DEVELOPING ARTIFICIAL INTELLIGENCE-BASED DECISION SUPPORT FOR RESILIENT SOCIO-TECHNICAL SYSTEMS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Ali Lenjani

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

Aug 2020

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL STATEMENT OF DISSERTATION APPROVAL

Dr. Shirley J. Dyke, Co-chair

School of Mechanical Engineering

Dr. Ilias Bilionis, Co-chair

School of Mechanical Engineering

Dr. Julio A. Ramirez

Lyle School of Civil Engineering

Dr. Karen Marais

School of Aeronautics and Astronautics Engineering

Dr. Antonio Bobet

Lyle School of Civil Engineering

Approved by:

Dr. Nicole Key

Head of the School Graduate Program

ACKNOWLEDGMENTS

First, I want to express my sincere gratitude to my advisors Dr. Shirley J. Dyke and Dr. Ilias Bilionis, for their continuous support of my Ph.D. study. I will always be grateful for their patience, encouragement, and tireless efforts to improve my research and skills. I feel both lucky and smart to have two of the most outstanding mentors and scholars as my Ph.D. advisors.

Next, I thank my committee members, Dr. Julio A. Ramirez, Dr. Antonio Bobet, and Dr. Karen Marais, for sharing their valuable time with me. I would like to thank my fellow graduate students and postdoctoral researchers at the Intelligent Infrastructure Systems Laboratory, the Predictive Science Laboratory, and Resilient Extraterrestrial Habitat Institute, especially Dr. Amin Maghareh, Dr. Chul Min Yeum, Dr. Francisco Pena and Dr. Jongseong Choi.

I wish to acknowledge the data contributions from Dr. Ricardo Monteiro at EUCentre (Pavia, Italy), Thomas Smith at TLSmith Consulting Inc., Google Street View Service, FEMA, NOAA, and Steer Network.

Finally, I acknowledge the financial support from multiple sources, including the NSF under Grants No. 1608762, and 1835473, CRISP at Purdue University, and NASA under award No. 80NSSC19K1076.

TABLE OF CONTENTS

			I	Page
LIS	ST OF	TABLE	s	vi
LIS	ST OF	FIGURI	ES	vii
ΑF	BSTR <i>A</i>	ACT		X
1	INTE	RODUC	ΓΙΟΝ	1
	1.1	Illustra	ative application	4
2			CE-BASED METHOD FOR PRIORITIZING POST-EVENT BUILD-TION	10
	2.1	Metho	dology	11
		2.1.1	Post-event inspection as a decision-making problem	11
		2.1.2	Quantifying the conditional probability of a building's safety state given the event intensity	13
		2.1.3	Predicted safety level optimization	15
		2.1.4	Pareto front	15
	2.2	Illustra	ative Example	16
		2.2.1	The data set	16
		2.2.2	Cost function to quantify losses due to incorrect classification	21
		2.2.3	Discussion on consequences of communities risk-oriented decisions	s 23
3			ED BUILDING IMAGE EXTRACTION FROM 360-DEGREE PANORADST-DISASTER EVALUATION	30
	3.1	Proble	m Statement	31
	3.2	Metho	dology	35
	3.3	Experi	mental Validation	52
		3.3.1	Description of the test site	52
		3.3.2	Construction of residential building image database	53
		3.3.3	Training and testing residential building detector	57

				Page
		3.3.4	Sample implementation	. 61
4			AND POST-EVENT DATA COLLECTION AND ANALYSIS USING INTELLIGENCE	. 65
	4.1	Techni	cal approach	. 66
		4.1.1	Design of the classification schema	. 69
		4.1.2	Information fusion	. 76
	4.2	Experi	mental validation	. 83
		4.2.1	Post-event stream validation	. 87
		4.2.2	Pre-event stream validation	. 95
5			NG DATA-DRIVEN PROBABILISTIC VULNERABILITY MODEL GE	
	5.1	Metho	dology	. 100
		5.1.1	Automated vulnerability model generating	. 101
		5.1.2	Regional loss prediction	. 102
	5.2	Illustra	ative Example	. 103
6	SUM	MARY A	AND CONCLUSION	. 110
RE	FFRE	NCFS		11/

LIST OF TABLES

Table		P	age
2.1	Categories used for the building inventory to represent floor area		17
2.2	Cost function for residential buildings (in $\$$ per floor area category)		22
2.3	Cost function for commercial buildings (in $\$ per floor area category). $\ \ldots$.		22
2.4	Cost function for critical facilities (in $\$$ per floor area category)		22
3.1	Parameters and definitions		36
4.1	Hyper-parameters used to train the classifiers		83
4.2	Loss function		87
4.3	Confusion matrix using a loss function with parameters ($\alpha_1 = \alpha_2 = 0.3$); assuming all buildings are sufficiently covered		88
4.4	Confusion matrix using a loss function with parameters ($\alpha_1 = \alpha_2 = 0.3$); considering a sample coverage probability, $q_1 = 0.2, q_2 = 0.5, q_3 = 0.9, q_n = 1$ for $n \geq 4$		92
4.5	Loss function		96
4.6	Confusion matrix of first floor elevation using a loss function with parameters $(\alpha_1 = \alpha_2 = 0.3)$		96
4.7	Confusion matrix of number of stories using a loss function with parameters $(\alpha_1 = \alpha_2 = 0.3)$		96
4.8	Confusion matrix of construction material using a loss function with parameters ($\alpha_1 = \alpha_2 = 0.3$)		96

LIST OF FIGURES

Figu	re	Pa	ıge
1.1	Schematic of system dynamics		2
1.2	General diagram showing the required steps for resilience evaluation of a socio-technical system		3
1.3	Relationship between general framework and illustrative application		4
1.4	Diagram showing the four modules to accomplish the socio-technical systems resilience evaluation		7
2.1	Building inventory taxonomy.		17
2.2	Floor area category distribution of the building inventory		18
2.3	Example showing (a) the fragility functions for a 2 story RC residential building, and (b) the corresponding probability functions		20
2.4	Sample communities		24
2.5	Statistics of the results for three events with different intensities for the different communities		25
2.6	Pareto front		27
2.7	Cost distribution and Pareto front for regions with different hourly inspection rates		27
2.8	Non-exceedance probability of the cost for each community for regions with different hourly inspection rates		28
3.1	Sample images collected during post-event building reconnaissance missions (Courtesy of Timothy P. Marshall and Thomas P. Smith, respectively)		33
3.2	Rectilinear images created from the same panorama		33
3.3	Overview of the technique developed		34
3.4	Geometry of a panorama and rectilinear image.		40

Figu	re I	age
3.5	Test site for experimental validation, Holiday Beach	52
3.6	Sample panorama and the corresponding optimal rectilinear image for a target house	53
3.7	Sample building images used for training a residential building (house) classified	r. 53
3.8	The performance of the residential building detector for the testing set	55
3.9	Samples of detected residential buildings using Faster R-CNN algorithm	56
3.10	Post-disaster building reconnaissance images of two different houses after Hurricane Harvey.	59
3.11	Samples of the pre-disaster house images generated from Google Street View: (a) and (b) shows the same houses in Fig. 10 (a) and (b), respectively, before Hurricane Harvey and (c), shows a house in the same neighborhood with two others but its post-disaster geo-tagged image cannot be accessed	60
3.12	Reason to observe the house from multiple viewpoints: Optimal rectilinear images with the bounding boxes of the detected house from multiple views. Since the views of the house are partially or entirely obstructed by the foreground objects (here, trees), many images from different viewpoints should be considered to observe the entire front and side views of the house	62
4.1	Diagram showing the steps in the automated procedure	67
4.2	Hierarchy of classifiers used in pre-event and post-event data analysis streams.	70
4.3	Detailed steps in the post-event data analysis stream	70
4.4	Samples of images classified as overview (OV) and non-overview (NOV)	71
4.5	Samples of images classified as major damage (MD) and non-major damage (NMD)	72
4.6	Detailed steps in the pre-event data analysis stream	74
4.7	Samples of images classified as Elevated and Non-elevated building images	77
4.8	Samples of images classified as (a) One-story and (b) Two-stories	77
4.9	Samples of images classified as (a) Wood and (b) Masonry	78
4.10	Post-event reconnaissance dataset collected after Harvey and Irma and published on DesignSafe-CI and Fulcrum [89, 107, 108]	85
4.11	Accuracy plots of classifiers	86
4.12	Density of the fusion predictive probabilities corresponding to each different decision, assuming all buildings are sufficiently covered (with $q_n = 1$ for $n \ge 1$)	88

Figu	re Page
4.13	Sample of a correct MD detection
4.14	Sample of a ND building categorization
4.15	Sample of an incorrect building categorization
4.16	Fusion probabilities
4.17	Accuracy vs ND rate
5.1	Schematic of the process developed to generate empirical vulnerability models 100
5.2	Predicting regional loss of a future event with a regional inventory and potential range of event intensity
5.3	Taxonomy of the raw data collected after hurricane Harvey [89, 101, 107] 104
5.4	Assigning the approximate wind speed to the building inventory using the hurricane Harvey wind map [116]
5.5	Damage states for residential construction classes [117]
5.6	Visualization of the 156,099 buildings visited by FEMA after hurricane Harvey [118]
5.7	Validation and prediction results
6.1	Overview of the approach

ABSTRACT

Lenjani, Ali Ph.D., Purdue University, Aug 2020. Developing Artificial Intelligence-Based Decision Support for Resilient Socio-Technical Systems. Major Professors: Shirley J. Dyke, Ilias Bilionis.

