, but if the software only provides the projection matrix, a singular value decomposition (SVD) can be similarly used (Hartley and Zisserman, 2004). In Eq. (3.3), the camera calibration matrix consists of four parameters, where α_x and α_y represent the focal length of the camera, and (x_0 and y_0) are a principal point in terms of pixel dimensions in the x and y directions, respectively.

$$P = KR[I \mid -\tilde{C}]$$

where

$$K = \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

The projection of the sphere on the image generates an ellipse at the center R, which is the projection of S on the image. An arbitrary point, E, located on the boundary of the ellipse, is a projection of D. D is the point where a (tangential) line from the camera center and the sphere meet. Thus, Point E satisfies Eq. (3.4), as illustrated in (Sun et al., 2016). Eq. (3.4) is derived using the fact that $\angle\mathrm{DCS}$ (θ) is constant with respect to Point D. In Eq. (3.4), x and y correspond to the point E on the image plane in pixels and [α, β, γ] is a unit direction vector of $\overrightarrow{\mathrm{CS}}$ ($\overrightarrow{\mathrm{RS}}$), which means $\alpha^2 + \beta^2 + \gamma^2 = 1$.

$$\frac{\alpha}{\gamma}\left(\frac{x - x_0}{f_x}\right) + \frac{\beta}{\gamma}\left(\frac{y - y_0}{f_y}\right) - \frac{\cos\theta}{\gamma}\sqrt{\left(\frac{x - x_0}{f_x}\right)^2 + \left(\frac{y - y_0}{f_y}\right)^2} + 1 + 1 = 0 \tag{3.4}$$

To compute the smallest rectangle that encompasses the ellipse, generated from Eq. (3.4), Eq. (3.4) is first simplified as Eq. (3.5) by introducing new variables:

$$A\left(\frac{x - x_0}{f_x}\right)^2 + B\left(\frac{x - x_0}{f_x}\right)\left(\frac{y - y_0}{f_y}\right) + C\left(\frac{y - y_0}{f_y}\right)^2 + D\left(\frac{x - x_0}{f_x}\right) + E\left(\frac{y - y_0}{f_y}\right) + F = 0 \tag{3.5}$$

where

$\lambda = \alpha / \gamma$, $\mu = \beta / \gamma$, $\sigma = \cos\theta / \gamma$
$A = \lambda^2 - \sigma^2$, $B = 2\lambda\mu, C = \mu^2 - \sigma^2$
$D = 2\lambda$, $E = 2\mu$, $F = 1 - \sigma^2$

The bounding box enclosing the ellipse consists of four lines each with a slope of either 0 or ∞. With this relation, Eq. (3.5) is represented as a single variable. For example, the derivative of Eq. (3.5) with respect to x becomes:

$$\frac{dy}{dx} = \frac{2Af_y(x - x_0) + Bf_x(y - y_0) + D}{-2Cf_x(y - y_0) - Bf_y(x - x_0) - E} \tag{3.6}$$

When the numerator of Eq. (3.6) becomes 0 (horizontal tangent line), the solutions of Eq. (3.6) define the two points at which the top and bottom horizontal lines of the bounding box meet the ellipse. Thus, the difference between their y values become the height of the bounding box, as shown in Eq. (3.7.2). Similarly, when the denominator of Eq. (3.6) becomes ∞, the solution of Eq. (3.6) defines the two points at which the left and right vertical lines of the bounding box meet the ellipse. The width of the bounding box is computed using Eq. (3.7.1).

$$BB_x = \frac{2f_x\sqrt{(2C^2D - CBE)^2 - (-B^2C + 4AC^2)(-CE^2 + 4C^2F)}}{-B^2C + 4AC^2} \tag{3.7.1}$$

$$BB_y = \frac{2f_y\sqrt{(2A^2E - ABD)^2 - (-AB^2 + 4A^2C)(-AD^2 + 4A^2F)}}{-AB^2 + 4A^2C} \tag{3.7.2}$$

Finally, (x_{ij}^x, x_{ij}^y) and (BB_{ij}^x, BB_{ij}^y) are obtained, which are the 2D center location and the size of the bounding box on the images, respectively, where *i* indicates the image index and *j* shows the index of the TRI. The superscript illustrates the x and y coordinates on the image.

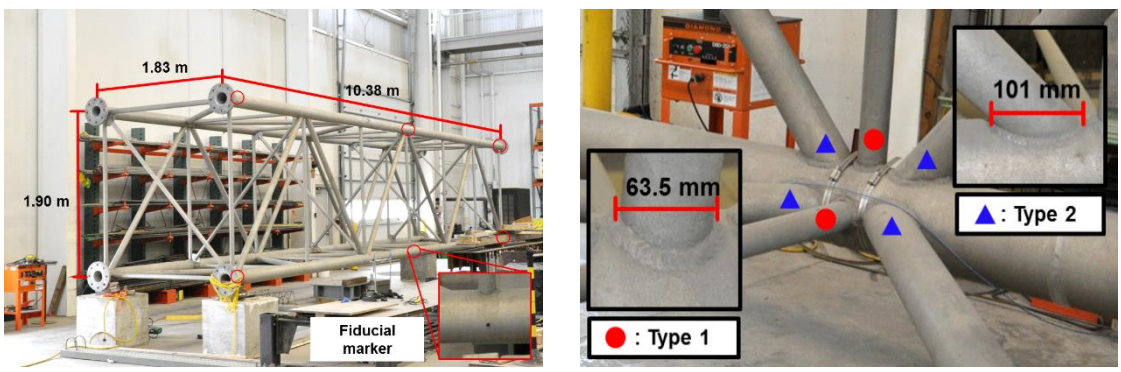
However, not all acquired images include TRIs, and they are not useful for visual inspection unless they are visible. The TRIs must be adequately visible on the images. Thus, the following two constraints should be satisfied. **Constraint 1:** $\min(BB_{ij}^x, BB_{ij}^y)$ should be larger than TP_j. Low resolution ROIs are not useful for visual inspection and are likely to produce unwanted false detection. Thus, the minimum side of the bounding box should be larger than TP_j, which is the minimum resolution of the ROIs, required for visual inspection in Eq. (3.1). **Constraint 2:** Each bounding box should be entirely visible on the

image. In another words, the following relation should hold $0 < x_{ij}^{x} \pm 0.5 \cdot BB_{ij}^{x} < w_i$ and $0 < x_{ij}^{y} \pm 0.5 \cdot BB_{ij}^{y} < h_i$ where w_i and h_i are the width and height of the image i. Bounding boxes are selected in each image when they satisfy the above two constraints.

3.3 Experimental Validation

A full-scale highway sign structure, shown in Fig. 3.5(a), is used for validating RILVI. This structure was originally built for supporting highway message signs and was previously installed for use on the highway. Currently, it is located inside the Bowen large-scale structural laboratory facility at Purdue University (Bowen Laboratory, 2016).

3.3.1 Description of experiment



(a) (b)

Figure 3.5 Description of a full-scale highway sign structure: (a) dimensions of the structure and layout of the circular fiducial markers for 3D coordinate transformation, and (b) two different sizes of the welds defined as TRIs.

This segment of the full-scale high way sign structure is composed of six cubic segments. It has four main chords, twenty-eight vertical braces, twenty-four diagonal braces, and seven internal diagonal-to-main chord braces. All braces 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. All braces are connected to the main chords by welds, for a total of 118 welded connections. These welded connections are assigned as the TRIs for purposes of validation of RILVI. The structure generally has a clean surface condition with no unusual corruption or damage to its components.

The fillet weld connections between the braces and main chords are selected as our target TRIs. As shown in Fig. 3.5(b), since cylindrical braces and chords are attached, the shape of the weld is a warped circular line. The specific shape and size of each weld line depend on the diameter of the brace and angle from the main chords. All weld connections can be divided into two types in Fig. 3.5(b). The two types have different center location values, but their radius is either 31.75 mm (Type 1) or 50.5 mm (Type 2). The internal diagonal braces, to be more precise, are attached to a junction plate and the shape of their welded connection is a straight line. However, its size is identical to Type 1, the diameter of the brace. Type 2 connections are larger than the radius of the diagonal brace due to the angled attachment. Here, the center location of each TRI is obtained from a 3D drawing of the structure. Thus, the TRI coordinate system is the same as the coordinate system in the drawing (model). For an actual implementation in which there is greater uncertainty in the as-built dimensions, there may be some sources of errors such as in the estimation of the projection matrices or a possible dimensional discrepancy between the drawing and the physical structure. Thus, the exact dimensions are multiplied by a factor (2) to conservatively set the radius of the virtual sphere so as to fully enclose the TRI in the bounding box.

In this study, VisualSfM is used to compute the projection matrices. VisualSfM is a non-commercial free software having a user-friendly graphic user interface (GUI) (Wu, 2013). This software has been widely used for many applications due to its accuracy and speed. Other than the SfM process, the remainder of the process is implemented using MATLAB. Twelve fiducial markers are manually marked on the main chords (six dots on the front and six on the back in Fig. 3.5(a)). Each circular dot has a 1 cm diameter, which is chosen to be visible on the collected images. Each of their 3D locations in the TRI coordinate system are known in advance using the 3D drawing, and the corresponding dots are found across the images to define 3D locations in the SfM coordinate system. As mentioned, even this process can be readily automated if coded fiducial markers are used. Once fiducial markers are detected on images, a nonlinear triangulation method is used to estimate the 3D location in the SfM coordinate system (Hartley and Zisserman, 2004).

3.3.2 Images collection from the test structure

For validation of the method, images of the structure are collected using a Nikon D90 camera with 18–105 mm lens. No zoom or flash functions are used, and the location of the camera is not pre-determined for each photo. A UAV is not flown for this validation, although parameters related to image acquisition are designed to be close to those needed for practical use of this technique using UAVs. A total of 789 images are collected, all of which are used for localizing ROIs. The camera setup has a fixed focal length of 18 mm, a resolution of 4,288 x 2,848 pixels, and the aperture is f/10 and the shutter speed is fixed at 1/30sec. The effect of motion blur is not considered in this experiment. Based on Eq. (3.1), a distance threshold, *WD*, is computed using the following parameters: *FL* = 18 mm (camera setup), *SR* = 4288 pixel, *SS* = 23.6 mm (width), *TS* = 63.5 x 2 (factor) mm and *TP* = 127 pixel. This assumes that the accuracy for visual inspection may be more than 1px/1mm. Note that *TS* varies depending on the task to be performed. α is selected to have a uniform distribution with limits of 0 and $\pi/3$ rad. The minimum angle of β is 0.92 rad, which is computed using *SS* and *FL*. β is also selected to have a uniform distribution with limits of 0.92 and $\pi/2$ rad. *WD* is determined based on the average value obtained from 1000 simulations of Eq. (3.1) by randomly selecting combinations of α and β with the above parameters. *WD* is approximately 2,200 mm which is the value used for image collection. Note that this process is conducted before collecting the images, and the *WD* provides a rough guideline for planning a flight path for a UAV.

The images are sequentially taken from roughly 2~3 m away from the structure. It is important to note that these images are acquired from locations that are not pre-determined. The images are captured by circling around the structure six times collecting images from a similar height but, during each lap, the camera angles are changed. Six angle variations are configured including three angles in a vertical direction (up-middle-down) and two angles in a horizontal direction (right and left oblique). Fig. 3.6 shows sample images of the structure collected for the validation. The images in each row are captured from same vertical angle, which corresponds to up, middle, or down from top to bottom row. For successful estimation of the projection matrices using SfM, image collection is designed in such way that large overlaps between images are obtained. Several tactics are employed to achieve this overlap. First, more than seven images are collected at each

segment of the structure without changing angles in each lap. Images are collected using small transitional movements without changing angles to obtain large overlaps between images. Next, front images are not collected from a perpendicular orientation. The test structure has a clean and shiny surface, which may not produce enough feature points to be shared between images. This causes failures in estimating projection matrices using SfM. In the case of our target structure, more feature points are generated from the geometric features of the structure or backgrounds. Thus, oblique images are collected by varying horizontal angles to include more of those areas in each image.

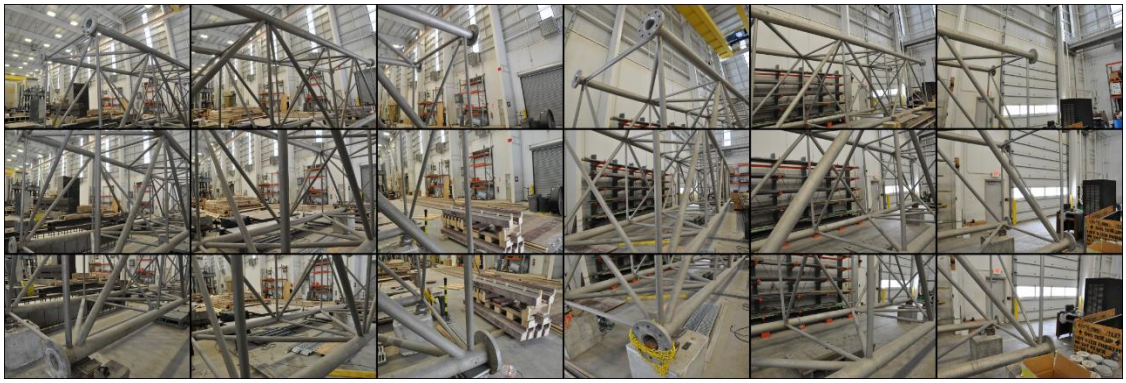


Figure 3.6 Samples images used for the demonstration.

3.3.3 Results of the ROI localization

Fig. 3.7 illustrates many localized ROIs on the images. Only the TRIs close to the camera location of the image (having sufficient ROI size) are selected as valid ROIs, which are marked as a red bounding box. *TP* in constraint 1 in chapter 3.2.4 varies depending on the radius of the TRI (a type of the welded connection). The accuracy for visual inspection (a ratio of *TP* to *TS*) is assumed to be 1pixel/1 mm and thus, 127 and 202 pixels are used as TP corresponding to a type 1 and 2, respectively. When the camera is closer to the TRIs or when the virtual sphere of the TRI has a larger radius, the size of the bounding box is bigger. The largest box in the first image in Fig.3.7 has a type 2 weld connection. This type of weld has a wide radius, and a closer distance to the camera. TRIs located in the background, are thus farther away from the camera, are not selected as ROIs on images.