During 2017 and 2018, two of the costliest years on record regarding natural disasters, the U.S. experienced 30 events with total losses of \$400 billion. These exuberant costs arise primarily from the lack of adequate planning spanning the breadth from pre-event preparedness to post-event response. It is imperative to start thinking about ways to make our built environment more resilient. However, empirically-calibrated and structure-specific vulnerability models, a critical input required to formulate decision-making problems, are not currently available. Here, the research objective is to improve the resilience of the built environment through an automated vision-based system that generates actionable information in the form of probabilistic pre-event prediction and post-event assessment of damage. The central hypothesis is that pre-event, e.g., street view images, along with the post-event image database, contain sufficient information to construct pre-event probabilistic vulnerability models for assets in the built environment. The rationale for this research stems from the fact that probabilistic damage prediction is the most critical input for formulating the decision-making problems under uncertainty targeting the mitigation, preparedness, response, and recovery efforts. The following tasks are completed towards the goal. First, planning for one of the bottleneck processes of the post-event recovery is formulated as a decision making problem considering the consequences imposed on the community (module 1). Second, a technique is developed to automate the process of extracting multiple street-view images of a given built asset, thereby creating a dataset that illustrates its pre-event state (module 2). Third, a system is developed that automatically characterizes the pre-event state of the built asset and quantifies the probability that it is damaged by fusing information from deep neural network (DNN) classifiers acting on pre-event and post-event images (module 3). To complete the work, a methodology is developed to enable associating each asset of the built environment with a structural probabilistic vulnerability model by correlating the pre-event structure characterization to the post-event damage state (module 4). The method is demonstrated and validated using field data collected from recent hurricanes within the US. The vision of this research is to enable the automatic extraction of information about exposure and risk to enable smarter and more resilient communities around the world.

1. INTRODUCTION

In the literature, *resilience* refers to the ability of a system to overcome disruptions and return to a normal state while minimizing loss [1–3]. The loss includes casualties, damage, socio-economic, and ecological impacts associated with the *disruption*. Here *disruption* can refer to any type of external hazard-event, internal malfunctioning or deterioration of systems elements. A significant amount of effort has been devoted to understanding what characteristics make a system resilient [4–9]. Since the concept of resilience can be applied to almost any type of system, it has been studied from many perspectives. The focus of much of the recent research ranges from assessing the risk in the system, studying the impact of an event on the system, or analyzing the recovery of the system over time [10–12].

A common approach to evaluate the resilience of a system includes four criteria, specifically robustness, rapidity, redundancy, and resourcefulness [13, 14]. However, the required methods to evaluate these criteria in a system are not clear. On the other hand, there is no guarantee that satisfying these criteria will improve the utility of the system.

Here, a general framework is proposed which includes the required steps to *assess* the resilience of a system. In general this framework breaks down into three majors phases of: 1) system state estimation, 2) short-term decision making, and 3) long-term decision making.

To evaluate the influence of possible strategic decisions on the resilience of the system, it is required to evaluate the overall system performance over the specified time period $\{0, \ldots, T\}$. The overall system performance can be evaluated through assessing the system state trajectory at each time step t. The system state at time step t depends on the systems state at time t-1, disruptions to the system, and short-term decision made at time step t-1. Short-term decisions at time t should be made based on the system state at time t-1, and availability of the resources. To assess the overall system performance over

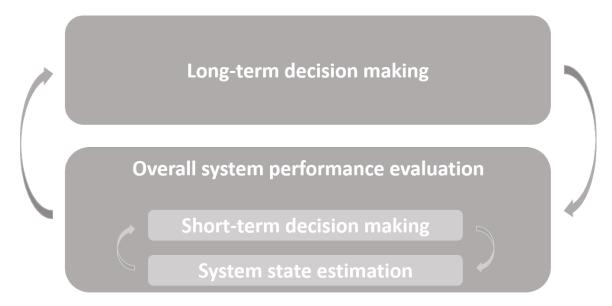


Fig. 1.1.: Schematic of system dynamics

time period $\{0, ..., T\}$, it is assumed that short-term decisions at time t are optimal. To develop such decision support to make the optimal operational decisions, predicting the future states of the system plays an important role. The bi-directional relationship of the three major phases are illustrated in Fig. 1.1.

Here are the detail steps required for implementation of the three major phases. Figure 1.2 shows a schematic of the proposed framework, including the associated ten potential steps required for the resilience evaluation of a system:

- 1. Modeling the disruptions in the system
- 2. Gathering required pre-disruption data of the system
- 3. Characterizing the system
- 4. Developing element-specific probabilistic vulnerability models
- 5. Modeling the interdependencies in the system
- 6. Modeling the uncertainties in the system
- 7. Estimating system state

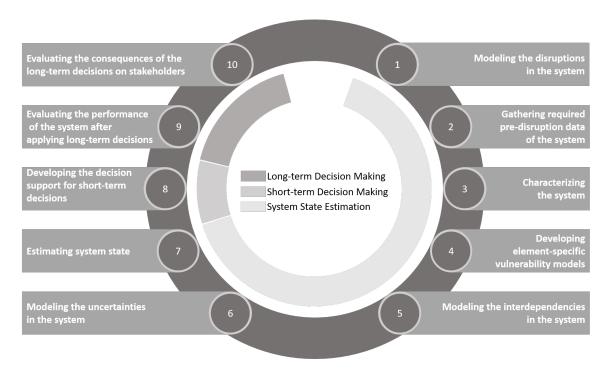


Fig. 1.2.: General diagram showing the required steps for resilience evaluation of a sociotechnical system.

- 8. Developing the decision support to make the post-disruption short-term decisions
- 9. Evaluating the performance of the system after applying long-term decisions (resources allocation, design)
- 10. Evaluating the consequences of the long-term decisions (resources allocation, design) on stakeholders

For some applications it is possible that certain steps shown in the framework are not applicable to a particular system, e.g., if inter-dependency between elements is negligible, step 5 is not applicable. On the other hand, it is also possible that some of the steps may already be available, e.g., if a model of the system is already available, information about steps 2 and 3 is already available.

The proposed framework is designed to apply to a wide range of socio-technical systems. To demonstrate the capabilities an illustrative application is used with particular characteristics and goals. The illustrative application focuses on automated techniques

General Framework	Application
Modeling the disruptions in the system	Available in literature
Gathering required data of the system	Module 1
Characterizing the system	Module 2
Developing element-specific vulnerability models	Module 3
Modeling the interdependencies in the system	N/A
Modeling the uncertainties in the system	Part of Module 4
Estimating system state	Part of Module 4
Developing the decision support for operational decisions	Part of Module 4
Evaluating the performance of the system after applying strategic decisions	Part of Module 4
Evaluating the consequences of the strategic decisions on stakeholders	Part of Module 4

Fig. 1.3.: Relationship between general framework and illustrative application.

to evaluate the effect of post-event inspection resource allocations on communities. Four research problems designed to address this module are introduced in Sec. 1.1. The Four accomplished research are discussed in detail in Secs. 2, 4, 3, and 5. Fig. 1.3 shows the linkage between general framework and illustrative application.

1.1 Illustrative application

As millions of people across the east coast, the Gulf of Mexico, Puerto Rico and other Caribbean Islands expecting severe hurricanes to make landfall each year and cause damage in their infrastructure, it is critical to ask how these communities could be made more resilient. During only 2017 and 2018, two of the costliest years on record regarding natural disasters, the U.S. experienced 30 events with total losses of \$400 billion. These exuberant costs arise primarily from the lack of adequate planning spanning the breadth from prevent preparedness to post-event recovery. Understanding the nature of the risk in our complex infrastructure before such a natural disaster will enable decisions to be made at a national, community, or individual level to reduce risk and mitigate losses.

There is a strong consensus that an essential element of resilience is the preparation for and conduct of rapid and efficient assessment of the post-event situation, e.g., see [15–17]. In the case of the built environment, post-event building assessment takes the form of expert inspection which is scheduled after the event. The classification of a structure's safety level is necessary both for preventing further loss of life and for planning recovery actions to return the community to normalcy. The approach used in ATC-20 is to classify structures as inspected (no considerable damage and residents are allowed to occupy the building), restricted use (partially damaged but some parts of the building are usable), or unsafe (severely damaged and all persons are restricted from entering) [18] Variations on this exist in various places in the world, and for different types of stakeholders. A major post-disaster challenge that hinders community resilience is the backlog created from the sheer volume of buildings needing inspection [19]. For example, the average wait time for a field inspection after Hurricane Harvey was 45 days, and after Hurricane Irma it was about a month [20]. As insurance payments are curbed and rebuilding is stalled, the recovery of the affected community is halted, and victims remain in limbo. There is an urgent need to make the post-event inspection process of the built environment more informative and more efficient.

To support decision-making during *routine* inspection procedures for infrastructure systems, several methods have been developed to strategically mitigate risks [21–24]. For instance, regulations in place in many nations around the world require that bridges over a certain size be inspected at least every 24 months, with certain local variations in the details [25]. However, in the immediate aftermath of a disruptive hazardous event, planning the post-event inspection of the infrastructure systems poses a different type of challenge [26]. Restrictions in the time and resources available and wide-ranging scope of the inspections needed require that strategic decisions be made quickly. [27] developed a handbook for the inspection of the Indiana bridge network, focusing on how to evaluate the condition of each type of bridge. The prioritization of those inspections was left to the state agency. [28] used a Bayesian network and influence diagrams to analyze the performance of a centrally-managed infrastructure system (here, a bridge network) and

its components after an extreme event. Using a network-level approach, they considered whether or not to inspect each bridge component, or to implement mitigation actions, e.g., reduce operation or completely shut down the component to avoid further losses. [28] also introduced the concept of the value of information (VoI) pertaining to assessment in spatially distributed infrastructures to determine a temporal ordering of the inspection of the structure's components based on the output of the influence diagram. VoI is used to quantify the benefit of gathering additional information before taking action, e.g., a decision to shut down a component or keep it in operation. Among the possible inspection alternatives, the approach taken here is that the highest-priority alternative is the one that derives the largest benefit from an inspection.

Indeed, the method discussed above is powerful in terms of integrating various types and levels of information in restoration decisions. However, implementation of this method does require access to comprehensive inventories, detailed asset descriptions and spatial information. It is well-suited for dealing with networks of privately-operated (e.g., railways) or publicly-managed (e.g., bridges and dams) infrastructure under the control of a single owner that has kept detailed maintenance records. There are three main reasons why, at the present time, this framework may not be appropriate in general for communities. First, the detailed datasets needed for the implementation of the framework is not typically available in communities because of the high monetary cost associated with their maintenance. Second, this method is not intended to weigh the estimated cost of inspections, or the potential consequences (e.g., costs) of making incorrect decisions under such traumatic conditions. Finally, this method focuses only on the response in the aftermath of an observed event. Limitations arise when dealing with uncertain future events. Forward-thinking communities interested in promoting resilience should be ready to act after an event, but should also prepare for such an unforeseen event by deciding on their objectives.

To reach a target level of resilience, we need to employ rigorous decision-making approaches to prepare for and mitigate the impact of disastrous events. The decision-making approach for such a complicated task can be effective, only if it is comprehensive

and analyzes the consequences from various perspectives. A comprehensive analysis of disruptive events management needs a clear understanding of processes involved in that event, and of the challenges these processes pose for administrators and communities [29,30]. To support more communities to cope with realistic large-scale disruptive events, a simple and flexible approach is needed that can be applied to communities with varying levels of data inventory.

Here, the research objective is to develop an end-to-end automated decision support methodology to improve resilience of the built environment. This research is focused on providing a community with the means to plan and allocate resources in advance of an event to achieve the desirable resiliency goals of the community. In this study, a methodology is developed intended to one of the strategic post-event decisions to estimate the amount of resources for post-event inspections which pose the minimum consequences on the administrators and community. The required actionable information to enable making a risk-informed decision would be generated through an automated vision-based system in the form of probabilistic pre-event prediction and post-event assessment of damage. This method requires the buildings physical characteristics and vulnerability models of the built environment, and the proper decision making algorithm to find the post-event inspection priorities. For developed communities available data repositories can provide buildings physical characteristics and vulnerability models. However these information is not available for all communities. To address this need automated techniques are developed to rapidly extract the building inventory.

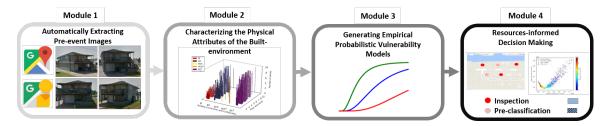


Fig. 1.4.: Diagram showing the four modules to accomplish the socio-technical systems resilience evaluation.

The research objective is accomplished through the following modules, See Fig. 1.4:

Module 1: Formulating the resources allocation of the post-event inspection

The objective of this module is to formulate the problem of allocating the resources toward the task of post-event inspection as a risk-informed decision-making problem. The working hypothesis is that a complex strategic decision of post-event inspection resource allocation can be formulated using limited information about the built environment. This method uses the building physical attributes and the corresponding vulnerability models to determine the post-event inspection priorities, and make risk-informed decisions to minimizes the consequences imposed on the community. The outcome of this module is a decision-making methodology that optimize the amount of resources for post-event inspection due to resilience objectives of the community, See chapter 2.

Module 2: Collect and preprocess images from building assets at risk

The objective of this module is to automatically extract data that characterize the assets included in a built environment before a disaster. These data will consist of GPS locations of individual assets along with a set of asset images. The working hypothesis is that these data can be extracted from existing 360° photos available through street-view service providers, e.g., Google street-view. These 360° photos include geolocation information and are suitable for capturing each building from different viewpoints by adjusting the viewing directions and field of view. Object recognition techniques and computer vision algorithms are implemented to identify, extract and construct multiple views of each asset, e.g. front and sides views. These images are needed to extract the physical attributes of the buildings that characterize the building performance during disruptive event. The outcome of this module is a methodology that automatically extracts asset geolocation and visual information from street-view image data, See chapter 3.

Module 3: Develop vision-based exposure detection

The objective of this module is to detect structural characteristics, e.g., structural type, number of stories, roof style, height of the first floor from ground, of each asset of the built environment using the dataset collected as part of module 2. The working hypothesis is this identification can be carried out using deep neural network (DNN) classifiers from

the extensive labeled datasets. Convolutional Neural Networks (CNNs) based classifiers are trained to automatically recognize these key structural characteristics in the images of each asset. The outcome of this module is algorithms that turn raw visual asset data to features that characterize the structural response, See chapter 4.

Module 4: Generate probabilistic vulnerability models

The objective of this module is to automate the process of developing empirical probabilistic vulnerability models using pre- and post-event images. The working hypothesis is that the vulnerability of the assets can be predicted based on the key structural and nonstructural characteristics of the built environment calibrated empirically to pre- and post-event information of the similar geographical regions. Using the dataset, the input-output pairs for each observed asset are generated consisting of structural characteristics before the hurricane (input) and its damage state after the hurricane (output). Using these input-output pairs a model is created that predicts the probability of a specific damage level conditional on the asset characteristics and the magnitude of the disaster (local wind speed, inundation depth, etc.). The outcome of this module is a methodology that automatically predict the vulnerability of assets, using the empirically calibrated probabilistic vulnerability models, with known structural characteristics, See chapter 5.

Chapter 2, explains the resilience-based methodology developed for prioritizing the post-event inspection, module 1. Chapter 3, describes the automated technique to extract multi-view pre-event images, module 2. The method developed to detect the characteristics of the built environment is explained in chapter 4, module 3. Chapter 5 describes the automated process of empirical vulnerability generation, module 4. Summary and conclusion are discussed in chapter 6.

2. A RESILIENCE-BASED METHOD FOR PRIORITIZING POST-EVENT BUILDING INSPECTION

One important step post-disruption planning is predicting the extent of the damage in the built environment for potential hazards. In this study, I developed a methodology to support rapid post-event decision-making about inspection priorities with limited information to enable communities to make rapid cost-based decisions related to inspection of their building inventory. The objective of this research is to provide a simple approach and an associated computational tool to support rapid decision-making related to postevent inspections. With such a capability, a community can make rapid decisions related to inspection of their building inventory, based on the likely economic cost associated with restricting access to that inventory, and given a pre-determined budget. This can be achieved through the combination of structure-specific fragility functions and cost-based decision-making. According to the scope of the decisions I can pose the decision making problem as a single or multi-agent system [31, 32]. Here, I formulate this problem mathematically by assuming that the community is a rational agent seeking to minimize the expected cost of the disruption as well as the risk of its actions. The inputs are the cost of closure of each building when it is the correct action to take, the cost of closure when it is unnecessary (i.e., when the structure is actually not unsafe and the building is mistakenly restricted), and the likelihood of damage to each structure for a given intensity event. The output of the approach is the prioritized order for inspection to most effectively allocate resources on a limited budget. With this capability a community will reduce recovery time after an event by accelerating the inspection process to restore confidence in our structures.

The advantages of this approach are that it (i) is simple, (ii) requires minimal inventory data, (iii) is easily scalable, and (iv) does not require significant computing power. In addition, it can be used either immediately after a disruptive event or in the planning stage

to set a budget to prepare for potential future disruptive events with that community's objectives in mind. This method also generates actionable information that a community can choose to implement to be prepared for future events, i.e., to become more resilient. The approach is demonstrated using a crowd-sourced dataset, collected as a part of the EU-funded project, SASPARM2.0 [33]. However, the approach can readily be adapted to consider spatially-distributed networks of other classes of infrastructure, or combinations thereof, and to support other types of decisions when resources are limited.

The remainder of this chapter is organized as follows. Sec. (2.1) presents the problem statement and formulation. Sec. (2.2) provides an illustrative example to demonstrate the methodology including results and discussion. Please see the archived [34] and published versions [3].

2.1 Methodology

2.1.1 Post-event inspection as a decision-making problem

Consider a community with n buildings with exposure to hazards. Let $i \in \{1, ..., n\}$ be the index assigned to identify each building. With the variable b_i I denote the building characteristics, e.g., number of stories, type of construction, floor area. The random variable (r.v.) $X \in [0, \infty)$ characterizes the hazard intensity. To indicate a building's safety level I use the discrete-valued r.v. S_i . The definition and number of these safety levels should be determined by relevant stakeholders, and different approaches have been taken in various regions [18, 35, 36]. Without loss of generality, I assume that S_i takes values in $\{1, 2, 3\}$ and that structural damage is more severe as S_i increases. The conditional probability of building i being at safety level $S_i = s_i$ after a hazardous event with intensity X = x is denoted by $\mathbb{P}[S_i = s_i | X = x]$, see Sec. 2.1.2.

Let $d_i \in \{1, 2, 3\}$ be the decision variable corresponding to the safety level assigned each building (1 = "safety level 1", 2 = "safety level 2", 3 = "safety level 3"). I can determine d_i in one of two ways: (1) I can perform a field inspection revealing the true state of the building, but at a fixed cost w_i , see Sec. 2.1.3; or (2) I can select d_i based on building char-

acteristics b_i and the observed hazard event intensity X = x without an inspection, i.e., using a decision function $d_i^*(x)$, see Sec. 2.1.3. The latter is an effective option when there is a high post-event probability of the building being at a given state, e.g., when the model is confident that the building is either safe or is damaged significantly (i.e., it is not safe to enter). Being wrong, however, can be costly to the community. This misprediction cost, denoted as $c_i(d_i, s_i) := c(d_i, s_i; b_i)$, is an increasing function of the discrepancy between the predicted state d_i and the true state s_i . The misprediction cost also depends on the use of the structure. To construct $d_i^*(x)$ I minimize the expected cost of a wrong safety level assignment, see Sec. 2.1.3 for the mathematical details. For this section, let $c_i^*(x)$ be the minimum expected cost of potential casualties, economic, social, and environmental losses, resulting from selecting $d_i^*(x)$. Given a fixed budget r > 0 allocated at time $t_0 = 0$, how should the community choose which buildings to inspect in case an event with intensity x occurs at time t? If the community is risk-neutral, then it should minimize the expected discounted cost of its actions, see Sec. 2.1.3. Let z_i be a binary decision variable representing whether to accept the optimal *predetermined* safety level for building *i*, $z_i = 0$, or to inspect the building, $z_i = 1$. Collectively, let $z_{1:N} = (z_1, \dots, z_N)$ be the vector representing the decisions for all buildings. The optimal decision, $z_{1:N}^*(t,x;r)$, minimizes the expected cost subject to inspection budget constraints. Mathematically, the optimal decision solves:

$$\min_{z_{1:n} \in \{0,1\}^N} \sum_{i=1}^N c_i^*(x) (1 - z_i), \tag{2.1}$$

subject to the budget constraint:

$$\sum_{i=1}^{n} e^{\gamma t} w_i z_i \le e^{\alpha t} r, \tag{2.2}$$

where $\gamma > 0$ is inflation rate, and $\alpha \ge 0$ is the return rate of the safe asset in which the community invested its budget at time $t_0 = 0$. The optimization problem by Eqs. (2.1) and (2.2) is known as a *knapsack problem* [37]. I can solve this problem through the *dynamic programming* algorithm implemented in OR-Tools Python library [38].