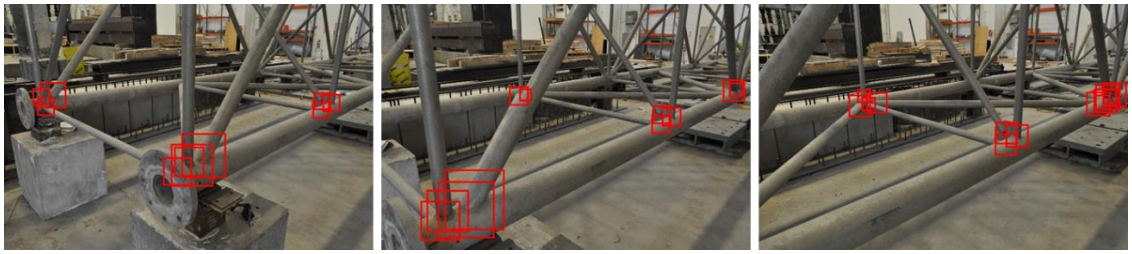


Figure 3.7 Variation of the bounding boxes (ROIs) on images.

The ROIs of all 118 TRIs are localized in each of the collected images. Examples of the ROIs obtained are shown in Fig. 3.8. The array of images on the left are the ROIs, for which each TRI is marked as a red box in the right image. Since the original images are collected from various angles and positions, the ROIs include various viewpoints of the TRI but their scale is almost identical. Ideally, the center of a given weld connection should always be located at the center of the ROI, and their size on the image should be identical. However, as mentioned in chapter 3.3.1, sources of error, for instance due to realistic construction uncertainties, can produce minor discrepancies in the ROI's scale (size of the TRI in the ROI) and center location. For example, there are slight erroneous mappings in the third TRI in Fig. 3.8. Only 30 ROIs for each TRI are listed in Fig. 3.8, but their actual numbers are 101, 62, 79, and 60, respectively. Using such a set of ROIs, a human inspector could readily inspect each of these welds.

The number of ROIs obtained depends on how many pictures include the corresponding TRIs. However, to achieve good quality for the ROIs, users also consider possible occlusion of the ROI. In the case of the TRI in the second and fourth row in Fig. 3.8, its view is impeded by the geometric features of the structure, and the ROIs are too occluded to be used for visual inspection. Thus, users should carefully design the camera angles and location variations to collect a sufficient number of images to yield good visibility of the TRI.

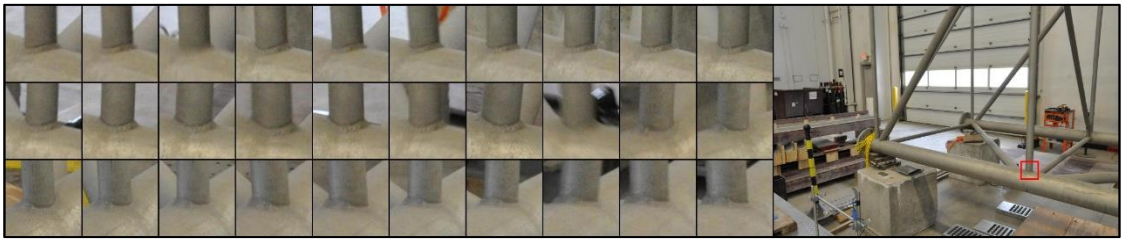


Figure 3.8 Examples of the ROIs: Tiled images on the left show first 30 ROIs of the identical TRI location, which are extracted from different images. A red box on the right image shows the corresponding TRI.

To further examine the feasibility of RILVI, experiments are conducted under different lighting conditions. This experiment is intended to prove that the developed method produces consistent localization outcomes even under light variations. A total of 836 images are thus collected with interior fluorescent lighting. All camera settings and the image collection strategy are identical to the previous experiment. Although these images have more noise than those captured in daylight, all collected images are successfully utilized in SfM for the estimation of the projection matrices.

Fig. 3.9 shows the ROIs extracted from this new image set. Their TRIs are identical to the first two ones in Fig. 3.8. This result shows that the TRIs on the ROIs are still clearly visible under lighting changes. Since the camera angles and locations are different for each image, the appearance of the TRIs on the ROIs are somewhat different from those in the previous image set. For example, a couple of the occluded ROIs are shown for the first TRI in Fig. 3.9 to demonstrate this effect.

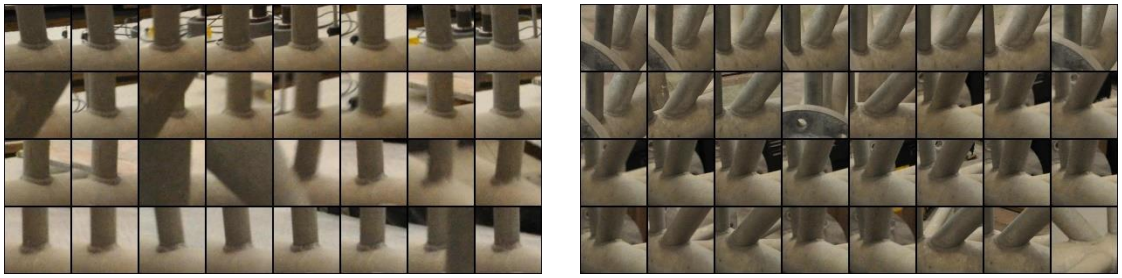


Figure 3.9 Localization of the ROIs from the images collected under lighting changes (The TRIs used here are the same as those shown in the first and second rows in Fig. 3.8.).

3.4 Conclusion

This study presents and validates an automated region of interest localization technique to support vision-based visual inspection, named RILVI. UAVs and other automated image capture systems can be used to collect a large volume of images for visual inspection. However, direct implementation of existing vision-based visual inspection methods on the raw images will generate large numbers of unwanted false-positive errors, and dramatically reduce the efficiency of the inspection process. The key contribution of the technique developed here is to extract appropriately cropped and scaled regions of interest from each of the collected images before applying either human-based or machine-based visual inspection techniques. The regions of interest correspond to the target regions if interest on the structure itself, and should be selected according to the inspection needs of the structure. The feasibility of RILVI is demonstrated using a full-scale highway sign structure. 118 welds, which are assigned as our target regions for visual inspection, are successfully localized using 789 original images. A virtual sphere is adopted here to enclose the target region of interest, although this choice can readily be modified as needed for other geometries. It is expected that RILVI will provide the means to apply existing automated vision-based visual inspection techniques on a large volume of UAV images for rapid and efficient visual inspection.

CHAPTER 4. SEMANTIC ANNOTATION OF EARTHQUAKE RECONNAISSANCE IMAGES

4.1 Background

Post-event reconnaissance teams have a critical mission: to collect scientific data to learn from disasters. Despite the large volumes of images that have been gathered from past earthquakes, only a small portion of these are accessible to the public and archived with certain basic information such as date, event, or location. Thus, the ability to access and facilitate reuse of these images based on the true semantic contents on images is limited. Currently, there is no established formal terminology and annotation method for describing visual contents in such images. This impedes the use of images for generating new knowledge, and large volumes of images remain largely unused for scientific research.

The value of visual data (e.g. image, or video) enhanced through accurate and useful annotation will empower researchers across several disciplines to distil the importance lessons that will enable engineers and researchers to improve the resilience of our communities against natural events. Determining appropriately structured and formalized descriptive information for these images will enable their scientific use and retrieval. For instance, longitudinal studies or regional studies comparing structural performance would lead to decisions regarding design practices. A structured set of descriptive information is essential for making use of these data. *Long-term preservation of the large volumes of images collected is only effective if it is discoverable by future researchers*. An example related to impact of reconnaissance data, a longitudinal study directly comparing the performance of school buildings in Turkey was conducted. Images collected during the 1999 Düzce and 2003 Bingöl earthquakes were used, and both time variations and regional techniques were examined. The conclusion was that, regardless of construction quality, the use of structural walls was the single important structural characteristic impacting life safety in these schools. The presence of structural walls drastically improved performance, and prevented collapse (Gur et al., 2009). This single example of data reuse is one of many that provide strong motivation for formalized annotation of such data to guide policy and code decisions.

The US National Science Foundation requires researchers to follow a data management policy, and other nations and sponsors are also requiring open access to data (Warren et al., 2008; Design safe-ci, 2016; EERI, 2016). Furthermore, research domain repositories often have policies governing the requirements for curating and publishing such data (Bosi et al., 2015). However, in spite of the enormous investment involved in the collection of visual data, it has not historically been as carefully documented or organized due to the time and resources needed to perform the documentation.

Recently, the earthquake engineering community has promoted data repositories, often integrated within science gateways, to gather various types of scientific data and lessons learned in the field. The Earthquake Engineering Research Institute (EERI) has archived a broad collection of earthquake reconnaissance data collected by multidisciplinary teams of researchers (e.g. earth scientists, engineers, social scientists) (eeri.org). Although general information can be retrieved about the event (e.g. data, event, seismic intensity and country) and other data resources are available (e.g. article, report or image), the actual image data collected in the field are categorized as “Other resources.” Structure researchers are not able to neither identify the types of data nor automatically search of their contents. Furthermore, images are not properly curated as well as available in use for only a few events. Another system, CEISMIC is designed to include a digital archive of multimedia contents related to the Canterbury earthquakes of 2010 and 2011 (ceismic.org.nz). Archived data can be filtered by several types (e.g. images, newspapers, or videos) and metadata (e.g. keyword, time, or provider). However, image search results are limited because images are retrieved based on associated text descriptions included in related documents, and not using specific tagged keywords. The Earthquake Engineering Online Archive and NISEE e-library is a database of literature, photographs, data and software in earthquake structural and geotechnical engineering (nisee.berkeley.edu). Various types of data from historical earthquakes are well documented and accessed by searching matching keywords. However, this is not data-intensive repository and provides very few information including images related with recent earthquake events (e.g. Haiti, 2010 or Nepal, 2014). Next, a complete well-curated image database is published through Disaster and Failure Data Repository in National Institute of Standards and Technology (NIST) (disasterhub.nist.gov). Images collected from Chile earthquake in 2010 and Joplin

Tornado in 2011. This database clearly defines 23 keywords for tagging photographs, to aid search for photographs in database. Lastly, "datacenterhub.org" is an ongoing project funded by Purdue University and the National Science Foundation (NSF-1443027), and provides comprehensive collections of earthquake reconnaissance data and images (datacenterhub.org). Data are curated in the form of a table to enable readily comparisons and searching. They are ongoing development of keyword-based image searching.

As our image collections from reconnaissance missions continue to grow, a formalized and structured method is needed to store and retrieve earthquake reconnaissance images (hereafter, earthquake images) along with *descriptive metadata* of these images. The descriptions should be based on the visual contents as extracted from the images collected, and integrated with the necessary information about the location and source of the image. However, currently, there is no schema or methodology to be used for representing visual semantic contents and engineer's explicit interpretation and knowledge in earthquake images. Unorganized and heterogeneous keywords or descriptions of semantic contents on images disrupt retrieving right contents and future reuse for scientific research.

Herein, a high-level annotation schema is proposed for describing the visual semantic contents of earthquake images in a structured way. The method is developed to incorporate original descriptions of earthquake images by preserving the original meaning of the description. Annotated contents and images can thus be fully retrieved using a semantic query. A core idea is to annotate images using formalized terms and relations constructed by Earthquake Image Ontology (EIO). With support of the associated annotation tool, human annotators can easily select proper terms and their relations for descriptions and convert them in structured forms. Ultimately, this study aims to initiate a discussion of the visual semantic annotation in earthquake images. The proposed method is intended to provide a significant step forward in handling these unstructured images in such a way as to be manageable and tractable.

4.2 Problem Statement

The term "annotation" as used in this chapter is defined first. *Image annotation* is defined as the practice of capturing and collecting the contents associated with image data. This

often indicates automated annotation through either natural language process or computer vision algorithms (Krizhevsky et al., 2012; Pustejovsky and Stubbs, 2013; Johnson et al. 2015), but, in this study, the scope of work is limited to a manual annotation, or "labeling," by human annotators. The database constructed by robust manual annotation methods must take precedence in order to perform the annotation in more advance ways.

In general, the contents are divided into two major categories: (1) properties of the image itself, such as size, resolution, location or date, and (2) actual visual content such as properties of the object, person or abstract concept (e.g. room, wall, failure?) depicted by the image (W3C, 2007). Simply, the former category answers "how, when and why was the image made or what information should be known to understand (the setting, or, to place) the content of the image?" and the latter category provides an answer to "what does the image depict or illustrate?". The terminology for the description and documentation necessary to address the first category has been well established and standardized, for example, Exchangeable Image File Format (EXIF) or Dublin Core (W3C, 2007; Dublin Core, 2016). Most of current image repositories were built based on extension of such basic metadata. However, the second category, "what should be known to understand the visual content of the image", called semantic annotations, is still a complex problem. Descriptions of images have large variations and level of details in domain terms and expressions for knowledge representations, and its dependence on the associated application and background knowledge of annotators. Storing those information as fully searchable forms is much more challenges. In general, a keyword based image tagging and retrieval by matching text keywords (or visual features using computer vision methods) is widely used in existing database and commercial service (e.g. Flickr, google). However, because users often generate the description and metadata using their own terminology without any structure, the ability to retrieve useful data is limited without consistently annotated contents and contextual information. Also, keywords based description suffer from lack of ability to relate adjectives to nouns (example in next paragraph). Lastly, tagging the keywords are not sufficient to describe earthquake images to be used for scientific and knowledge generation purposes.

Figure 4.1 Sample image of reinforced concrete shear wall damage in the 2010 Chile earthquake (Moehle et al., 2011; Telleen et al., 2012).

To better understand the intent of this research, consider the sample earthquake image in Fig. 4.1. This image was collected by an earthquake reconnaissance team after the 2010 Chile earthquake. This image opened a critical line of inquiry that strongly impact on the practice of structural wall design because the damage mode, called "overall wall buckling", had rarely been observed in past earthquakes (Moehle et al., 2011; Telleen et al., 2012). The following statement was provided as the original description of the image: "Reinforced concrete shear wall has longitudinal crushing, spalling at height of the wall, and buckling of vertical reinforcement at the boundary" (Moehle et al., 2011). This is a typical description, and includes information about the visual contents such as the structure component type, several damage types and directions, and relative location of the damage. The outcome for this research is to store information in the data so that retrieve all information by query. Keyword-based image tagging would not effectively handle so much information. For example, if reduced to keywords, "longitudinal" and "vertical" can refer to either the reinforcement or the damage, and the locations of "height" and "boundary" lost their relationship to targets in the image. In additions, it is difficult to represent spatial information using simple keyword tagging.