But how should the community set its initial inspection budget r? To answer this question, assume that the community responds to an event of intensity X occurring at a random time T by solving the above-mentioned knapsack problem. Then, the cost C to the community is the cost of inspection plus the cost to the community from making incorrect predictions (note that correct predictions do not add to the cost to the community), i.e., C is the r.v.

$$C(T, X, S_{1:N}) = e^{\gamma T} \sum_{i=1}^{N} [w_i z_i(T, X; r) + c_i(d_i^*(X), S_i)(1 - z_i^*(T, X; r))], \quad (2.3)$$

where $S_{1:N} = (S_1, ..., S_N)$. A *risk-neutral* community would seek to minimize its discounted expected cost, i.e., it would select the budget by solving:

$$\min_{r \in [0,\infty)} \mathbb{E}[e^{-\beta T} C(T, X, S_{1:N})], \tag{2.4}$$

where $\mathbb{E}[\cdot]$ denotes the expectation over all random variables, and $\beta > 0$ is the discount rate of the community. Now, a *risk-averse* community would be interested in keeping the variance of *C* under control, it would also seek to solve:

$$\min_{r \in [0,\infty)} \mathbb{V}[e^{-\beta T} C(T, X, S_{1:N})], \tag{2.5}$$

where $\mathbb{V}[\cdot]$ is the variance operator. In Sec. 2.1.4, I discuss how I derive the Pareto front of the stochastic multi-objective optimization problem defined by Eqs. (2.4) and (2.5).

2.1.2 Quantifying the conditional probability of a building's safety state given the event intensity

The conditional probabilities associated with the damage levels of the building after the hazard are defined as a set of fragility functions [39–41]. A fragility function $F_{i,l}(x)$ describes the conditional probability of the i-th building response Y_i exceeding a certain

threshold $\delta_{i,l}$ of damage level l, given the event intensity X=x, and is mathematically defined as:

$$F_{i,l}(x) = \mathbb{P}[Y > \delta_{i,l}|X = x], \text{ for } l = 0, 1.$$
 (2.6)

To predict the post-event safety state of the building I need to determine the probability of experiencing each safety state using the concept of fragility function. The first step is to associate the safety sates of the buildings with certain damage level ranges. Assuming that the building is in the "safety level 1" state when $Y_i < \delta_{i,0}$, in the "safety level 2" state when $\delta_{i,0} < Y_i < \delta_{i,1}$, and in the "safety level 3" state when $Y_i > \delta_{i,1}$, I have:

$$\mathbb{P}\left[S_{i} = \text{``safety level 1''}|X = x\right] =$$

$$\mathbb{P}\left[Y_{i} < \delta_{i,0} \middle| X = x\right] =$$

$$1 - \mathbb{P}\left[Y_{i} > \delta_{i,0} \middle| X = x\right] =$$

$$1 - F_{i,0}(x),$$

$$(2.7)$$

$$\mathbb{P}\left[S_{i} = \text{``safety level 2''}|X = x\right] =$$

$$\mathbb{P}\left[\delta_{i,0} < Y_{i} < \delta_{i,1}|X = x\right] =$$

$$\mathbb{P}\left[Y_{i} > \delta_{i,0}|X = x\right] -$$

$$\mathbb{P}\left[Y_{i} > \delta_{i,1}|X = x\right] =$$

$$F_{i,0}(x) - F_{i,1}(x),$$

$$(2.8)$$

and

$$\mathbb{P}\left[S_{i} = \text{``safety level 3''}|X = x\right] =$$

$$\mathbb{P}\left[Y_{i} > \delta_{i,1} \middle| X = x\right] =$$

$$F_{i,1}(x).$$
(2.9)

To assign the proper fragility function to the building *i*, I consider the pre-event characteristics of the building, e.g., number of stories and structural construction and configu-

ration, denoted as b_i . The pre-event characteristics of the buildings can be extracted by using automated methods, developed recently [42–44].

2.1.3 Predicted safety level optimization

Predicting the safety level of the buildings based on their pre-event characteristics is subject to errors. To minimize the adverse consequences of these decisions, first I need to quantify the imposed cost of each decision on the community. Specifically, let $c_i(d_i, s_i)$ be the cost (in dollars) imposed on the community by selecting the predicted safety level d_i when the actual building safety state is s_i . This cost represents a monetary expression of the potential casualties, the economic, social, or environmental loss, and it encodes the goals of the community. I assume that the cost grows with the inflation rate γ . I determine the optimal predicted safety level, $d_i^*(x)$, for building i, by minimizing the expected cost of this decision, i.e.,

$$d_i^*(x) = \arg\min_{d_i} \mathbb{E} \left[c_i(d_i, S_i) | X = x \right].$$
 (2.10)

The optimal expected cost is simply:

$$c_i^*(x) = \mathbb{E}\left[c_i(d_i^*(x), S_i)|X = x\right]$$
 (2.11)

2.1.4 Pareto front

To derive the Pareto front of the problem defined by Eqs. (2.4) and (2.5), I need to quantify the expected cost and the variance of the cost for all budget levels. First, I generate a set of budget levels, $r_1 < r_2 < \cdots < r_k$, where $r_1 = 0$, and r_K is the budget level required to inspect all buildings. Then, for each $k = 1, \ldots, K$, I sample M events, $\left\{\left(t^{(m)}, x^{(m)}, s_{1:N}^{(m)}\right)\right\}_{m=1}^{M}$

using the probability distributions of the occurrence time, T, the event intensity X, and the state of the building S_i conditioned on X. Then, I approximate the expected cost by:

$$\mathbb{E}\left[e^{-\beta T}C(T, X, S_{1:N})|r = r_k\right] \approx \bar{C}(r_k) := \frac{1}{M} \sum_{m=1}^{M} e^{-\beta t^{(m)}} C\left(t^{(m)}, x^{(m)}, s_{1:N}^{(m)}; r = r_k\right), \quad (2.12)$$

and the variance by:

$$\mathbb{V}\left[e^{-\beta T}C(T,X,S_{1:N})|r=r_{k}\right] \approx \sigma_{C}^{2}(r_{k}) := \frac{1}{M-1}\sum_{m=1}^{M}\left\{e^{-\beta t^{(m)}}C\left(t^{(m)},x^{(m)},s_{1:N}^{(m)};r=r_{k}\right)-\bar{C}(r_{k})\right\}^{2}, \tag{2.13}$$

respectively. After calculating the expected cost and variance of the cost for each possible budget level, I plot the Pareto frontier to visualize the budgets that are not dominated.

2.2 Illustrative Example

2.2.1 The data set

To demonstrate the approach and illustrate the information it can supply, I use a crowd-sourced dataset moderated by a group of researchers in the European Centre for Training and Research in Earthquake Engineering (EUCENTRE) in Pavia, Italy. This dataset was collected within the EU Consortium project, SASPARM2.0, to demonstrate a crowdsourcing based framework to facilitate the completion or creation of an exposure model and its corresponding physical vulnerability model. Citizens, practitioners, and students filled out specific forms developed for this project, which focused on documenting the structural characteristics of 581 buildings in the city of Nablus, a commercial and cultural center located in the northern West Bank that is adjacent to the seismically active Dead Sea Transform and associated geological faults. The dataset includes typological and metric data for the structures, e.g., building construction and configuration, number of stories, floor area, and associated fragility functions. The fragility function sets

were generated, based on SP-BELA procedures, to be appropriate for the structures in the dataset [45].

Description of the building inventory

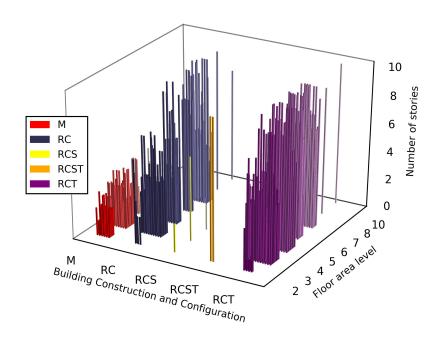


Fig. 2.1.: Building inventory taxonomy.

Table 2.1.: Categories used for the building inventory to represent floor area.

Floor area (m ²)	50	51-70	71-100	101-130	131-170	171-230	231-300	301-400	401-500	501-650	651-900	901-1200	1201-1600	1601-2200	2201-3000	>3001
Original Category	A	В	С	D	Е	F	G	Н	I	L	M	N	О	P	Q	R
Redefined																
Category		1		2	2	3	3	4	5	6	7	8	9	10	11	12

Fig. (2.1) shows a statistical summary of the key characteristics of the building inventory, including the type of construction, the number of stories, and the floor area category.

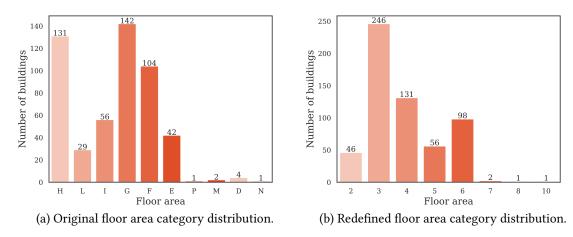


Fig. 2.2.: Floor area category distribution of the building inventory.