4.3 Methodology

Before providing details on how to achieve this goal for earthquake images, the technology needed to support the proposed annotation method is introduced. Resource Description Framework (RDF) is a formal language for describing structured information (Bönström et al., 2003; W3C, 2007). The RDF data model defines data as a 3–tuple, referred to as

triples (W3C, 2007). For example, suppose that the information to be stored is “John has a son, Brad.” This data is represented as a triple using “John–hasSon–Brad”. The RDF data model is mainly used as a plain format to store or exchange information, without providing any semantics. In other words, this data model explains how to represent the information as a triple but does not provide the mechanism to define names for the specific property or value. In the previous example, without knowing that “John” can have the property of “hasSon”, their relationship cannot be hypothesized. Thus, a schema to define resource types and names, and their structure (or relationships) is needed. Web Ontology Language (OWL) is an ontology language that can be used to represent the meaning of terms and their relationships for domain applications in a more expressive way. In this study, the OWL-DL language is employed for modeling our EIO (Staab and Studer, 2009). Here, DL stands for description logics, which is designed for reasoning systems.

The main OWL elements are *classes* and *properties*. A *class* is a set of terms that are used for describing concepts in a particular domain. A *property* expresses a relationship between two classes or with a description of an object using values. Actual data interested in this study are stored in instances in classes. In the previous example, “Father” and “Son” are possible classes, and these classes have *instances* of “John” and “Brad”, respectively. Their relationship is defined through the property “hasSon”. In our EIO, a class includes the terms that are needed for describing the visual content of the images, and a property is the relationship between classes. Here, the core idea underlying the proposed annotation method is that sentence-like information, such as “John has a son, Brad”, can readily be converted into a triple, John–hasSon–Brad. To use this approach, the classes and properties to be used would be selected from the developed ontology.

Based on a detailed review of current practices used to describe the earthquake images in the literature, the original descriptions used for earthquake images can be nearly automatically represented by appropriate set of multiple triples while maintaining much of the richness of expression in language. Each highly detailed description may be transformed and stored in the RDF model guided by the predefined EIO. Having done that, the features of the RDF model are exploited such as storing linked data and semantic query searching. For example, the long image description in chapter 4.2 might be stored as

multiple triples in the database. This would enable a researcher to retrieve the stored information using a simple semantic query, as illustrated in chapter 4.6.

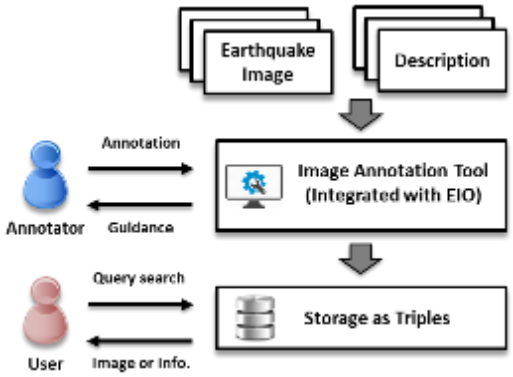


Figure 4.2 Overview of the proposed image annotation method.

Fig. 4.2 shows the general architecture of the proposed annotation method. The overall process is that annotators exploit an image annotation tool to store the original descriptions of images. The tool would employ the pre-defined classes and their properties in our EIO, assisting annotators to apply valid class names and types, while converting the original descriptions to multiple triples. With these in place, the stored information can be fully retrieved using query languages. Here, Protégé is used for developing our ontology suitable for annotating earthquake images. Protégé is ontology design software and has a user–friendly interface and support for exporting ontology definitions in the established RDF/OWL database format to compose and execute queries using RDF query languages (protege.stanford.edu).

All terminology used in the remainder of this paper are those used in Protégé. Vocabularies (objects or terminologies) are referred to as a *Class* and their relations (object property) and relation to value (data property) are described using a *Property* (Noy and McGuinness, 2001; Horridge et al., 2004). There is no mandatory naming convention for OWL-DL classes and properties, but in EIO all class names begin with a capital letter with no space between words, and property names start with ‘has’ or ‘is’ and have no spaces and use capitalization for the remaining words. This choice of convention helps clarify the intent of the property to annotators (Horridge et al., 2004). Hereafter, all class and property names are written using *italics*.

4.4 Earthquake Image Ontology

The development of EIO is intended for annotating earthquake image using a broad, but still focused, range of standardized terminology and structures. EIO is highly tailored to provide rich and natural descriptions of visual semantic content in earthquake images. EIO has been designed based on image descriptions in published articles, reconnaissance reports, and manuals related with earthquake building reconnaissance (Baggio et al., 2007; FEMA, 2015b; ATC, 2016). It is meant to be quite flexible, permitting changes and expansion of classes and properties over time. All classes and properties are designed using English, and the use of all morphological variants of a word, such as plurals of nouns or inflected forms of verbs (e.g., collapse, collapsing, collapsed) are ignored. Thus they all point to one single class defined in EIO.

EIO has two top classes: *Visual* and *Metadata*. This division is based on whether or not the information can be collected from visual semantic content. As mentioned in chapter 4.2, this study focuses on the visual semantic content included in *Visual*. Note that "semantic" means other non-semantic visual contents are considered here such as color, shape or pattern across the images unless they contain a particular semantic meaning.

In *Visual*, there are three subclasses. Together these cover most of the classes appearing in existing image descriptions. The three subclasses are: *Target*, *Feature* and *Damage*. Figs. 4.3(a) to (c) provide a list of the subclasses in Protégé and these subclasses inherit the characteristics of the superclasses. *Target* refers to the subject of an image. In Target, there are two broad classes, *Object* and *Place*. *Object* is a thing that has a visually clear boundary and illustratable shapes, such as *Column*, *Wall*, or *Chimney*. *Place* as the name suggests, is a space having a particular purpose, such as *Balcony*, *LivingRoom*, or *Basement*. This is typically inferred from the existence of objects and their spatial configuration. *Feature* includes any characteristics of a *Target* such as material, shape, or direction. *Feature* always modifies *Target* and cannot modify other classes in *Feature*.

Damage, a special class, is a frequently used class to describe the damage state of *Target*. However, classes in *Damage* can be used as either *Target* or *Feature* in the context of the original description. In other words, Damage is subclass of either *Target* or *Feature* according to the way of its usage. For example, "*Vertical Cracking*" indicates that *Vertical* in *Feature–Direction* describes a direction of *Cracking* in *Target–Damage*. On the other

hand, "*Collapsing Wall*", *Collapsing* is used as a characteristic (*Feature*) of *Wall* in *Target–Object*. Classes having the same semantic meanings may be used in the earthquake image description, and are registered as an equivalent class such as *Rebar* ≡ *Reinforcement*, or *Floor* ≡ *Story*. In addition, a class that requires clarification may include subclasses. For example, *Failure* can be modeled as a superclass of a kinds of severe damage such as collapsing or leaning, explained in chapter 4.6.

Object properties are defined based on the relationship between two classes, shown in Fig. 4.3(d). The relationship between classes is not unique and various object properties can be defined (but, actually, there are not many). For the above example, *hasDirections* is an appropriate object property linking *Cracking* and *Vertical*. Object properties also have hierarchical subproperties, for instance, some subproperties of *isLocatedAt* are *isLocatedOn*, *isLocatedNext*, and *isLocatedUnder*.

(a) Target (b) Feature (c) Damage (d) Object property

Figure 4.3 *Target*, *Feature*, *Damage* and object properties in Earthquake Image Ontology (EIO).

4.5 Image Annotation Tool

The image annotation tool is designed to assist annotators to write structured image descriptions. The tool is intended to support them by providing a normative selection of appropriate names of classes and properties. Also, based on EIO, the tool can perform several functions such as identifying synonyms (e.g., *Rebar* and *reinforcement*), autocompletion (e.g., a user types a few characters of a term and the tool suggests a proper class or property), or clarifying ambiguity in words used to specify the meaning (e.g., *Failure* in chapter 4.6). Once a proper description is written, the tool generates multiple

triples in a semi-automated manner (Schreiber et al., 2001; Hollink et al., 2003; Im and Park, 2014). Here, "semi-automated" indicates that annotators are able to select appropriate class names or properties from a list supplied by the tool.

As a result of our comprehensive review of earthquake image descriptions used in the literature, a major annotation pattern was identified in the annotations of earthquake images. The pattern consists of an illustration of a main target object by its characteristics, conditions, or spatial location relative to other objects, for example, "collapsing masonry wall", "circular column" or "spalling column" (for further examples see Fig. 4.4). Thus, the following template is proposed, which is best suited for rich annotation using such descriptions:

<u>Feature 1</u> <u>Target 1</u> (Object property) <u>Feature 2</u> <u>Target 2</u>

where each underlined slot is a single class field, and each slot in the parenthesis is a single object property.

Feature 1 and Feature 2 would contain a class from *Feature* or *Damage*, and Target 1 and Target 2 would contain a subclass from *Target* or *Damage*. These are denoted as F1, T1, F2, and T2, respectively, in the sequence. A description using this template is referred to as a statement, and considered to represent an English sentence. Roughly speaking, (F1 and T1), (Object property), and (F2 and T2) represent the subject, verb, and object or adjective, respectively). All fields are not necessarily required in such a statement. However, in each statement, annotators have to enter T1, which is the subject of the statement. A class in T1 (or T2) is automatically stored as *Image–has*{*}–T1 (or T2). Here, *Image* is a class that stores the annotation information. "–" represents a delimiter for tuples in a triple, and "{*}" is the top subclass name of the corresponding class in *Target*, *Feature* or *Damage*. When T1 is in *Target*, {*} can be either *hasObject*, *hasDamage* or *hasPlace* for the object property. Then, annotators can enter other fields in the template for constructing a given statement. F1 always describes the characteristics of T1, and the "*has*{*}" property between them is automatically assigned. Thus, a triple of T1–*has*{*}–F1 is generated. For instance, the statement of "F1: *Collapsing*, T1: *Wall*" is converted as *Image–hasObject–Wall* and *Wall–hasDamage–Collapsing*. Damage is the superclass of *Collapsing*. If both F2 and T2 are entered, a triple is generated between F2 and T2 in a similar fashion as (T2–has{*}–F2) and (Object property) in the template is selected based

on the object properties between T1 and T2. The annotation tool suggests a list of possible object properties, and the annotator selects an appropriate property that can best describe the meaning of the original description. If the annotator only provides an entry for F2, the property is selected according to the relationship between T1 and F2. For example, the statement "F1: *Concrete*, T1: *Wall*, F2: *Collapsing*" is converted as *Image–hasObject–Wall*, *Wall–hasMaterial–Concrete* and *Wall–hasDamage–Collapsing*.

The actual usage of this template for image annotation is straightforward. For example, annotation of the original long description in chapter 4.2 is demonstrated in Table 4.1. The original description can be represented as 6 statements, and 15 triples are generated, allowing for search and ready retrieval in the future. Note that the integration of multiple triples results in a description that includes almost the same information and has the same meaning as the original description.

Table 4.1 Annotation example using the description "Reinforced concrete shear wall has longitudinal crushing, spalling at height of the wall, and buckling of vertical reinforcement at the boundary"

Statements	**Triples**
F1: *ReinforcedConcrete*, T1: *ShearWall*, F2: *Longitudinal*, T2: *Crushing*	*Image – hasObject – ShearWall* *Image – hasDamage – Crushing* *ShearWall – hasDamage – Crushing* *ShearWall – hasMaterial – ReinforcedConcrete* *Crushing – hasDirection – Longitudinal*
T1: *ShearWall*, F2: *Longitudinal*, T2: *Spalling*	*Image – hasDamage – Spalling* *ShearWall – hasDamage – Spalling* *Spalling – hasDirection – Longitudinal*
T1: *Spalling*, T2: *ShearWall*	*Spalling – isLocatedInTop – ShearWall*
T1: *Crushing*, T2: *ShearWall*	*Crushing – isLocatedInTop – ShearWall*
F1: *Vertical*, T1: *Reinforcement*, T2: *Buckling*	*Image – hasDamage – Buckling* *Image – hasObject – Reinforcement* *Reinforcement – hasDamage – Buckling* *Reinforcement – hasDirection – Vertical*
T1: *Reinforcement*, T2: *ShearWall*	*Reinforcement – isLocatedInSide – ShearWall*

* Note that overlap triples that are generated in the previous statement are removed.

More examples are shown in Fig. 4.4. All images or original descriptions in this figure have been extracted from published sources including data repositories, reports or

articles, which document image data in previous earthquake events. On the whole, regardless of an inevitable unnaturalness coming from the use of template annotation and triple conversion, the original meanings of descriptions can be preserved in the triples.

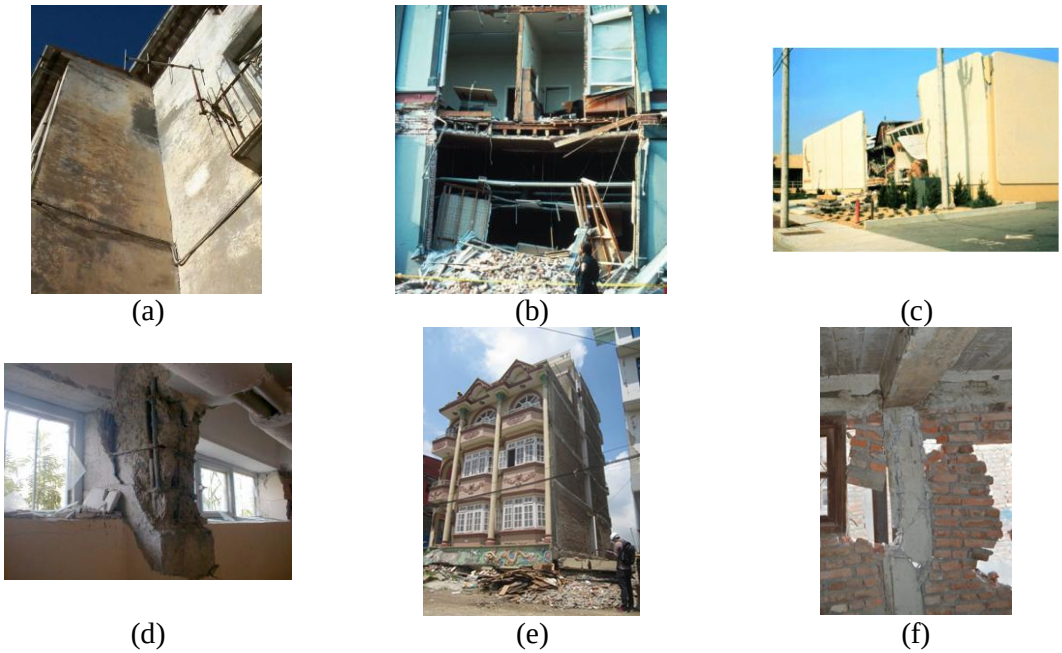


Figure 4.4 Example annotations of real-world earthquake images:

(a) Description: Vertical cracks along the Orthogonal Wall (Italy, 1998) (Baggio et al., 2007)
Statements: (F1: *Orthogonal*, T1: *Wall*, F2: *Vertical*, T2: *Cracking*)
Triples: (*Wall – hasDamage – Cracking*), (*Wall – hasShape – Orthogonal*) and (*Cracking – hasDirection – Vertical*)

(b) Description: Failure of an unreinforced masonry wall in a building (USA, 1989) (FEMA, 2015a)
Statements: (F1: *UnreinforcedMasonry*, T1: *Wall*, F2: *Failure*) and (T1: *Wall*, T2: *Building*)
Triples: (*Wall – hasMaterial – UnreinforcedMasonry*), (*Wall – hasDamage – Failure*) and (*Wall – isLocatedAt – Building*)

(c) Description: Collapse of a tilt–up bearing wall (1994, Northridge earthquake) (FEMA, 2015a)
Statements: (F1: *Collapsing*, T1: *TiltWall*)
Triples: *TiltWall – hasDamage – Collapsing*

(d) Description: Failed captive column in the basement (1999, Turkey earthquake) (Gur et al., 2009)
Statements: (F1: *Failure*, T1: *CaptiveColumn, T2: Basement*)
Triples: (*CaptiveColumn – hasDamage – Failure*) and (*CaptiveColumn – isLocatesdAt – Basement*)

(e) Description: Soft story failure (2015, Nepal earthquake) (Shah et al., 2015)
Statements: (F1: *SoftStory*, T1*: Building*)
Triples: *Building – hasDamage – SoftStory*

(f) Description: Shear failure reinforced concrete column next to collapsed masonry wall (2015, Nepal earthquake) (Shah et al., 2015)

Statements: (F1: *ShearFailure,* T1: *Column,* F2*: Collapsing,* T2*: Wall*) and (F1: *ReinforcedConcrete,* T1: *Column,* F2*: Masonry,* T2*: Wall*)https://datacenterhub.org/resources/14160
Triples: (*Column – isLocatedNext – Wall*), (*Column – hasDamage – ShearFailure*), (*Wall – hasDamage – Collapsing*), (*Column – hasMaterial – ReinforcedConcrete*) and (*Wall – hasMaterial – Masonry*)

* Parentheses in statements and triples are used to separate entries for clarity.
** Triples of *Image-has*{*}-T1 (or T2) in Triples are omitted.

Although our triple representation uses the names of classes in EIO, the actual annotated data are stored as individuals (instance in OWL) in the corresponding classes. For example, suppose that the description of a specific image is "collapsing wall". An individual, Image1, in *Image* is created and all annotation data related to the corresponding image are stored (hereafter, an underline for the name of each individual is used). Individuals Collapsing1 and Wall1 are also generated. Then, the actual triple statements to be stored become Images1–*hasObject*–Wall1 and Wall1–*hasDamage*–Collapsing1. This statement is interpreted as "wall" in the description of a specific wall in *Wall* in the corresponding image, and this wall is named as Wall1.

4.6 Evaluation of the Proposed Approach for Image Retrieval

The RDF data model and query searching languages are well established and already in use in many application domains (W3C, 2008). Thus, the readers are recommended to review how to use RDF query languages for data retrieval and to provide the capability of query searching for data already written in triples. Rather, our focus in this subchapter is to demonstrate how closely query searching results can yield ground-truth original descriptions, and through that process the effectiveness of the proposed annotation method is validated. Protégé provides a powerful query searching utility called DL–query. The query language (class expression) supported by the plugin can compose and execute queries with the Manchester OWL syntax, a user–friendly syntax for OWL-DL used in Protégé. However, as long as identical triple information produced from original image description is stored, query search results are almost identical regardless of the query languages or platforms.

For demonstration purposes, the images are annotated shown in Fig. 4.4 based on their descriptions. The annotation data for image Figs. 4.4(a) to (f) store the individuals

Image1~6, respectively. The annotated classes in each image produce individuals having the same naming convention, for example having number as suffix. The annotation example of Fig. 4.4(a) is presented in Fig. 4.5. the data are visualized using *OntoGraf* in Protégé. All individuals, their classes, and object properties between classes ("Arc Types" son the right of Fig. 4.5) that are used for annotation of Fig. 4.4(a) are visualized.

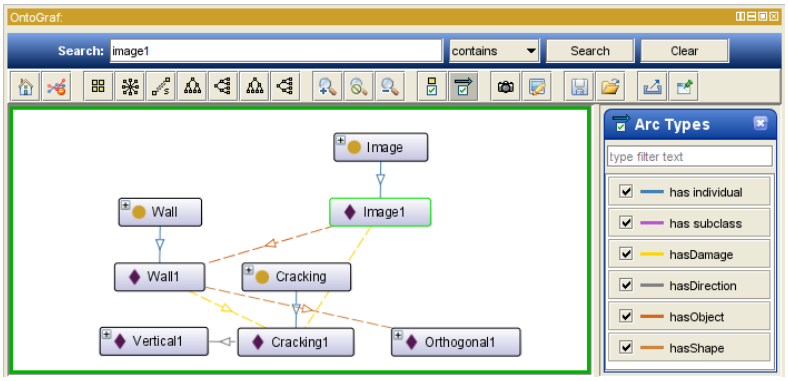


Figure 4.5 Visualizing the annotation data of Fig. 4.4(a) using the *OntoGraf*.

Some sample queries that can be used for retrieving annotated data are examined below. Note that when individuals in DL-query like Fig. 4.6 are checked, a query is used for searching an individual in a corresponding class, which stores actual annotated data.

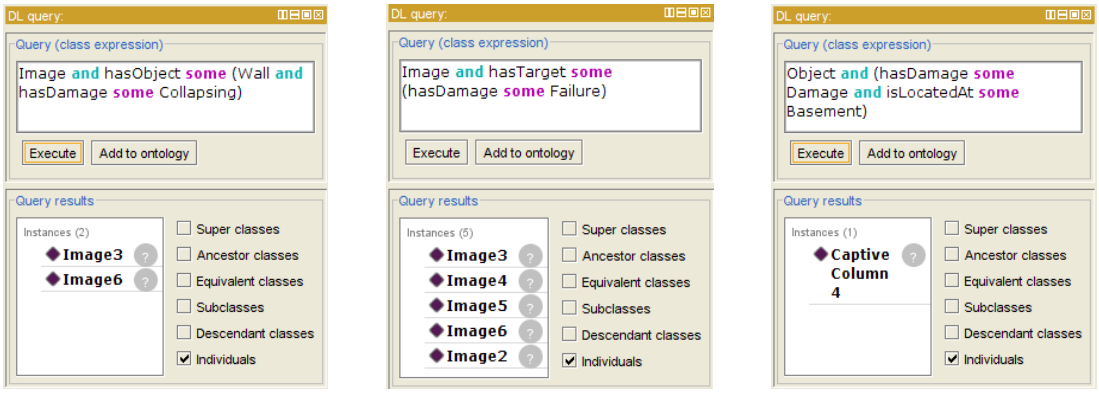
Query 1. Which image has a collapsed wall? (Fig. 4.6(a)): This is a relatively simple query. All classes *Image* having an object *Wall* with damage *Collapsing* are found. However, strictly speaking, individuals in *Image* that include a specific wall having collapsing damage are found here. The query expression is "*Image* and *hasObject* some (*Wall* and *hasDamage* some *Collapsing*)". The query result is Image3 and Image6. Interestingly, Image2 is not captured with this query because of its damage types as *Failure*. *Failure* is ambiguous and has no specific definition regarding image annotation. However, typically, as saying "failure of the components", it indicates that the components have severe damage and are not serving their function. Thus, in EIO, *Failure* is classified as a superclass of *Buckling*, *Collapsing*, *Crushing*, *Drifting*, *Dislocating*, *Learning*, *StoftStory*, *ShearFailure*, and *SevereSpalling*. Here, *SevereSpalling* is defined as "*Spalling* and (*hasDamageLevel* some (*Large* or *Severe*))", which means "large or severe spalling". Thus, Figs. 4.4(b) and (d) were acceptably described as containing "failure".

Unfortunately, there is no way to retrieve Image2 using above query expression because *Failure* subsumes *Collapsing*. However, based on EIO, the tool suggests that

annotators must specify the type of *Failure* in the annotation stage or retrieve Image2 when images using a different query *Failure* (Query 2) are searched.

Query 2. Which image has a failure? (Fig. 4.6(b)): The query expression is "Image and hasTarget some (hasDamage some Failure)". The query result is Image2 ~6 and all images having a *Failure* target are detected. Note that Image2 is detected in this query despite of having no syntactic match (*Failure* ≠ *Collapsing*). Here, *hasTarget* is a superclass of *hasObject* and *hasPlace*, which can find damaged objects and places. Thus, the above query is identical to "Image and (*hasObject* some (*hasDamage* some *Failure*) or *hasPlace* some (*hasDamage* some *Failure*))".

Query 3. Which damaged object is located in the basement? (Fig. 4.6(c))**:** A specific object with a certain condition is retrieved. The query expression is "Object and (*hasDamage* some *Damage* and *isLocatedAt* some *Basement*)". This query will detect all images with damaged objects located in the basement. The query result is CaptiveColumn4, which is a captive column in Fig. 4.4(d). When containing these objects are searched, a query of "*Image* and *hasObject* some (above original query expression)" can be written.



(a) Query 1 (b) Query 2 (c) Query 3

Figure 4.6 Examples of query searching results using annotated data in Fig. 4.4.

4.7 Conclusion

As saying goes, one picture is worth a thousand words. This means, in other words, that a single picture contains a large amount of information and can be documented with several descriptions. This study enables documentation and retrieval of various visual semantic contents in earthquake images using the proposed ontology and annotation tool. EIO is created based on vocabularies and their relationships frequently used for descriptions of

earthquake images. Using EIO and image annotation tool, it is demonstrated that the meaning of original descriptions can be transformed into a searchable form using triples to facilitate future retrieval based on the visual contents. Our annotation method can store these descriptions without any degradation or loss in the original meaning. Stored annotation data using the proposed approach can be fully retrieved with various semantic queries.

This annotation method has been designed for earthquake images and is focused, to date, on images of buildings and intended for researchers focused on structural design and performance. It might be possible to extend it to be applicable to describe the contents of images used for other purposes, which is worth consideration in the future. As it stands, this method represents a major step forward toward the development of an ontology and associated annotation method to support relevant scientific research with these images.

CHAPTER 5. VISUAL DATA CLASSIFICATION IN POST-EVENT BUILDING RECONNAISSANCE

5.1 Introduction

After every disaster, a great many images are collected by teams of trained professional engineers, academic researchers and graduate students. The primary functions of a post-disaster reconnaissance team are to collect readily available, perishable data to enable scientific research intended to: (1) learn as much as possible about the nature of the event and extent of the consequences; (2) identify potential gaps in existing research or in the practical application of scientific, economic, engineering or policy knowledge; and (3) make recommendations regarding the need for further investigations, and/or changes to codes, standards and design guidelines. Damaged structures and their components provide critical information regarding performance during the event, and lessons learned from structures that do not experience damage are just as important.

In a typical mission, a group of data collectors is dispatched to a region where an event has taken place. In a well-organized team, information about the local construction, severity of the event, as well as maps of the region are often made available in advance so that planning can take place. The larger group is divided into small teams with at least one more experienced structural engineering evaluator on each team. Each team visits 4-5 buildings a day, collecting images at each site and taking measurements from each building. The teams may follow the procedures outlined in established guidelines (e.g. ATC-20 (earthquakes) and ATC-45 (windstorms and floods)) for this process, which is intended for rapid structural evaluation after events (although these teams are not directly rating these buildings) (www.atcouncil.org). Each evening teams return to the base to discuss the findings and to review plans regarding where to spend time and effort collecting data on the next day.

Currently, the primary approach available to researchers for the analysis of such data is tedious and time-consuming manual sorting and analysis of these photographs or videos. Only a small portion of these data, which are collected at great cost, are being used for scientific purposes or decisions in the field. There is a compelling need to offer

automated tools to aid the structural engineering researcher or the human decision-maker. These tools should provide the analytic power to classify suitable large-scale collections of visual data from disasters in a rapid and efficient manner.

In this study, our goal is to develop and demonstrate a method for the automatic classification of post-disaster images collected during reconnaissance missions. An enabling factor in the proposed method is that a convolutional neural network algorithm (CNNs) implemented for scene (image) classification and object detection is exploited to identify and localize target components of interest on the images. The parameters in the neural network are trained using a large quantity of images so that the resulting trained classifier can achieve robust analysis of the images collected in a disaster, which are typically unordered and complex. This strategy is demonstrated by classifying collapse and detecting and localizing spalling damage using real-world image data.

The key contribution of this research is that a feasible solution is provided for automatically analyzing large-scale collections of real-world images from disasters. Previous research has validated the use of certain techniques (e.g. object detection, damage detection, quantification) using a small quantity of images that were collected with the intention of using them for a particular purpose or application. However, in real circumstances after a disaster, there is no assurance that those techniques will be able to handle large-scale, complex, and unstructured images in such a way as to be tractable. Rather, the proposed method enables accurate and rapid analysis of visual contents in a large volume of real-world images. Herein, the proposed method is demonstrated using an unprecedented number of real images collected from several previous events. Furthermore, the methods developed and validated herein complement past research, and can be coupled with existing methods to incorporate new or existing vision-based damage detection methods for broad application to various situations in a range of disasters. For example, for crack detection, the proposed technique provides images or region of interest on images of target components, which are vulnerable to cracking (e.g. columns or walls after the earthquake).