Each building is assigned a designation in terms of its construction and structural configuration as: masonry (M), reinforced concrete (RC), reinforced concrete shear wall (RCS). In the case of geometric irregularities that would result in torsional behavior the letter "T" is added to its designation, e.g., RCT or RCST. The fragility functions of these irregular buildings are also updated with respect to their regular counterparts using a simplified approach that makes use of correction coefficients [33]. The vast majority of the buildings contained in this inventory are M, RC, and RCT, and there are only three buildings designated as RCS and two designated as RCST [45]. Each building in the inventory is also assigned to a category based on its floor area. The original building inventory uses alphabetic letters to represent these categories (see Table (2.1)). However, I redefine these categories according to the assumed cost of field inspection grouped by area. Fig. (2.2) shows the distribution of buildings using both the original and redefined categories. The actual use of each of the buildings in the inventory is not documented. However, for purposes of demonstrating the method, I assign each building into one of three usage categories. In particular, 565 buildings are classified as residential, 12 as commercial, and 4 as critical facilities (e.g., hospitals, police/fire stations). To implement and demonstrate the methodology, I need to identify: (i) the type of construction and the number of stories to properly assign a representative fragility function to each building; (ii) the floor area category, which is used to estimate the field inspection cost; and (iii) the designated use of each building. These data are used to quantify the cost imposed on the community.

Fragility functions of the data set

Fragility functions are assigned to each building to estimate its most probable state after the event. An original set of fragility functions for this particular building inventory was developed by [45] based on the available data, including construction, geometric information (e.g., floor area, number of stories) and the structural configuration (e.g., regular, irregular) of each of the buildings. Observed damage data were not available for Palestine, and thus [45] used results obtained for Italian buildings with similar construction. The set of fragility functions was generated using a simplified pushover-based earthquake loss assessment (SP-BELA). SP-BELA was initially developed as a means to rapidly assess the vulnerability of Italian buildings. SP-BELA procedures were specified for masonry buildings, RC frame buildings, and precast concrete buildings [46–48]. Originally SP-BELA featured three limit states: light damage (LS1), significant damage (LS2), and collapse (LS3). However, [49] adapted the set of fragility functions developed based on this data set to correspond to the EMS98 scale [49]. This scale involves five damage levels, i.e., slight damage (D1), moderate damage (D2), extensive damage (D3), complete damage (D4), and collapse (D5). The relationship between damage level and limit state was defined using observed damage data in a series of recent Italian earthquakes beginning in 1976 with the Friuli event through to 2002 with the Emilia event [50].

According to the EMS98 scale, D4 and D5 refer to building states defined as completely damaged and collapsed, respectively, and thus are clearly well beyond a state in which they can be considered usable. Inspection to distinguish between these two states is not necessary. Thus, based on the definitions in EMS98 and for purposes of illustration, I pair the D1 fragility function with a safety level 1, D2 with the safety level 2 state, and D3 and above with safety level 3. Also, the original building inventory used to determine the set of fragility functions consists of buildings that are not seismically designed. Our simulations

show that, with even a small intensity event, there is a high probability of all buildings in the inventory reaching safety level 3. Thus, to consider an inventory that is more representative of a typical community with modern construction and designed according to seismic building codes, I modify the set of fragility functions by multiplying both μ and σ by a selected coefficient. The coefficient is selected based on judgment as 2.5, 3.0 and 3.5 for residential buildings, commercial buildings and critical facilities, respectively, to better represent reasonable performance levels for seismically designed buildings.

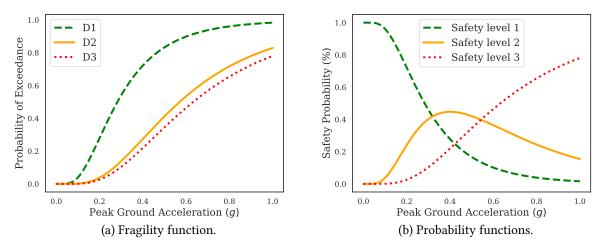


Fig. 2.3.: Example showing (a) the fragility functions for a 2 story RC residential building, and (b) the corresponding probability functions.

To apply the method developed herein, I need to interpret the fragility functions for a given building to represent the probability of the occurrence of each safety level. To accomplish this, I first select the fragility function that corresponds to a given safety level and then transform that fragility function, which is defined as the probability of exceeding a particular limit state, to a curve that corresponds to the probability of the post-event building state being associated with each safety level. To demonstrate these steps, consider the fragility functions of a 2 story RC residential building. Fig. (2.3a) shows the fragility functions for this building corresponding to damage states D1 (green dashed), D2 (yellow solid), and D3 (red dotted). Fig. (2.3b) shows the corresponding probability

functions, which represent the probability of that building being in the corresponding state (in this case, the associated safety level) after the event, as explained in Sec. (2.1.2).

2.2.2 Cost function to quantify losses due to incorrect classification

I define an intuitive cost function to compare the consequences of the decisions. The cost function has two terms, corresponding to: (i) the estimated cost of building inspections to the community, (ii) the estimated cost associated with making an incorrect decision regarding the state of a building. The values in the cost function are approximated and for a specific community, and they should be adjusted to represent the actual costs for that target community. The cost of a field inspection for a building is determined based on the size of the building, and is constant within a given category of building. Thus, it is the product of the number of stories, the average floor area of the building based on its category (as defined in Table (2.1)), and the cost per unit area for a field inspection. The inspection cost increases with the floor area category, and is selected as \$500 for the first category $(0-100 m^2)$ and increases by \$500 for each subsequent floor area category. This value represents the actual monetary cost for a structural engineer to do a field inspection [51,52]. Due to the high-demand for qualified structural engineer's time in emergency conditions, I adjust this field inspection rate. Here I magnify this value by a factor of 10. The second term in the cost function is associated with the incorrect assignment of predetermined safety states, and represents the cost that a wrong decision will impose on the community. If every building is inspected by a qualified engineer after the event, the resulting cost to the community will be a fixed amount. However, if, for instance, the *predetermined* safety level and the actual state of every building match, the resulting additional cost to the community would be zero. If any single building is under-rated (i.e., the predetermined safety level is lower than the actual safety level), that incorrect decision introduces a degree of risk associated with the error (i.e., allowing residents to enter a safety level 3 building), and this may result in casualties and thus may impose a tremendous additional cost to the community [53]. On the other hand, if the

building is over-rated, the incorrect decision is conservative (i.e., unnecessarily restricting access to a building that is functional), and this may impose a considerable additional cost to the community in the form of lost revenue for commercial buildings, hotel costs for occupants of residential buildings, or a gap in critical services (e.g., hospital services, police and fire services). These costs may not, however, be as massive as in the previous case.

Table 2.2.: Cost function for residential buildings (in \$ per floor area category).

		Decision								
e		safety level 1	safety level 2	safety level 3						
State	safety level 1	0	350,000	750,000						
al S	safety level 2	3,750,000	0	500,000						
tu	safety level 3	7,250,000	3,600,000	0						
Ā										

Table 2.3.: Cost function for commercial buildings (in \$ per floor area category).

		Decision								
е		safety level 1	safety level 2	safety level 3						
State	safety level 1	0	3,000,000	5,500,000						
	safety level 2	89,000,000	0	4,000,000						
ctual	safety level 3	14,500,000	7,200,000	0						
Ac										

Table 2.4.: Cost function for critical facilities (in \$ per floor area category).

		Decision							
e		safety level 1	safety level 2	safety level 3					
tate	safety level 1	0	7,750,000	15,000,000					
al S	safety level 2	25,750,000	0	10,000,000					
ctual	safety level 3	36,250,000	18,000,000	0					
Ac									

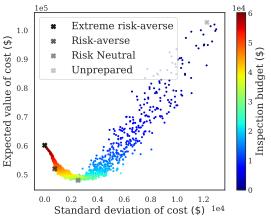
The cost function for the three types of buildings includes the same terms, but the contributing costs and their weightings are different. For this example, the added cost of under-rating in the case of commercial buildings and critical facilities are set to be

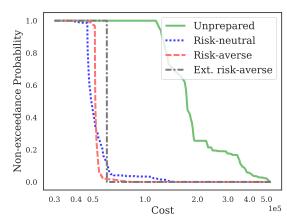
two times and five times that of residential buildings, respectively. The added cost of over-rating in the case of commercial buildings is much larger due to both the short-term and long-term effects on the community's economy. If a commercial building is over-rated as "safety level 3", any businesses in the building will be closed until an inspection can be performed. For buildings containing critical facilities, over-rating has a severe impact in terms of the resulting gap in critical services available to the community. Here I include an importance coefficient associated with each building type to represent the relative costs. The importance coefficients are 1, 2, and 5 for residential, commercial, and critical buildings, respectively. Tables (2.2), (2.3) and (2.4) provides the relative additional costs associated with *predetermined* safety levels used in this case study.

2.2.3 Discussion on consequences of communities risk-oriented decisions

To demonstrate the method, I consider four communities with different attitudes toward risk. The four communities are described as: unprepared, risk-neutral, risk-averse, and extremely risk-averse. Here the term risk refers to the risk originating from erroneous decisions in pre-classifying the post-event safety level. Thus, at the one extreme, I assume that the extremely risk-averse community will prefer to inspect all buildings in the community to eliminate any uncertainty due to this source of risk. Alternatively, at the other extreme, the unprepared community does not allocate a budget for inspection, and must rely on the *predetermined* safety levels assigned to all buildings, which is the typical output of many recent past projects that targeted the development of urban or regional risk models to assist decision-making. The risk-neutral community simply aims to minimize the expected present value of the total cost imposed on the community. The risk-averse community prefers to allocate a reasonable inspection budget, which supports minimizing both the risk and the expected total cost imposed on the community.

The results for these sample communities are shown in Fig. (2.4a) which shows the mean vs. the standard deviation of the cost imposed on each of the communities for a range of budgets between zero and the maximum budget required to perform a field in-





(a) mean vs. the standard deviation of the cost. (b) Non-exceedance probability of the cost for each community.