5.2 Literature Review

In the last few decades, researchers have realized amazing improvements in vision-based structural evaluation for civil engineering applications. These applications span a broad range of studies such as roads, bridges, and buildings. Existing techniques are commonly grouped according to one of the following categories: a given type of structure (e.g. bridge, road, building), damage (e.g. crack, spalling, corrosion), or material (e.g. concrete, steel) (Abdel-Qader et al., 2003; Jahanshahi et al., 2009; Yamaguchi et al., 2010). However, there are still significant challenges in their implementation, and such techniques often require significant modifications to apply them successfully across many situations even when their application is intended for the same target structure or damage type. Strategies for structural evaluation are often completely different depending on how images are collected and what prior information is available for target areas or damage. The existing techniques are classified according to three different categories.

First, images acquired from a known structure with a stationary background are collected using fixed or highly controlled camera(s). The basic premise for this setup is that damage results in a noticeable visual change in the images. Because the scale of the target areas and their visual contents are quite uniform either spatially and temporally, change detection based on established image processing techniques is often sufficient to detect damage. Applications in this category have considered vision-based monitoring of critical areas (Jahanshahi et al., 2011), railroad damage identification (Hashmi and Keskar, 2014), or crack scanning at tunnel (Zhang et al., 2014). It is possible that damage can be quantitatively evaluated based on prior information including the locations of the cameras, their calibration parameters, and the dimensions of the target structures. Existing image processing techniques that have been developed and used for more traditional machine vision applications (e.g. quality inspection, counting) can be directly applicable for applications in this category.

Second, images acquired from a known structure under varying backgrounds are collected. Typical applications in this category consider periodic vision-based visual inspection using images collected by human inspectors or robotic platforms. Specific applications include pavement cracks (Zou et al., 2012) or pothole inspection (Koch and Brilakis, 2011), pipeline inspection (Sinha and Fieguth, 2006), crack quantification

(Jahanshahi and Masri, 2013), or corrosion detection (Jahanshahi and Masri, 2013; Bonnin-Pascual and Ortiz, 2014). Images are collected on a regular basis from the same target inspection locations, but significant background variations are present in these images due to environmental noise (e.g. light variation, shadow), color changes (e.g. material degradation, decolonization) or the presence of irrelevant objects (e.g. debris, bugs, and traffic lines) (Chen et al., 2012). Thus, damage is not the sole source of visual changes in the images, and false positives are frequently triggered unless strong features of damage can be modeled and extracted, or unless the effects of background variations can be reduced. Regardless of such difficulties, once the inspection techniques and their parameters are calibrated in the early stage, the resulting procedures may be repetitively applied to conduct later inspections with just minor updates. Qualitative and quantitative damage evaluations are thus facilitated based on prior knowledge of the dimensions and location of the target inspection areas (Yeum and Dyke, 2015).

Lastly, images containing unknown buildings/structures are collected. This situation is the most challenging in many respects, which is addressed herein for post-disaster evaluation. A major difference from the previous two categories is that image collection processes cannot be optimized in advance for target structure(s) or inspection purposes due to insufficient prior knowledge about the situation and structure. In the literature, a handful of researchers has made contributions to address this field of study due to the complexity involved. Brilakis and co-workers have contributed to the establishment of image-based post-disaster evaluation methods by developing several techniques such as concrete damage evaluation (Zhu et al., 2011), spalling detection (German et al., 2012) or 3D reconstruction using videogrammetry (Brilakis et al., 2011). Recently, Torok et al. developed an image-based 3D reconstruction technique for quantification of cracks and material loss using a structural-from-motion technique (Torok et al., 2014).

However, there are still several realistic challenges in the evaluation of damage using post-disaster images. A major challenge is faced in processing complex, unordered, and unstructured images because many individuals collect large quantities of valuable images from different regions within a short time. As an example of our experience introduced in chapter 5.4, 90,000 real-world images that were collected during past reconnaissance missions for research purposes are gathered. In addition to containing

images intended for reconnaissance purposes, such collections also include a significant number of irrelevant (e.g. people, random objects, vehicles) or corrupted (e.g. blurred, noisy, dirty on the camera lens) images. Within these image collections, reconnaissance teams also frequently incorporate metadata in the form of images such as drawings, GPS devices, or measurements (e.g. image of a structural column with a measuring tape). Such diversity in images should be expected during such a mission, as vast amounts of images are collected by dozens of individuals over a short span of time. Taking steps to filter out unnecessary or irrelevant images before classification and detailed analysis can yield drastic improvements in computational speed and outcomes in the end.

Despite such difficulty, two opportunities are enabling automated analysis of such images. First, computer vision methods and machine learning algorithms have been established within computer science and engineering, and related disciplines. Recent CNNs have led to major breakthroughs in image recognition and object detection and enable the development of high-level abstractions using massive labeled data and parameters (LeCun et al., 1990, 2015; Krizhevsky et al., 2012). They facilitate the development of reliable solutions for object recognition in damage detection applications. Second, a large-scale image database is built gathered from researchers and practitioners after past natural disasters. In computer vision, groun-truth image annotation databases have established the foundation for developing robust object recognition techniques through large-scale training and validation. However, the overwhelming majority of these databases target general, everyday objects and images, rather than real-world visual data having the complexity encountered with building reconnaissance images. Based on these successful examples, it is anticipated that the database will become a domain testbed for a variety of techniques including labeled image data for collapse classification and spalling detection, proposed and demonstrated in this study.

5.3 Technical Approach

In the last few years, CNNs have shed light on several computer vision applications and enabled the learning of high-level and deep features for image recognition using large-scale databases (LeCun et al., 1990, 2015; Krizhevsky et al., 2012; Karpathy et al., 2014; sRussakovsky et al., 2015). 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 numerous convolutional filters. Recent successes of CNNs resulted from training a large number of images, a large number of parameters, “dropout” regularization, and GPU 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 topics of active research in different domains of applications. However, it is evident that CNNs provide exceptional performance in object recognition on natural images (Russakovsky et al., 2015).

Among the several possible implementations of CNNs, facilitating scene (image) classification and object detection for our applications are focused (Xiao et al., 2014; Girshick et al., 2016). Scenes are defined by a subject (e.g. playing soccer, dancing) or a place (e.g. room, street), which is what the entire image represents in general. Scenes from images may be understood by the presence of single or multiple objects and their spatial arrangement. Thus, scenes are considered to be a class of an image (Xiao et al., 2014; Zhou et al., 2014). For instance, a scene of a building façade can be understood by one or more objects and their spatial arrangement including an entrance at the bottom, and an array of windows or floor borders. Scene classification can be performed using coarse resolution images because scenes are typically recognized by low-dimensional features (e.g. general shape, colors, or compositions) and need not be interpreted using the detailed appearance of objects. For example, people often experience that thumbnail images are sufficient to understand the “scene” of an image even though the details regarding the contents are not available. Overall, scene classification is applicable for fast browsing through large visual data to extract relevant scenes of interest where target objects are present as well as to perform image-level classification, not individual contents on images. Fig. 5.1 (a) shows possible scene categories to be useful in our applications, including (1) street, alley, and freeway for roads; (2) building façade, first floor, basement and roof for buildings and (3) building collapse, building foundation, or low-rise building.

Collapse Building façade Column Spalling

(a) (b)

Figure 5.1 Promising applications of scene classification in (a) and object detection in (b) using disaster images: The bounding boxes in (b) are target objects of interest on images.

On the other hand, object detection provides object identity (class) and object location within each image. Unlike scene classification, for object detection trained classifiers from CNNs are applied to small sub-regions of the images rather than over the entire area of the images. Thus, object detection requires an additional step that is used for extracting candidate object areas. In each image, many candidate windows having several different sizes and locations are proposed for accurate detection and localization. In Fig. 5.1(b), possible applications aligned with the procedures for post-earthquake safety evaluation of buildings (ATC-20) include the detection of damage (spalling or cracking in the pattern), structural components (wall, columns, or beams), or falling hazard components (e.g. chimney or parapet). However, small features such as hairline cracks will only be visible in high resolution in the raw images. In this case, for example, a crack detection algorithm must be applied to the relevant target areas in high resolution after extracting base objects. Images in Fig. 5.1 are publically available at datacenterhub.org.

An overview of scene classification and object detection procedures using CNNs is provided in Fig. 5.2. Two-class (binary) classification is implemented to train a complex discriminative boundary between a single class of interest (positive) and the rest (negative). A major difference between scene classification and object detection in the training process is the generation of input images for the CNNs. For scene classification, the original images are resized to be square, and random square regions on the resized image are used as inputs

for the CNNs. On the other hand, for object detection, small sub-regions, called object proposals, become input images (Girshick et al., 2016). The object proposals (red boxes on the image at “object proposals” in Fig. 5.2) having a large overlap with the class of interest (spalling in Fig. 5.2) are assigned as positive, and the remainder of the windows are assigned as negative. Both the original images in scene classification and the object proposals in object detection are directly resized and cropped to be transformed into the input size of the CNNs, typically a square with low resolution (e.g. 200~300 pixel on each side depending on algorithms) (Girshick et al., 2016). Then, the resulting images are augmented to avoid overfitting on translation, color or light variations, by producing more training images without adding new original images. Once the inputs of the CNN are prepared using the original images, 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 of estimating true labels of training images. Stochastic gradient descent with a batch size of images is trained to optimize the parameters in the network. At each epoch, a batch at each iteration is assigned using randomly ordered images after augmentation. Depending on the learning rates, a number of images, network configuration, or hardware specification, the training process typically takes from a couple of hours to some weeks.

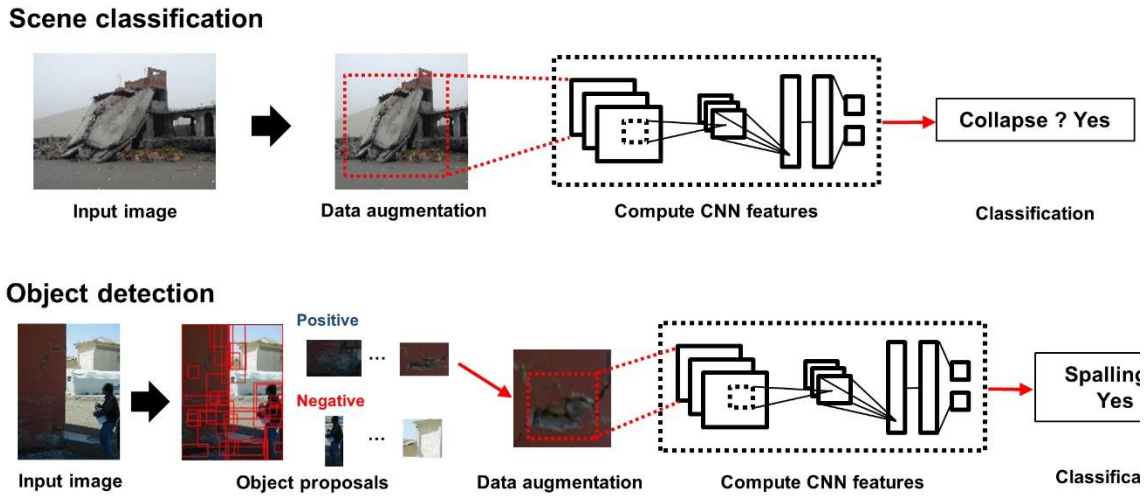


Figure 5.2 Overview of scene classification and object detection using CNNs for binary classification.

5.4 Post-Event Reconnaissance Image Database

Our post-event reconnaissance image database is first introduced. An unprecedented collection of around 90,000 color images is gathered from researchers and practitioners after past natural disasters (hurricane, tornado, and seismic events) (e.g., from datacenterhub.org at Purdue University, disaster responders, or the Earthquake Engineering Research Institute image collection). Sample images in our database are arranged in Fig. 5.3. These samples are publically available, e.g. at datacenterhub.org. Nearly all of the images preserve the original quality (resolution) as well as the basic metadata (e.g. date, time, and event), and a small portion of images contain GPS information or a picture of a GPS navigator. However, no annotations were available to describe the visual contents of the images. At this time, the distribution of the types of disasters is earthquake (90%), hurricane (6%), tornado (3%), and others (1%). These images are collected from several different events such as earthquakes (e.g. Haiti in 2010, L'Aquila in 2009, Nepal in 2015, Taiwan in 2016), hurricanes (e.g. Florida in 2004, Texas in 2008), tornadoes (Florida in 2007, Greensburg in 2007) (datacenterhub.org). Images are still being collected from such natural disasters to integrate into the database.

Figure 5.3 Post-event reconnaissance image database.

In this study, two specific applications are targeted to demonstrate the technique, to classify images of collapsed buildings and building components for scene classification, and to detect spalling in the images for object detection. The reasons for selecting these specific situations are: (1) collapse and spalling are important issues of damage of interest when collecting images for earthquake reconnaissance investigations; (2) a significant

number of images are included in the database for these cases and are thus available for training and validation; and, (3) there is no existing annotation image database for these damage types. To perform this case study, an extensive annotation image database is constructed for validation of the method.

To classify images in this way, the terminology should have a clear definition. A clear definition is necessary to enable human annotators to establish a ground truth data set for training. Our definition of "collapse" for purposes of this study includes images that show buildings or building components that (1) have lost their original shapes; (2) produce a significant amount of debris; or (3) are not serviceable or have restricted access. "spalling" or "flaking" is defined as damage in the image that includes (1) exposed masonry areas in a wall due to cracking followed by flaking; (2) exposed rebar in a column as the concrete cover is lost; or (3) small section loss due to substantial cracking. However, there is considerable variety in the visual appearances for each, and it can be quite a challenge to define the terminology in words clearly. Thus, annotators are provided with a set of sample ground truth images in advance. Some sample images are presented in Fig. 5.4 (datacenterhub.org).

All of the images used for this demonstration are manually labeled from our image database using in-house annotation software. During annotation of collapse scenes and spalling damage, a single image is shown centered in the screen and annotators are asked to answer a yes or no question, for instance, "Does the image contain a collapse scene?" or "Is spalling damage present in the image?" In our experience, such binary classification is most appropriate in that it results in better performance than multiple choice questions. When labeling such a large volume of images, multiple choices are more likely to lead to erroneous selections due to handling many buttons/options, and the resulting decreases in the annotator's concentration as annotation time goes by. Once the images that have spalling are extracted, the areas in each image with spalling are manually identified with a tight bounding box, as demonstrated in Fig. 5.4(b).