Fig. 2.4.: Sample communities.

spection on all buildings. In this case, this value is \$60,120,000. To obtain these results, I sample the event intensity from a lognormal distribution with mean and standard deviation of -0.8 and 0.3, respectively, i.e., $\ln X \sim \mathcal{N}(-0.8, 0.09)$ and the next occurrence time of the event from an exponential distribution with a rate of 300, i.e., $T \sim \mathcal{E}(1/300)$. For each budget level I run 1000 simulations and I calculate the actual total cost for each simulation. Using these 1000 simulations I can estimate the expected cost and risk as described in Sec. (2.1.4). I assume the interest rate, the discount rate and the inflation rate are equal in this case, $\alpha = \gamma = \beta = 0.03$. I used *dynamic programming* algorithm to solve the *knapsack problem* [38]. In case of a shortage of memory, e.g., for larger-scale analysis, I can use a meta-heuristic algorithm to solve this optimization problem [54–56].

Based on the results shown in Fig. (2.4a), it is clear that changing the allocated inspection budget can have a dramatic effect on the expected cost to the community. Additionally, the allocated inspection budget will also affect the actual level of risk in the community, or the volatility, which is captured in the standard deviation. The two noteworthy budget levels mentioned in Section 2 are noted in the figure including: (i) the risk-neutral community having the budget level which causes the lowest Sharpe ratio

(the minimum risk-adjusted cost); and (ii) extremely risk-averse community having the budget level which causes the lowest volatility (minimum standard deviation).

Next, I consider the distribution of the resulting cost for each of the defined communities based on a certain pre-determined inspection budget. To examine this distribution, I simulate the event by sampling the event parameters from intensity level and occurrence time distributions, and sampling the pre-classifying decisions from the safety level distributions. I perform 1,000 random simulations, and for each simulation I compute the total cost imposed on the community. Fig. (2.4b) shows the resulting non-exceedance probability of the total cost for each sample community.

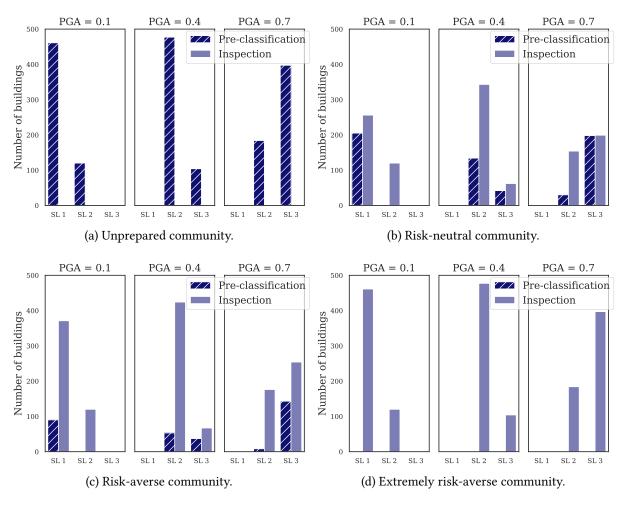


Fig. 2.5.: Statistics of the results for three events with different intensities for the different communities.

Figs. (2.5a), (2.5b), (2.5c), and (2.5d) show the corresponding results for the unprepared, risk-neutral, risk-averse, and extremely risk-averse communities, respectively. Here I assume an event happened right now, and I consider three different fixed intensity levels for the event, 0.1, 0.4 and 0.7. For each intensity level, I demonstrate the distribution of the optimal predetermined safety states of the buildings. Furthermore, for each predetermined safety state, I show the distribution of the buildings that are selected to be either inspected or pre-classified. Regardless of the intensity of the event, the extremely risk-averse community allocates the full inspection budget and performs a structural inspection on each building. Based on the building inventory used in this example, and the inspection costs assumed, \$60, 120,000 is necessary to inspect all buildings in the community no matter what event occurs. Fig. (2.5d) provides the statistics of the results for the extremely risk-averse community. The distribution of the minimum cost predetermined safety levels assignment for each intensity level in the case of the unprepared community is the same as that of the extremely risk-averse community. However, the resulting decisions regarding performing an inspection or using the *predetermined* safety levels for each of the buildings are entirely different. Because the budget allocated for inspection in the unprepared community is zero, all buildings must use their *predetermined* safety levels, as shown in Fig. (2.5a). It is likely that the extremely risk-averse approach to inspection is not feasible in the real-world for economic reasons, but this is included for purposes of illustrating the consequences of different approaches. The opposite approach, the unprepared community (an especially risk-taking attitude) will also impose a significant cost on the community which is likely to exceed that of the other cases. The risk-neutral community takes the approach of minimizing the total cost, which, based on these results, is expected to occur by specifying \$30,450,000. Fig. (2.5b) shows the distribution of decisions made, considering the three levels of event intensity, for the risk-neutral community.

The budget that minimizes the imposed cost on the community may not necessarily minimize the variation of the cost. So let us consider how to determine the inspection budget that minimizes both the expected value and the variation of the cost. I thus define a risk-averse community as one which decides to rationally reduce the total cost imposed

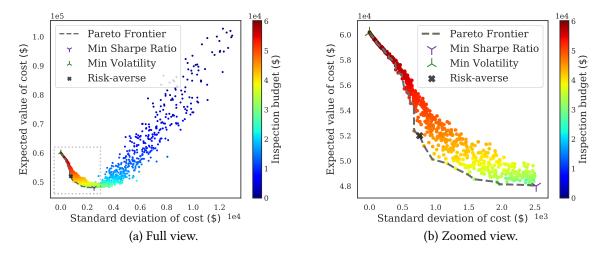


Fig. 2.6.: Pareto front.

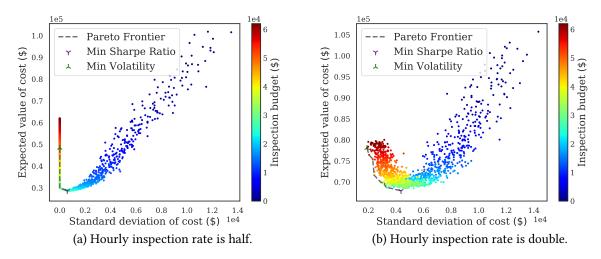


Fig. 2.7.: Cost distribution and Pareto front for regions with different hourly inspection rates.

on community, avoiding risk as much as possible. To accomplish this goal, I use modern portfolio theory which argues that an investment's risk and return characteristics should not be viewed alone, but should be evaluated by how the investment affects the overall portfolio's risk and return [57]. Because the application examines investing the budget in performing post-event inspections, instead of maximizing the return I need to minimize

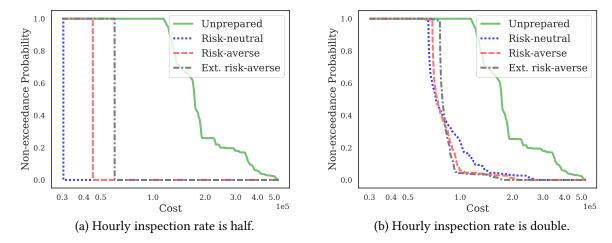


Fig. 2.8.: Non-exceedance probability of the cost for each community for regions with different hourly inspection rates.

the cost while minimizing the risk. The fundamental objective of this analysis is to identify an efficient set of budgets, known as an efficient frontier, that offers the minimum expected costs for a given level of risk. The gray dashed line in Fig. (2.6a), and (2.6b) shows the efficient frontier. Fig. (2.6b) is the zoomed view of the dotted box shown in (2.6a). Once I have the efficient frontier, a decision-maker can determine the desired inspection budget, considering other criteria, e.g., a maximum threshold of the budget or a maximum risk tolerance of the community. Here I assume that such a risk-averse community has set a maximum threshold of \$45,090,000 on the budget, as shown in Fig. (2.6). This result indicates that as long as the allocated budget is selected to be on the Pareto front, applying the other decision criteria of the community would not lead to a catastrophic result.

Fig. (2.5c) provides the statistics of the results for the risk-averse community for three events with different levels of intensity, including 0.1, 0.4 and 0.7. By selecting a reasonable inspection budget in advance, a majority of the high priority buildings are identified for field inspection while the buildings with lower priority are pre-classified.

The appropriate budget for a community to allocate does depend on the relative cost of inspection and the cost to the community for making incorrect predictions, see Eq. (2.3), which depend on the region. To consider such regional variations, I consider two other communities with lower (half) and higher (double) relative hourly inspection rates. Fig. (2.7a) shows the resulting cost distribution and Pareto front for the region with lower inspection rates. Here, half of the maximum budget is sufficient to inspect all the buildings. However, in a region with higher inspection rates, shown in Fig. (2.7b), spending the maximum considered budget will reduce the standard deviation considerably but it can not make it zero. Figs. (2.8a) and (2.8b) show the non-exceedance probability of the total cost for the sample communities in regions with both lower and higher inspection rates, respectively. In a community with a lower inspection rate, Fig. (2.8a) shows that the inspection budget assigned for the risk-neutral community will result in a very small chance of imposing more than 30,000 \$ cost on community. However, in a community with a higher inspection rate, Fig. (2.8b), it is probable that the resulting cost on the community would be 10 times larger, in this case 300,000 \$.

3. AUTOMATED BUILDING IMAGE EXTRACTION FROM 360-DEGREE PANORAMAS FOR POST-DISASTER EVALUATION

In this study, I develop an automated technique to detect and extract curated pre-event building images from typical street view panoramas. This technique is intended to support research investigating the impact of disasters on a building inventory. The extracted images capture the external appearance of a building from several viewpoints. As a preliminary step, a classifier is first trained to detect buildings within images using a large ground-truth building image set. The region-based convolutional neural network algorithm is exploited to design a robust building classifier [58]. Ideally, in the real-world application of this technique, a user can simply provide, as the input, a geotagged image (or similarly, a GPS coordinate) recorded near the target building, and the rest of the process is fully automated. The physical location of the target building is estimated using the geometric relationship between the panoramas and the building. Then, the optimal projection plane for each of the panoramas is determined to produce a high resolution, undistorted 2D building image from the corresponding panoramas. The region (position) of the building on each 2D image is determined using the trained classifier, and an image of the building is extracted from each 2D image. The output of the technique is a set of several undistorted images of the building, taken from all available viewpoints. Generating multiple image reduces the possibility of an obstruction (e.g. trees, cars or fences) in a particular viewpoint and, thus, enables robust visual assessment. Also, by using this procedure with data from multiple street view services, a user can collect street view images obtained over many past years to observe the target building over time. The performance of the technique developed here is demonstrated and validated using residential buildings affected by Hurricane Harvey (in 2017) in Holiday Beach, TX. Real-world post-event reconnaissance images, collected by engineers in the field, are used as the input for this

validation, and pre-disaster views of the buildings in those reconnaissance images are automatically extracted from Google Street View [59].

The major contribution of this study is to provide a practical and feasible solution to a problem that is grounded in the needs of engineers who seek to learn from disasters. With the capability to analyze post-disaster building scenarios by readily accessing pre-disaster images and post-disaster data, engineers are equipped with the tools to rapidly develop a greater understanding of the performance of our infrastructure. The technique developed automatically provides high-resolution undistorted multiple view pre-event images of each building. The key technical contributions are in automatically removing the inherent distortion of the 2D projection of panorama images and in rapidly extracting a set of images of each post-disaster target building with multiple viewpoints. Additionally, a side benefit of this technique is the ability to exploit the vast amounts of legacy visual data that exist from past disasters. With such visual databases being collected and established, and the cost of image acquisition and data storage decreasing, this technique offers just one example of how to automate the reuse of existing data through the novel application of state-of-art computer vision algorithms to solve real-world problems.

This chapter is structured as follows. First, the challenges in using street view images for this application are introduced in Section 2. The technical details of the technique are explained in Section 3, and in Section 4, it is demonstrated using Google Street View images gathered from Holiday Beach in Texas, United States, which was heavily affected by Hurricane Harvey in 2017. Please see the archived [44] and published versions [60] for more details.

3.1 Problem Statement

The 360-degree panorama image is an image that is able to capture all around (spherical) scenes from a given location. The merit of panoramas is that, after the data are collected, one may quickly navigate to scenes of interest in any direction from the given location. This advantage does not apply to 2D images. With 2D images, the direction in

which the data are being collected must be determined at the data collection stage. Thus, panoramas are quite useful when for reconnaissance because when they are collected from multiple locations, several different external views of a given building are automatically recorded (e.g., front or sides). In the panorama viewers typically implemented by street view service providers, 2D rectilinear images are rendered from the panoramas in real-time based on the selected viewing direction (represented by a pitch and yaw) and zoom level. When rendered in this way, the rectilinear images are just the 2D images that would typically be obtained with an ordinary (non-fisheye) camera. These images represent scenes in the world as people actually see and perceive them. For instance, a straight line in the 3D world is represented as a straight line in the corresponding rectilinear image [61]. Such viewers also frequently overlay directional arrows to enable a user to move to nearby panoramas. Although such viewers provide accessible, easy-to-use and easy-to-view panoramas, a great deal of manual effort is now required to retrieve data for a particular building. First, the user must read the GPS coordinate from a geotagged image and enters the GPS coordinates into the viewer. Then, by panning and zooming, the user finds the target building and determines the best perspective for a clear view the front of the building. The user captures the rectilinear image shown on the screen and crops the image to extract and save the building region to document this view. Next, to observe the side of the building (or any other angle), the user must click on the directional arrows to move to nearby panoramas and repeat the extraction procedure. This entire process is repeated for each of the panoramas (locations) yielding several images of the building from various viewpoints. This manual process is time-consuming and inefficient. Gathering multiple pre-disaster images for a large number of buildings in a subdivision or city would take a great deal of time for a human.

To automate this process, I incorporate two key capabilities into the technique developed. First, the position of the building on each image is identified. Herein, to avoid confusion, I use the term 'position' when I mean the location of the building on the 2D images. The term 'location' is used only for its geolocation in the world (3D). I exploit a recently developed deep convolutional neural network algorithm, which has led to break-



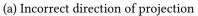


(a) Hurricane Katrina in 2005

(b) Hurricane Harvey in 2017

Fig. 3.1.: Sample images collected during post-event building reconnaissance missions (Courtesy of Timothy P. Marshall and Thomas P. Smith, respectively).







(b) Correct direction of projection

Fig. 3.2.: Rectilinear images created from the same panorama.

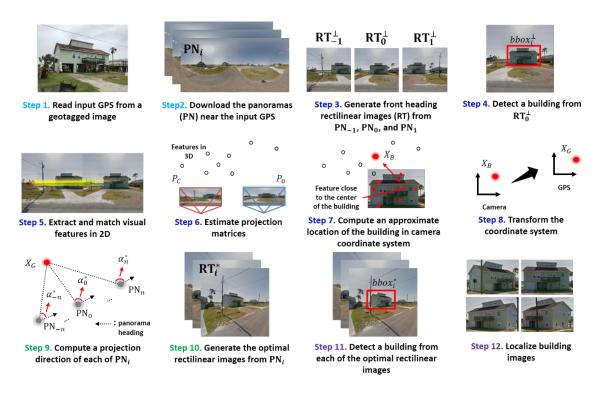


Fig. 3.3.: Overview of the technique developed.

throughs in object recognition [58]. Using a large number of labeled building images, a robust building classifier is trained to accurately identify the presence of the building and detect its position in the pictures. The details of this algorithm are explained in Section 3. Next, the optimal viewing direction (direction of projection) is determined. The direction of projection is defined as the viewing direction that generates a rectilinear image from the panorama to yield an undistorted view in the selected direction. However, inaccurate selection of the direction of projection may result in significant distortion of the target building. For example, Fig. 2 demonstrates the effect of

projection direction on rectilinear images. Both sets of images are generated from the same panorama. When the target building is located further from the center of a rectilinear image, its quality is degraded due to a large distortion. Thus, the most suitable direction of projection of each panorama should be determined to generate the 2D rectilinear image with the best available quality. The implementation of this capability is presented in Section 3.

3.2 Methodology

The technique developed herein is explained in Fig. 3. The details are as follows. Users provide a geotagged image of the target building (which includes the GPS coordinate near that target building) and the input, and several images of the corresponding building are automatically extracted from street view images and the outputs. The resulting images contain views of the building acquired from various viewpoints and with negligible distortion. This technique will enable the user to readily examine images of the entire front and sides of the building in its pre-event condition. The overall technique is divided into four main processes: Steps 1 and 2 (captioned in light blue) download all panoramas acquired near the target building from the street view service. Steps 3 to 8 (captioned in dark blue) approximate the building location in the GPS coordinate system using the geometric relationship between the building and the panoramas. Rectilinear images are generated from a couple of the closest panoramas, and the trained building detector is applied to find the building position on the rectilinear images. Then, the location of the building is identified in each of the camera coordinate systems, and transformed into the GPS coordinate system. Note that this intermediate step is not yet intended to extract high-quality building images because the rectilinear images are generated without considering the optimal direction of rectilinear projection. Steps 9 to 10 (captioned in green) are generate the rectilinear image from each of the panoramas by considering its optimal direction of projection. Lastly, Steps 11 to 12 (captioned in purple) detect the target building in each of the optimal rectilinear images using the same building detector and extract the building images for the final use. Again, once a geotagged image is provided in Step 1, the rest of the process is fully automated. The detailed description and pseudocode for each step are provided in the following paragraphs. Table 1 shows all parameters used in the pseudocodes and their definitions.

Table 1. Parameters and definitions

In Step 1, the GPS coordinate near the target building (hereafter, the input GPS coordinate) is obtained from the EXIF metadata of the input geotagged image(s). Note that

Table 3.1.: Parameters and definitions.

Parameter	Definition
img	Geotagged post-event image
C_{img}	GPS location of <i>img</i>
C_i	GPS location of PN_i
PN_i	<i>i</i> th panorama
M_i	Metadata of PN_i
θ	Field of view (FOV)
α	Projection angle in degrees
\perp	Perpendicular projection angle, ⊥∈ {90°, 270°}
$lpha_i^*$	Optimal α for PN_i
$lpha_i^* \ RT_i^lpha \ RT_i^*$	Rectilinear image generated from PN_i using α
RT_i^*	Rectilinear image generated from PN_i using α_i^*
$bbox_i^{\perp}$	Detected bounding box of a building in RT_i^{\perp}
$bbox_i^*$	Detected bounding box of a building in RT_i^*
kp_i	Detected key points in RT_i^{\perp}
des_i	Descriptors corresponding to kp_i
mp_i	Matched key points among kp_i
nmp	Number of the matched key points presented in the bounding box
x_i	kp_i , $i \in \{0, -1, 1\}$ where i is selected by comparing nmp
P_i	<i>i</i> th projection matrix
E	Essential matrix
ix_i	Inliers matched key points among x_i
R	Rotation matrix between P_i
T	Translation matrix between P_i
px_i	Building image location on RT_i^{\perp}
X_B	Estimated location of a building in camera coordinate
X_G	Estimated location of a building in GPS coordinate
bimg _i	Resulting building image extracted from RT_i^*

this image is only needed for providing the GPS information near the target building and thus, its quality or visual contents do not affect this technique. Alternatively, a user may manually provide approximate GPS information for the target building. The pseudocode for step 1 is shown in Algorithm 1.

Algorithm 1

Input: imgOutput: C_{imq}

1: $C_{img} \leftarrow \text{Read GPS location from metadata in } img$

Step 2 is to download a sequence of panoramas acquired near the input GPS coordinate, denoted as $PN_i(i = -n, ..., +n)$, from the Street View API. Herein, i = 0 represents the index for the panorama that is closest to the building, and i = -n and i = +n represent the indices for the far left and far right panoramas in the selected set, respectively. The sequence to be downloaded begins with PN(-n) where the nearest available panorama is set to PN_0 . For example, I assign that PN_1 becomes the panorama closest to PN_0 in one direction along the route where the panoramas are collected. Here, n is the maximum allowable number of panoramas to be used in one direction. The user may initially select this number. Panoramas that are captured far from the input GPS coordinate are not so useful because the building will be too small on the image. Thus, a reasonable quantity should be set (e.g., 3 or 5). Note that, depending on the building location, the number of available panoramas may not be 2n+1. For instance, a building located close to a dead-end or a cul-de-sac will have fewer panoramas available in one direction. Each downloaded panorama contains panorama heading information in the form of an angle with respect to North (see Fig. 4) which is necessary for finding the optimal direction of projection. In Google Street View, since this panorama heading is set to the driving direction of the data collection vehicle, the panorama heading of a given panorama is closely aligned with the direction of the street. The pseudocode for step 2 is shown in Algorithm 2.