(a) (b)

Figure 5.4 Samples of ground truth (labeled) images: (a) collapse and (b) spalling.

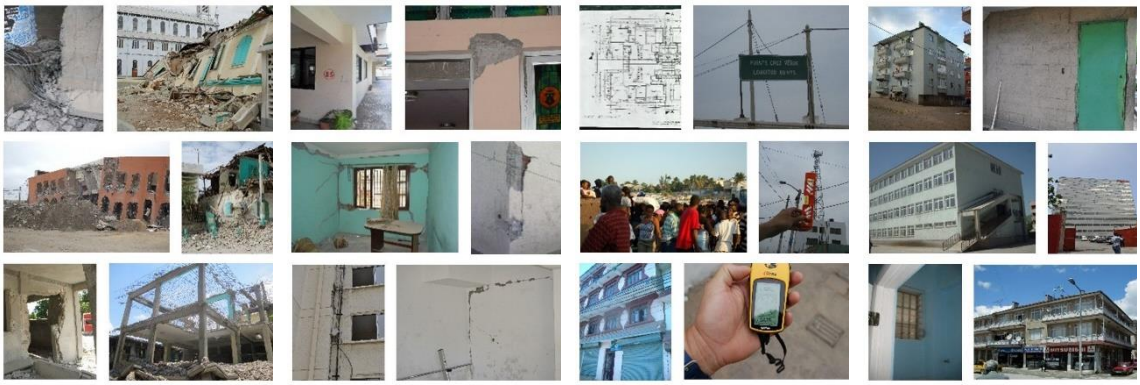
5.5 Collapse Classification

5.5.1 Configuration

In practical implementations, a trained binary classifier will be applied to distinguish images containing collapsed buildings and building components (hereafter, collapse images) from a whole set of the images. This collection of images contains, in addition to the images of buildings and their components, other images with irrelevant or extraneous visual contents. For instance, images of cars, people, drawings, rubble, and so forth, may represent a small portion of the image collection which must also be dealt with. Thus, appropriate unbiased negative samples should be included in the training images, and these should have similar visual content or appearance as the ones used for actual implementation (testing). In other words, for training classifiers for the detection of collapse (positive) images, undamaged buildings, and building components should not solely be included in the set of negative images.

For demonstration purposes, a dataset collected during earthquake reconnaissance missions is manually annotated. Our dataset is composed of 1,850 collapse images (as defined in chapter 5.4) as positive and 3,420 non-collapse images as negative, for a total of 5,270 images. Non-collapse images mainly consist of undamaged buildings, damaged buildings, and irrelevant pictures, annotated from our image database, which represents a typical data set collected during an earthquake reconnaissance mission. The sample images

for this category are shown in Fig 5.5. All labeled images are divided into 2,636 (50%), 1,317 (25%), and 1,317 (25%) images for training, validating and testing.



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

Figure 5.5 Sample images used for collapse classification: (a) collapse buildings, (b) damage on buildings, (c) irrelevant images, and (d) undamaged buildings: (a) is assigned in a positive class and the others are in a negative class.

In this study, a popular ImageNet CNN model called Alexnet (TorontoNet in Caffe) is implemented, which is framed in MatConvNet library (Vedaldi and Lenc, 2014). Alexnet exhibited superior implementation of 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 (Russakovsky et al., 2015). The network architecture is presented in detail in the following reference (Krizhevsky et al., 2012). Although some improved architectures have also been introduced and demonstrated after 2012, in this study, Alexnet is selected as an established method to implement this proof of concept of a simple and general CNN model.

As a pre-processing step, the raw images are first resized to 256 x 256 (pixels) to produce a set of regularized input images. In the original implementation, the images were rescaled such that the shorter side of the image becomes 256 and then cropped out the central 256 x 256 patch from the resulting image (Krizhevsky et al., 2012). This general-purpose method was based on the fact that a majority of images collected by humans are captured in such way that its subject is located at the center of the image. However, because of dealing with scientific images, a different resizing and cropping strategy is applied. The images are directly stretched or shrunk to 256 x 256 without preserving their aspect ratios. This approach is valid because the way engineers capture and understand collapse through images is not particularly dependent on the actual aspect ratio of the visual contents of the

images. Cropping out a horizontal or vertical edge of the original image would cause a loss of important visual features needed to recognize collapse. For example, debris can be a crucial visual clue (and feature) and typically appears at the bottom of images. Our resizing method can avoid accidentally eliminating these features on the input images. Once the set of images is resized, the mean RGB value is subtracted on the training set from each pixel to generate features that are unbiased on colors (Girshick et al., 2016).

For data augmentation, 227 x 227 patches are randomly cropped from each of the 256 x 256 images in each epoch. For further augment the training set, these patches are randomly flipped horizontally (mirroring) for inclusion in the data set. The last 1000-way softmax layers in the original implementation of the ImageNet competition is modified to 2-way ones for binary classification (Russakovsky et al., 2015). 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 (Krizhevsky et al., 2012). Our models are trained using stochastic gradient descent with a batch size of 256 images, the momentum of 0.9, and weight decay of 0.0005. The network is trained for 300 epochs, and the learning rate is logarithmically decreased from 0.01 to 0.0001 during training. A workstation having a Xeon E5-2609 CPU and a GPU, NVidia Titan X with 12 GB video memory is used for training and testing the algorithm. The MatConvNet library installed on Matlab 2016a is used for this study (Vedaldi and Lenc, 2014). Training of 2,636 images including 1,317 images for validation in each epoch takes around one minute.

5.5.2 Collapse classification results

In this study, collapse classification successfully attains a relatively high accuracy. Rates of 90.26% (417/462 images) true-positive (true classification of collapse images) and 92.16% (788/855 images) true-negative are obtained, respectively. The precision is 0.862, defined as the number of true positives over the number of positives. Random samples of testing results with their predicted labels are shown in Fig. 5.6. Figs. 5.6(a) and (b) are obtained from ground-truth collapse and non-collapse images, respectively. Although these rates will vary slightly depending on CNN architectures and their parameters, the overall performance of this approach is quite successful.

(a)

(b)

Figure 5.6 Random selection of images with the predicted classes: (a) ground-truth collapse images and (b) ground-truth non-collapse images. All images listed here are transformed into a square for the arrangement.

5.6 Spalling Detection

5.6.1 Configuration

Unlike collapse classification, spalling detection requires an additional step for localizing one or many regions with spalling within an image. One popular approach is that the trained classifier is applied to each of numerous small sub-regions across each image and makes a decision on whether it contains or represents a target object of interest. Historically, sliding window detectors are typically implemented, which consist of scanning the entire image area with fixed windows (Viola and Jones, 2001; Zitnick and Dollar, 2014). However, its implementation requires the assumption that all objects to share a common aspect ratio, such as a pedestrian or face. Thus, this approach is not feasible for spalling detection due to the tremendous uncertainty in the size and shape (aspect ratio) of the spalling to be detected. Instead, the concept of object proposals is adopted through the extraction of

candidate segmentations (a portion of the image) derived from colors, saliency or edges, implemented using regional based convolutional neural networks (R-CNN) (Girshick et al., 2016). This approach assumes that a target object is included in at least one of these segmented regions, facilitating localization of regions of interest. For spalling detection, the use of this method is highly acceptable because spalling is visually a blob-like region having prominent colors and textures as compared to the background, and thus one of detected object proposal windows includes a true object with a high accuracy. The selective search algorithm, which is well known for robust object proposal detection, is used for this study (Uijlings et al., 2013).

A total of 1,086 images with 3,158 spalling annotations are used for this demonstration. All labeled images are divided into 75% (815 images) for training and validation and 25% (271 images) for testing. Contrary to collapse classification, object detection does not require a separate a set of negative images (e.g. non-spalling images) because all object proposals which do not have a large overlap with a ground-truth box of spalling are inherently treated as negative images. The raw images are rescaled such that the shorter side of the image becomes 512 pixels. The selective search algorithm in a fast mode is performed on each of the rescaled images, which generates around 2,000 ~ 4,000 object proposals in each image. Object proposals with ≥ 0.3 intersection-over-union (IoU) overlap with a ground-truth box of spalling are selected as positive images, and the remainder of the object proposals is assigned as negative images (Russakovsky et al., 2015; Zitnick and Dollar, 2014). Object proposals are not tested if the length of one of the four sides in the box is less than 20 pixels because it is too small to be worth evaluating. Ground-truth spalling images with a width or height smaller than 20 pixels are removed. Overall, the selective search algorithm produces around 10~30 positive object proposal in each ground-truth spalling. As a result, 815 training images generate 65,652 positive and 2,075,453 negative object proposals. 75% of positive and negative object proposals (49,239 and 1,556,590) are used for training, and the remainder is assigned for validation. The object detection strategy assumes that at least one object proposal is generated from each spalling location visible in the image. In our dataset, positive object proposals cover 98.8% of ground-truth spallings (3,119/3,158 spalling), which yields a high rate of object inclusion (Russakovsky et al., 2015).

Fig. 5.7 shows an example that explains how to obtain input images (object proposals) for training a CNN. First, a tight bounding box is annotated for each spalling location, marked as blue boxes in Fig. 5.7(a). Among the extracted object proposals (2,469 in this sample image), those that have a certain amount of overlap with ground-truth spalling are assigned as positive, marked as green, (68 object proposals) in Fig. 5.7(b) and the rest become negative, marked as red (2,401 object proposals) in Fig. 5.7(c). For visual clarity, only 50 negative object proposals are randomly selected in Fig. 5.7(c). The sample object proposals are shown in Fig. 5.7(d). Then, these images are warped to be a square for inputs to the CNN as in Fig. 5.7(e).

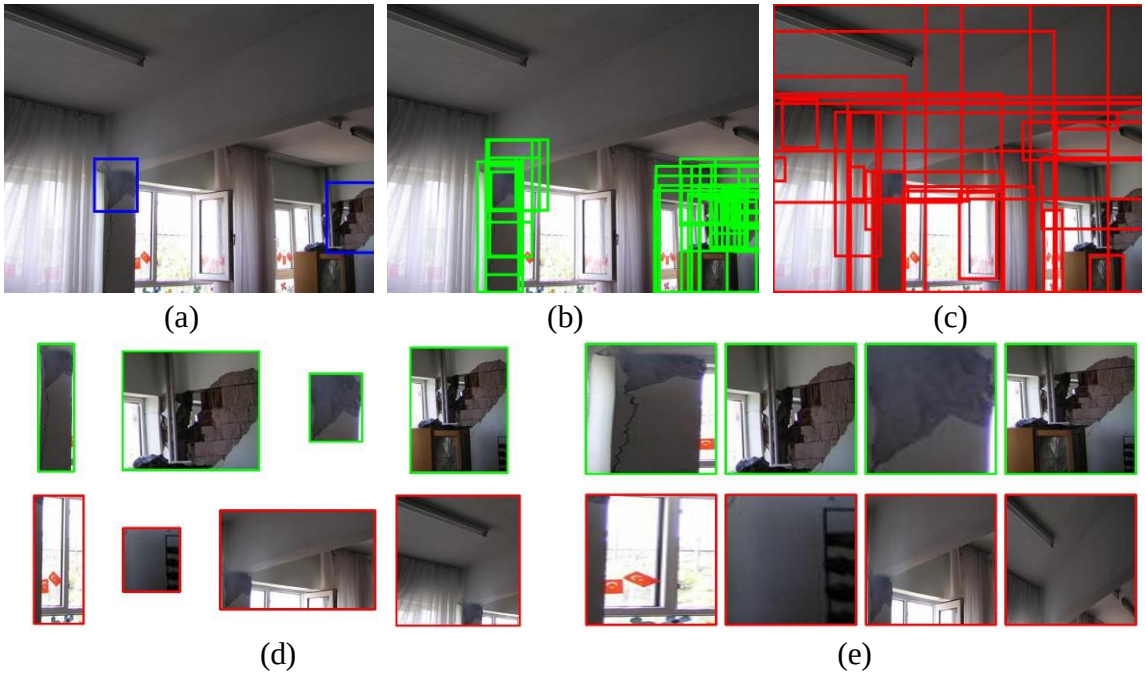


Figure 5.7 Generation of positive and negative images for training CNN: (a) ground-truth spallings, (b) positive object proposals, (c) negative object proposals, (d) samples of object proposals with original aspect ratios, and (e) transformation of object proposals as a square to be an input for CNN. Note that the top and bottom rows in (d) and (e) are positive and negative object proposals.

Once valid object proposals for positive and negative are extracted, they must be transformed into inputs for the CNNs. All object proposals are warped to 256 x 256, as with those used in collapse classification. Although this process may produce considerable distortion when the object proposals have high aspect ratio, it is advantageous in that warping associated with stretching or shrinking can contain the entire contents of the original images. For data augmentation, 227 x 227 patches are randomly cropped from 256

x 256 images in each epoch and random horizontal flipping is allowed. All configurations are the same as in the previous experiment except for the method for batch assignment and normalization. In each iteration, 153 positive and 359 negative patches are uniformly assigned to construct a batch of 512 patches. the sampling is intentionally biased towards positive object proposals because the number of positive object proposals is extremely small in number compared to the negative ones. Without using biased sampling, the classifier is trained toward predicting all negatives due to this large unbalance in the number of training samples. In each epoch, a total of 4,348 batches and 1,046 batches for training and validation are built, respectively. It takes around 6~7 hours for each epoch. Batch normalization is additionally implemented for training (Ioffe and Szegedy, 2015). Batch normalization allows us to use much higher learning rates but provides a fast rate of learning. The network is trained for 20 epochs, and the learning rate is logarithmically decreased from 0.1 to 0.0001 during training. The batch normalization technique is implemented in the MatConvnet library (Vedaldi and Lenc, 2014).

At the test step, object proposals are first extracted from the test images and warp them in the same manner as in the training. 271 test images having 814 spalling locations, generates 704,314 object proposals. Each of the object proposals is forward-propagated through the trained CNN to identify its class (spalling or not) and its confidence score. Then, all scored object proposals in an image predicted as spalling (positive) are identified. Multiple overlapping positive object proposals are typically detected close to the correct location of spalling. For accurate estimation of the spalling locations, these repetitious predictions are penalized or eliminated using confidence values of each predicted object proposals to find the best fit a bounding box encompassing the corresponding spalling (Girshick et al., 2016). A simple non-maximum suppression algorithm is implemented. The confidence scores of the detected object proposals are sorted, and the highest scoring ones are greedily selected, denoted as **A**, while skipping detections with the object proposal, denoted as **B** if area $(\mathbf{A} \cap \mathbf{B})$/ min (area $(\mathbf{A})$, area $(\mathbf{B})) > 0.3$ (MATLAB, 2015). Fig. 5.8 shows an example of multiple positive (spalling) detections followed by localizing the bounding box using non-maximal suppression. Multiple positive object proposals in Fig. 5.8(a) are thus reduced to a limited number as shown in Fig. 5.8(b).

(a) (b)

Figure 5.8 Localization of spalling refined by a non-maximal suppression: (a) all object proposals predicted as positive and (b) final predicted spalling location after applying non-maximal suppression. The blue box is a ground-truth spalling. Note the numeric value on the top of each of the bounding boxes is a confidence value.

5.6.2 Spalling detection results

First, how well the trained classifier correctly identifies the class of object proposals is evaluated. Among 704,314 object proposals extracted from a set of test images, the trained classifier predicts 16,454 object proposals as positive and remainder, as negative. Rates of 59.39% of true-positive (9,772/16,454 object proposals) and 1.7% of false negative (11,965/687,860 object proposals) are obatined. This result means that approximately 60% of the positive object proposals are correctly predicted as positive, and those do have a large overlap (> 0.3 IoU overlaps), while the remainder (around 40%) are falsely predicted, and those are generated either from the background or without sufficient overlap of the actual spalling.

Next, repeated detection of object proposals at similar locations is penalized to predict the most accurate location of spalling(s). Finally, a total of 1,529 bounding boxes through non-maximum suppression are identified, and from these 619 spalling locations are correctly detected and localized (> 0.3 IoU overlap). The true positive rate is 40.48% (619/1529) and detected bounding boxes capture 62.16% (506/814) of all of the ground-truth spallings. Since multiple bounding boxes are detected near some spalling locations, the number of bounding boxes (619) is somewhat larger than then number of detected ground-truth spallings (506). Some sample images of predicted spalling locations are shown in Fig. 5.9. Note that these rates will also vary depending on the CNN architectures and their parameters. Overall the performance of this approach is reasonably successful

and with this trained classifier, one of two detected bounding boxes likely include true spalling regions and they cover two-third of entire spallings on the testing images.

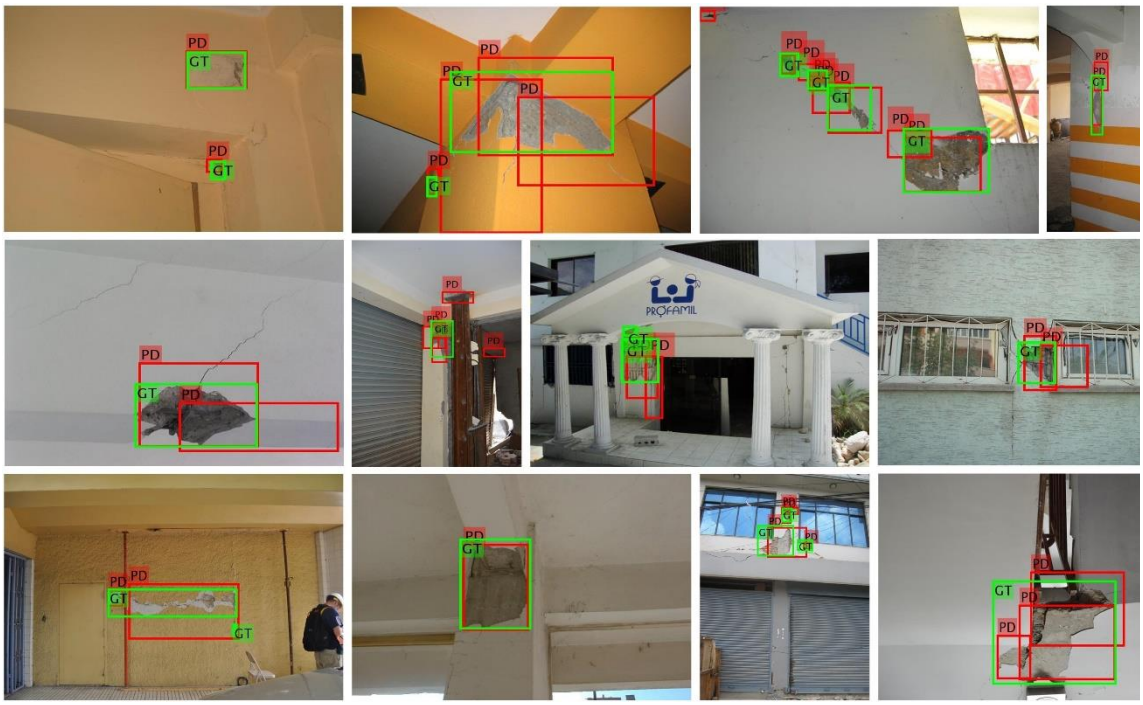


Figure 5.9 Samples of spalling detection: Green boxes with a tag of “GT” are ground-truth spalling areas, and red boxes with a tag of “PD” are predicted ones. Note that the aspect ratios of some images are changed for arrangement in this figure.

5.7 Conclusion

This study discusses a method for automating the process of post-event reconnaissance image analysis. The emerging class of techniques known as convolutional neural network algorithms is implemented to autonomously classify images by a scene-of-interest, and identify and localize target objects on the images. By incorporating the necessary domain knowledge, data augmentation and standardization methods that are suitable for this application are chosen and implemented. This strategy is successfully applied to classify image data in our post-event reconnaissance image database. Specific classification examples are considered to demonstrate the technique. Collapse image classification and spalling detection are identified from extensive collection of images. High classification accuracy in both cases is successfully achieved. A key contribution of this work is to develop and validate the method using a first-of-a-kind post-event database containing real-

world images collected from past natural disasters. A significant volume of images our database enables training robust classifiers that can be applied for analysis and classification visual contents. This strategy will be a breakthrough in the understanding and analyses of post-event images in a rapid and efficient way.

CHAPTER 6. SUMMARY AND CONCLUSION

In this dissertation, computer vision-based visual assessment techniques have been developed for civil engineering applications, focusing on the specific needs associated with visual inspection, and classification and documentation of visual data collected from post-event reconnaissance. Harnessing the capability to use new visual sensors, sensing platforms, and high-performance computation resources, the techniques developed herein are capable of automated visual assessment in a rapid and efficient manner. Such research, which occurs at the intersection of disciplines, often results in significant contributions impacting multiple fields. Overall, the key contribution of the techniques developed is to serve as a solid foundation for enabling realistic vision-based visual assessment for civil engineering applications by exploiting computer vision techniques to their full potential. In past research, the only small quantity of images was collected or used in the development and validation of visual assessment techniques. Because those images are often captured under controlled circumstances, such as from favorable angles and distances with the intention of using them with specific algorithms, their use is limited for demonstrating their robustness and feasibility in real-world applications. However, in the real world, such specific and high-quality images are rarely collected or available, and cannot be isolated from the larger collection. In this study, the focus is on using large volumes of realistic images for enhancing and improving computer vision techniques as well as for validating those techniques. Here, modern computer vision techniques are fully incorporated in a task-oriented manner based on the needs of the civil engineer, demonstrating that they can efficiently and accurately accommodate the complexity observed in real-world data sets. To my best knowledge, there is no literature that considers the development of vision-based visual assessment techniques using these concepts.

Several techniques are proposed and validated. A brief summary of each technique are summarized as follows:

- In chapter 2, a new damage detection technique is developed which can automatically process and analyze a large volume of images. Rather than searching for damage over the entire area of the images, objects that have areas susceptible to

damage (crack on bolts in this study) are first detected in all of the images with appropriate object recognition techniques. In addition, damage is detected based on prior engineering knowledge of how damage form on the object. Incorporating this experience increases its detectability, and also decreases false-positive detection. Using images from many different angles and prior knowledge of the typical appearance and characteristics of this class of faults, the technique can successfully detect cracks near bolts on a large steel beam.

- In chapter 3, an automated image localization technique is developed to extract regions of interest (ROIs) on each of the images in a large set of image data before utilizing vision-based inspection techniques. ROIs are the portions of an image that contain the region of the structure that is targeted for visual interrogation. ROIs can be computed based on the geometric relationship between the collected images and the target areas on the structure. Analysis of such highly relevant and localized images would enable efficient and reliable visual inspection. The capability of the technique is successfully demonstrated to extract the ROIs of weld connections using a full-scale highway sign structure.
- In chapter 4, the next generation of a structured annotation method is developed for describing the semantic contents of images originating from earthquake reconnaissance. Images of buildings focused on structural design and performance resisting earthquake are explored. An earthquake image ontology (EIO) is designed for formalized and structured descriptions of images. EIO integrates a broad set of terms and the associated relationships, enabling rich descriptions of images based on their contents and addressing the types of queries of interest to researchers. It is adequately extensible and flexible to successfully deal with a much broader range of images in the future. An image annotation tool assists human annotators as they choose appropriate terms and their relations in EIO. EIO facilitates image annotation and conversion of data into a searchable form using various queries. The feasibility and usefulness of the method are demonstrated using images from past earthquake.
- In chapter 5, a method for post-disaster evaluation is developed by processing and analyzing big visual data in an autonomous manner. Recent convolutional neural

network (CNN) algorithms are implemented to extract the visual content of interest automatically from the collected images. Image classification and object detection are incorporated into the procedures to achieve accurate extraction of target contents of interest. As an illustration of the technique and its capabilities, two specific cases, collapse classification and spalling detection in concrete structures are demonstrated using a large volume of images gathered from past earthquake disasters.

Some recommendations for future studies are scheduled to expand and improve the related techniques developed in this dissertation:

- Promising technology applicable to drone-based visual inspection will be explored to build more robust, efficient, and fast systems. Recent developments in depth cameras, including Lidar, time-of-flight (ToF) cameras, and RGB-D cameras will expand an inspection by adding an extra depth dimension, which is not exactly captured with human eyes. These new devices represent an opportunity for 3D modeling of a structure and localization (navigation) in pre-built maps. They also can be integrated with a mobile platform due to their light weight and low cost (although it does depend on its range and accuracy). In addition, affordable and light real-time kinetic GPS enable high precision (centimeter-level) positioning. This technology will be implemented for autonomous vehicles (elaborate flight plan), aerial surveying and other uses in the future.
- The autonomous image localization technique in chapter 2 will be explored to facilitate lifecycle management of infrastructure systems using citizen science and crowdsourcing images, as a part of National Science Foundation under Grant No. NSF-1645047. Citizen science and crowdsourcing provide the capacity to collect a large number of photos of certain structures, from many perspectives, at frequent intervals, and under many conditions. The developed techniques will provide an opportunity to make use of visual data collected from citizens to exploit automation and information extraction from images. With this technique, the human inspector and decision-maker can automatically extract and view the portion of every image that contains the core region of the structure to be examined.

- Applications of the developed image classification and object detection will be comprehensively searched (e.g. the objects and scenes listed in chapter 5.3). Collapse and spalling are just one of many examples that can benefit from the developed techniques. This work is a part of National Science Foundation under Grant No. NSF-1608762 and related works will continue to build more robust and various classifiers by expanding the number of image collections and their ground-truth annotations.

REFERENCES

Abdel-Qader, I., Abudayyeh, O., & Kelly, M. E. (2003). Analysis of Edge-Detection Techniques for Crack Identification in Bridges. Journal of Computing in Civil Engineering, 17(4), 255–263.

Adams, S. M., & Friedland, C. J. (2011). A survey of unmanned aerial vehicle (UAV) usage for imagery collection in disaster research and management. publisher not identified. Retrieved from http://blume.stanford.edu/sites/default/files/RS_Adams_Survey_paper_0.pdf

Agisoft. (2016). Photoscan. Retrieved from http://www.agisoft.com/

Applied Technology Council. (2016). ATC 20-2 Rapid Evaluation Safety Assessment Form. Retrieved from https://www.atcouncil.org/pdfs/rapid.pdf

Baggio C, Bernardini A, Colozza R, Corazza L, Della Bella M, Di Pasquale G, Dolce M, Goretti A, Martinelli A, Orsini G, Papa F. (2007). Field Manual for Post-Earthquake Damage and Safety Assessment and Short Term Countermeasures (AeDES) - JRC Science Hub - European Commission.

Bentley. (2016). ContextCapture. Retrieved from https://www.bentley.com/en/products/brands/contextcapture

Bonnin-Pascual, F., & Ortiz, A. (2014). Corrosion Detection for Automated Visual Inspection. In M. Aliofkhazraei, Developments in Corrosion Protection. InTech.

Bönström, V., Hinze, A., & Schweppe, H. (2003). Storing RDF as a graph. In Web Congress, 2003. Proceedings. First Latin American (pp. 27–36).

Bosi, A., Kotinas, I., Martínez, I. L., Bousias, S., Chazelas, J. L., Dietz, M., … Pegon, P. (2015). The SERIES Virtual Database: Exchange Data Format and Local/Central Databases. In F. Taucer & R. Apostolska (Eds.), Experimental Research in Earthquake Engineering (pp. 31–48).

Bowen Laboratory - Lyles School of Civil Engineering - Purdue University. (2016). Retrieved from https://engineering.purdue.edu/CE/Bowen

Brilakis, I., Fathi, H., & Rashidi, A. (2011). Progressive 3D reconstruction of infrastructure with videogrammetry. Automation in Construction, 20(7), 884–895.

Chaiyasarn, K., Kim, T.-K., Viola, F., Cipolla, R., & Soga, K. (2016). Distortion-Free Image Mosaicing for Tunnel Inspection Based on Robust Cylindrical Surface Estimation through Structure from Motion. Journal of Computing in Civil Engineering, 30(3), 4015045.

Chen, P.-H., Shen, H.-K., Lei, C.-Y., & Chang, L.-M. (2012). Support-vector-machine-based method for automated steel bridge rust assessment. Automation in Construction, 23, 9–19.

Chen, X., Wei, P., Ke, W., Ye, Q., & Jiao, J. (2014). Pedestrian detection with deep convolutional neural network. In Computer Vision-ACCV 2014 Workshops (pp. 354–365). Springer.

Cireşan, D. C., Giusti, A., Gambardella, L. M., & Schmidhuber, J. (2013). Mitosis detection in breast cancer histology images with deep neural networks. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013 (pp. 411–418). Springer.

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

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (Vol. 1, pp. 886–893). IEEE.

Deans, S. R. (2007). The Radon Transform and Some of Its Applications. Courier Corporation.

Design safe-ci. (2016). A cloud-based environment for research in natural hazards engineering. Retrieved from www.designsafe-ci.org

Dollár, P., Tu, Z., Perona, P., & Belongie, S. (2009). Integral channel features.

Dublin Core. (2016). Dublin Core. Retrieved from http://dublincore.org/

Earthquake engineering research institute (EERI). (2016). Learning from Earthquakes Retrieved from https://www.eeri.org/projects/learning-from-earthquakes-lfe/

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 32(9), 1627–1645.

FEMA. (2015a). Earthquake-Resistant Design Concepts: An Introduction to the NEHRP Recommended Seismic Provisions for New Buildings and Other Structures | FEMA.gov. Retrieved from https://www.fema.gov/media-library/assets/documents/21866

FEMA. (2015b). Rapid Visual Screening of Buildings for Potential Seismic Hazards | FEMA.gov. Retrieved from https://www.fema.gov/media-library/assets/documents/15212

Frangi, A. F., Niessen, W. J., Vincken, K. L., & Viergever, M. A. (1998). Multiscale vessel enhancement filtering. In Medical Image Computing and Computer-Assisted Interventation—MICCAI'98 (pp. 130–137). Springer. Retrieved from http://link.springer.com/chapter/10.1007/BFb0056195

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). The Annals of Statistics, 28(2), 337–407. Retrieved from https://doi.org/10.1214/aos/1016218223

German, S., Brilakis, I., & DesRoches, R. (2012). Rapid entropy-based detection and properties measurement of concrete spalling with machine vision for post-earthquake safety assessments. Advanced Engineering Informatics, 26(4), 846–858.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2016). Region-based convolutional networks for accurate object detection and segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 38(1), 142–158.

Guan, J., Deboeverie, F., Slembrouck, M., van Haerenborgh, D., van Cauwelaert, D., Veelaert, P., & Philips, W. (2015). Extrinsic Calibration of Camera Networks Using a Sphere. Sensors, 15(8), 18985–19005.

Gur, T., Pay, A., Ramirez, J. A., Sozen, M. A., Johnson, A. M., Irfanoglu, A., & Bobet, A. (2009). Performance of School Buildings in Turkey During the 1999 Düzce and the 2003 Bingöl Earthquakes. Earthquake Spectra, 25(2), 239–256.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., … Ng, A. Y. (2014). Deep Speech: Scaling up end-to-end speech recognition. arXiv:1412.5567.

Hartley, R. I., & Zisserman, A. (2004). Multiple View Geometry in Computer Vision (Second). Cambridge University Press, ISBN: 0521540518.

Hashmi, M. F., & Keskar, A. G. (2014). Computer-Vision Based Visual Inspection and Crack Detection of Railroad Tracks. Recent Advances in Electrical and Computer Engineering, 102–110.

Hollink, L., Schreiber, G., Wielemaker, J., Wielinga, B., & others. (2003). Semantic annotation of image collections. In Knowledge capture (pp. 41–48).

Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America A, 4(4), 629.

Horridge, M., Knublauch, H., Rector, A., Stevens, R., & Wroe, C. (2004). A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools Edition 1.0. University of Manchester.

Im, D.-H., & Park, G.-D. (2014). Linked tag: image annotation using semantic relationships between image tags. Multimedia Tools and Applications, 74(7), 2273–2287.

Indiana Department of Transportation. (2013). INDIANA BRIDGE INSPECTION MANUAL. Retrieved from http://www.in.gov/dot/div/contracts/standards/bridge/inspector_manual/INDIANA%20BRIDGE%20INSPECTION%20MANUAL.pdf

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv Preprint arXiv:1502.03167. Retrieved from http://arxiv.org/abs/1502.03167

iWitness. (2016). Retrieved from http://www.iwitnessphoto.com/

Jahanshahi, M. R., Kelly, J. S., Masri, S. F., & Sukhatme, G. S. (2009). A survey and evaluation of promising approaches for automatic image-based defect detection of bridge structures. Structure and Infrastructure Engineering, 5(6), 455–486.

Jahanshahi, M. R., & Masri, S. F. (2013). A new methodology for non-contact accurate crack width measurement through photogrammetry for automated structural safety evaluation. Smart Materials and Structures, 22(3), 35019.

Jahanshahi, M. R., Masri, S. F., & Sukhatme, G. S. (2011). Multi-image stitching and scene reconstruction for evaluating defect evolution in structures. Structural Health Monitoring, 10(6), 643–657.

Johnson, J., Krishna, R., Stark, M., Li, L.-J., Shamma, D. A., Bernstein, M. S., & Fei-Fei, L. (2015). Image retrieval using scene graphs. In Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on (pp. 3668–3678). IEEE.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 1725–1732).

Koch, C., & Brilakis, I. (2011). Pothole detection in asphalt pavement images. Advanced Engineering Informatics, 25(3), 507–515.

Kraus, K. (2011). Photogrammetry, Geometry from Images and Laser Scans (2nd. ed.). Berlin, Boston: De Gruyter.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097–1105).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., & Jackel, L. D. (1990). Handwritten Digit Recognition with a Back-Propagation Network. In D. S. Touretzky (Ed.), Advances in Neural Information Processing Systems 2 (pp. 396–404).

Lee, B. J., Shin, D. H., Seo, J. W., Jung, J. D., & Lee, Y. J. (2011). Intelligent bridge inspection using remote controlled robot and image processing technique. In International Symposium on Automation and Robotics in Construction (ISARC), Seoul, Korea (pp. 1426–1431).

Lee, S., Chang, L.-M., & Skibniewski, M. (2006). Automated recognition of surface defects using digital color image processing. Automation in Construction, 15(4), 540–549.

Lindeberg, T. (1994). Scale-Space Theory in Computer Vision. Norwell, MA, USA: Kluwer Academic Publishers.

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

MATLAB. (2015). MATLAB and Image Processing Toolbox.

Miller, J. (2004). Robotic Systems for Inspection and Surveillance of Civil Structure. University of Vermont. Retrieved from http://www.robot.bmstu.ru/files/books/Robotic%20Systems%20for%20Inspection,%20Surveillance%20of%20Civil%20Structures.pdf

Moehle, J. P., Ghodsi, T., Hooper, J. D., Fields, D. C., & Gedhada, R. (2011). Seismic Design of Cast-in-Place Concrete Special Structural Walls and Coupling Beams. NEHRP Seismic Design Technical Brief No, 6. Retrieved from http://www.nehrp-consultants.org/publications/download/nistgcr11-917-11REV1.pdf

Moller, P. S. (2008). CALTRANS Bridge Inspection Aerial Robot (contract number UCD02-02371) Final Report.

Moore, M., Phares, B. M., Graybeal, B., Rolander, D., & Washer, G. (2001). Reliability of visual inspection for highway bridges, volume I: Final report. Retrieved from http://trid.trb.org/view.aspx?id=680608

Moulon, P., Monasse, P., & Marlet, R. (2012). Adaptive structure from motion with a contrario model estimation. In Computer Vision–ACCV 2012 (pp. 257–270). Springer. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-37447-0_20

Neogi, N., Mohanta, D. K., & Dutta, P. K. (2014). Review of vision-based steel surface inspection systems. EURASIP Journal on Image and Video Processing, 2014(1), 1–19.

New America. (2015). Drone and aerial observation. Retrieved from http://drones.newamerica.org/primer/

Noy, N. F., McGuinness, D. L., & others. (2001). Ontology development 101: A guide to creating your first ontology. Stanford knowledge systems laboratory technical report KSL-01-05 and Stanford medical informatics technical report SMI-2001-0880, Stanford, CA.

Ozden, K. E., Schindler, K., & Gool, L. V. (2010). Multibody Structure-from-Motion in Practice. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(6), 1134–1141.

Photomodeler. (2016). Photomodeler Scanner. Retrieved from

http://www.photomodeler.com/index.html

Pix4D. (2016). Pix4D. Retrieved from https://pix4d.com/

Portland Cement (2001). Concrete slab surface defects: Causes, prevention, repair. Portland Cement Association. Retrieved from http://www.oboa.on.ca/events/2009/sessions/files/Slab%20Surface%20Prevention%20Repair.pdf

Pustejovsky, J., & Stubbs, A. (2013). Natural language annotation for machine learning. Sebastopol, CA: O'Reilly Media. Retrieved from https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., … Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 115(3), 211–252.

Schanda, J., & International Commission on Illumination (Eds.). (2007). Colorimetry: understanding the CIE system.

Schreiber, A. T. G., Dubbeldam, B., Wielemaker, J., & Wielinga, B. (2001). Ontology-based photo annotation. IEEE Intelligent Systems, (3), 66–74.

Shah, P., Pujol, S., Puranam, A., & Laughery, L. (2015). Database on Performance of Low-Rise Reinforced Concrete Buildings in the 2015 Nepal Earthquake. Retrieved from https://datacenterhub.org/resources/238

Sinha, S. K., & Fieguth, P. W. (2006). Automated detection of cracks in buried concrete pipe images. Automation in Construction, 15(1), 58–72.

Snavely, N., Seitz, S. M., & Szeliski, R. (2008). Modeling the World from Internet Photo Collections. International Journal of Computer Vision, 80(2), 189–210.

Staab, S., & Studer, R. (Eds.). (2009). Handbook on Ontologies. Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-540-92673-3

Sun, J., He, H., & Zeng, D. (2016). Global Calibration of Multiple Cameras Based on Sphere Targets. Sensors, 16(1), 77.

Szeliski, R. (2010). Computer Vision: Algorithms and Applications (1st ed.). New York, NY, USA: Springer-Verlag New York, Inc.

Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In 2014 IEEE Conference on Computer Vision and Pattern Recognition (pp. 1701–1708).

Telleen, K., Maffei, J., Heintz, J., & Dragovich, J. (2012). Practical lessons for concrete wall design, based on studies of the 2010 Chile earthquake. In Proceedings of the 15th world conference on earthquake engineering, 15WCEE, Lisboa (pp. 24–28). Retrieved from http://www.iitk.ac.in/nicee/wcee/article/WCEE2012_4435.pdf

Torok, M. M., Golparvar-Fard, M., & Kochersberger, K. B. (2014). Image-Based Automated 3D Crack Detection for Post-disaster Building Assessment. Journal of Computing in Civil Engineering, 28(5), A4014004.

Torralba, A., Murphy, K. P., & Freeman, W. T. (2004). Sharing features: efficient boosting procedures for multiclass object detection. In Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on (Vol. 2, p. II–762). IEEE. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1315241

Uijlings, J. R., van de Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition. International Journal of Computer Vision, 104(2), 154–171.

Vedaldi, A., & Fulkerson, B. (2010). VLFeat: An open and portable library of computer vision algorithms. In Proceedings of the 18th ACM international conference on Multimedia (pp. 1469–1472). ACM. Retrieved from http://dl.acm.org/citation.cfm?id=1874249

Vedaldi, A., & Lenc, K. (2014). MatConvNet - Convolutional Neural Networks for MATLAB. arXiv:1412.4564. Retrieved from http://arxiv.org/abs/1412.4564

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 1, p. I–511). Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=990517

W3C. (2008). SPARQL Query Language for RDF. Retrieved from https://www.w3.org/TR/rdf-sparql-query/

W3C. (2007). Image Annotation on the Semantic Web. Retrieved from https://www.w3.org/2005/Incubator/mmsem/XGR-image-annotation/

Warren Mills, J., Curtis, A., Pine, J. C., Kennedy, B., Jones, F., Ramani, R., & Bausch, D. (2008). The clearinghouse concept: a model for geospatial data centralization and dissemination in a disaster. Disasters, 32(3), 467–479.

Westoby, M. J., Brasington, J., Glasser, N. F., Hambrey, M. J., & Reynolds, J. M. (2012). "Structure-from-Motion" photogrammetry: A low-cost, effective tool for geoscience applications. Geomorphology, 179, 300–314.

Wu, C. (2013). Towards linear-time incremental structure from motion. In 3D Vision-3DV 2013, 2013 International Conference on (pp. 127–134). IEEE.

Xiao, J., Ehinger, K. A., Hays, J., Torralba, A., & Oliva, A. (2014). SUN Database: Exploring a Large Collection of Scene Categories. International Journal of Computer Vision.

Yamaguchi, T., & Hashimoto, S. (2010). Fast crack detection method for large-size concrete surface images using percolation-based image processing. Machine Vision and Applications, 21(5), 797–809.

Yang, M., Kpalma, K., & Ronsin, J. (2008). A survey of shape feature extraction techniques. Pattern Recognition, 43–90.

Yeum, C. M., & Dyke, S. J. (2015). Vision-Based Automated Crack Detection for Bridge Inspection. Computer-Aided Civil and Infrastructure Engineering, 30(10), 759–770.

Yeum, C. M., Dyke, S. J., Ramirez, J. A., Hacker, T., Pujol., S., & Sim, C. (2017). Annotation of Image Data from Disaster Reconnaissance. Proceedings of the 16th World Conference on Earthquake Engineering, Santiago, Chile, Jan 2017.

Yeum, C. M., Dyke, S. J., Ramirez, J. A., & Benes, B. (2016). Big Visual Data Analysis for Damage Evaluation in Civil Engineering. Proceedings of International Conference on Smart Infrastructure and Construction, Cambridge, U.K. June 2016.

Zhang, W., Zhang, Z., Qi, D., & Liu, Y. (2014). Automatic Crack Detection and Classification Method for Subway Tunnel Safety Monitoring. Sensors, 14(10), 19307–19328.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., & Oliva, A. (2014). Learning deep features for scene recognition using places database. In Advances in neural information processing systems (pp. 487–495).

Zhu, Z., German, S., & Brilakis, I. (2011). Visual retrieval of concrete crack properties for automated post-earthquake structural safety evaluation. Automation in Construction, 20(7), 874–883.

Zitnick, C. L., & Dollár, P. (2014). Edge boxes: Locating object proposals from edges. In Computer Vision–ECCV 2014 (pp. 391–405). Springer.

Zou, Q., Cao, Y., Li, Q., Mao, Q., & Wang, S. (2012). CrackTree: Automatic Crack Detection from Pavement Images. Pattern Recogn. Lett., 33(3), 227–238.