In Step 3, front projection rectilinear images, denoted RT_i^{α} , are generated from $PN_i(i = -1, 0, 1, \alpha = 90^{\circ} \text{ or } 270^{\circ})$. As mentioned in Section 2, a rectilinear projection is one type of projection to reproduce 3D scenes using 2D images, and results in the most natural

Algorithm 2

```
Input: C_{imq}
Output: PN_{-n}, \ldots, PN_n, M_{-n}, \ldots, M_n
 1: M_0 \leftarrow \text{Read} metadata of the nearest panorama to C_{imq} using Street View APIs
 2: PN_0 \leftarrow Download the panorama corresponding to M_0
 3: for j \in [-1, 1] do
         i \leftarrow 1
 4:
         while i \leq n do
 5:
             k \leftarrow j * i
 6:
             M_k \leftarrow Find the linked panorama from M_{j*(i-1)}
 7:
             if M_k exists then
 8:
                  PN_k \leftarrow Download the panorama corresponding to M_k
 9:
                  i \leftarrow i + 1
10:
             else
11:
                  break
12:
```

way for the viewer [62]. The scenes (images) available in typical street image viewers are rectilinear images that have been projected to the direction that the user selects. This direction corresponds to the direction of projection in Fig. 4. Rectilinear images are produced by mapping the panorama scene (here, panorama scene shown along the curve from A to B) to the rectilinear image plane (AB). Due to the available quality of street view imagery, which reduces the chance of imperfection in panoramas, the projection error is negligible. (AB) is determined by defining a projection angle (α) and a field of view (θ) . Thus, an infinite number of rectilinear images can be generated from a single panorama by using different projection directions. PN in Fig. 3, and (AB) in Fig. 4 are represented by pixels. The field of view θ should generally be less than 120° because a large field of view produces large distortion in the contents of the scene at the edges of the image. In this step, only those panoramas immediately adjacent to PN_0 ($PN_{(}-1)$ and PN_1) are used for computing the building location in the GPS coordinate system. I only consider $PN_i(i=-1,0,1)$ because the corresponding RT_i^{α} are most likely to include the target building (i = -1, 0, 1). Since the true GPS coordinate for the building is unknown, the precise direction of projection for each RT_i^{α} cannot be determined. Here, I reasonably assume that the building is located along the route of the street where the panoramas are captured. Thus, the direction of projection is approximately set to either 90° or 270°, making it perpendicular to the direction of the panorama heading (the direction of the street). Depending on the location of the input GPS coordinate with respect to the panorama location and the projection angle is selected to be either 90° or 270°, here denoted as \perp . Note that this is not the projection direction used for extracting the final building images. Thus, images extracted using this projection direction are referred to here as front heading rectilinear images, RT_i^{\perp} . Herein, the actual projection direction for generating rectilinear images is represented by two angles, azimuth and pitch. However, I only consider the azimuth angle in this study, which is illustrated in Fig. 4. Thus, θ and α are the viewing and projection "azimuth" angles, respectively. Accordingly, (AB) becomes the projection plane in the horizontal direction. For the vertical direction, the pitch angle of the projection is set to zero, and θ for the vertical direction is selected to be identical

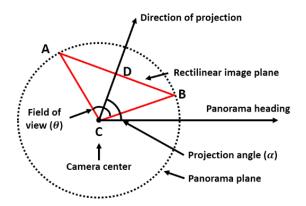


Fig. 3.4.: Geometry of a panorama and rectilinear image.

to the value for the horizontal. In low-rise buildings, since the optimal projection pitch angle is relatively small compared to the azimuth angle, the pitch angle does not need to be controlled, and when it is set to zero, distortion in the rectilinear linear image is insignificant. However, in the case of high-rise buildings, the pitch angles should also be controlled so that the view of the building in a vertical direction can be fully included in a corresponding rectilinear image. The pseudocode for step 3 is shown in Algorithm 3.

```
Algorithm 3
```

```
Input: PN_{-1}, PN_0, PN_1, C_{img}, \theta_0 = 120

Output: RT_{-1}^{\perp}, RT_0^{\perp}, RT_1^{\perp}

1: C_0 \leftarrow Read the GPS location of PN_0 from M_0

2: \theta \leftarrow Set FOV of rectilinear images as \theta_0

3: \beta_{relative} \leftarrow Calculate an angle between the true north and the vector \overrightarrow{C_0C_{img}}

4: \beta_{reference} \leftarrow Read a panorama heading angle w.r.t. the true north from M_0

5: \beta \leftarrow \beta_{relative} - \beta_{relative}

6: if \beta < 180 then

7: \bot \leftarrow 90

8: else

9: \bot \leftarrow 270

10: for i \in [-1, 0, 1] do

11: RT_i^{\perp} \leftarrow Generate a rectilinear image from PN_i using \bot and \theta
```

In Step 4, the position of the target building in RT_0^{\perp} is detected using a trained building (object) classifier. This building classifier is trained in advance using a large volume of

ground-truth building images. The details of the object detection procedure are explained in Sections 3 and 5.1. The same trained classifier is also used later in the technique in Step 11. The ground-truth building images used for training the classifier must include images of buildings that have a similar appearance as the target building. For instance, if the building images used for training only contain wooden buildings, the classifier may not be sufficiently accurate when classifying images of masonry or concrete buildings [63]. After applying the classifier to RT_0^{\perp} , I determine a tight bounding box for the building in that image. If more than one building is detected in RT_0^{\perp} , the bounding box that is closest to the center of RT_0^{\perp} is selected, denoted as $bbox_0^{\perp}$ (the subscription indicates the order, or index, of the image). The pseudocode for step 4 is shown in Algorithm 4.

Algorithm 4

Input: RT_0^{\perp} Output: $bbox_0^{\perp}$

1: $bbox_0^{\perp} \leftarrow$ Detect a bounding box of a building closet to the center of RT_0^{\perp}

Step 5 is to extract and match the visual features between RT_i^{\perp} (i=-1,0,1). The features extracted from RT_0^{\perp} must be matched with the corresponding features from RT_{-1}^{\perp} and RT_1^{\perp} . One of the conventional visual features and descriptors (SURF) is used for this process [64]. These visual features and their descriptors represent unique key points across the images and are used for computing the geometric relationship (essential matrix in this study) between images. Then, a single rectilinear image is selected from either RT_{-1}^{\perp} or RT_1^{\perp} , which is chosen as the one with the larger number of matched features, denoted as RT_c^{\perp} . This process is intended to improve the accuracy of building location estimation by using more information. For example, the building on RT_0^{\perp} is often shifted slightly because the input GPS coordinate may not be recorded at the location of the panorama closest to the house. In such a case, either RT_{-1}^{\perp} or RT_1^{\perp} may be far from the panorama closest to the building, and the other panorama is unlikely to include the same building, causing a failure in estimating the building location. Unless the building is far from the street along which the panoramas were captured, the building often occupies a large portion of each image, and thus many features generated from the building regions will be matched. A

subset of key points on RT_0^{\perp} which are matched with RT_c^{\perp} and included in the boundaries of $bbox_0^{\perp}$ is called x_0 . The corresponding subset of key points on the RT_c^{\perp} is called x_c . The pseudocode for step 5 is shown in Algorithm 5.

```
Algorithm 5
```

```
Input: RT_{-1}^{\perp}, RT_{0}^{\perp}, RT_{1}^{\perp}
Output: x_0, x_c
  1: kp_0, des_0 \leftarrow \text{Extract keypoints and descriptors from } RT_0^{\perp}
  2: nmp \leftarrow 0
  3: for i \in [-1, 1] do
           kp_i, des_i \leftarrow \text{Extract keypoints and descriptors from } RT_i^{\perp}
  4:
           mp_i, mp_0 \leftarrow Find matched keypoints from kp_0 and kp_i using des_0 and des_i
  5:
           nmp_i \leftarrow \text{Count the number of matched features inside } bbox_0^{\perp}
  6:
           if nmp < nmp_i then
  7:
                nmp \leftarrow nmp_i
  8:
  9:
                \mathbf{x_0} \leftarrow mp_0
10:
                \mathbf{x_c} \leftarrow mp_i
```

In Step 6, the projection matrices of RT_0^{\perp} and RT_c^{\perp} are computed. The projection matrix is a 34 matrix which represents the mapping from 3D points in the world to 2D points in an image. In this study, the projection matrices of RT_0^{\perp} and RT_c^{\perp} are computed using an essential matrix between two images (see Eq. 3), and the corresponding projection matrices are denoted as P_0 and P_c in (see Fig. 3), respectively. The essential matrix is a special case of a fundamental matrix for which the intrinsic (calibration) matrix for both cameras is known [65]. Since the essential matrix has only five degrees of freedom to be estimated from RT_0^{\perp} and RT_c^{\perp} , fewer degrees of freedom than the fundamental matrix, more accurate projection matrices can be computed. Based on the pairs of visual feature matches in Step 5, the essential matrix is estimated using the five-point algorithm combined with RANSAC (RAndom SAmple Consensus) [66,67]. The Principal points and focal length in the intrinsic matrix are obtained using the panorama's geometry in Fig 4. The intrinsic

matrices K for RT_0^{\perp} and RT_c^{\perp} are identical. For the mathematical representation of this process, the intrinsic matrix is represented as:

$$K = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f & 0 & p \\ 0 & f & p \\ 0 & 0 & 1 \end{bmatrix}$$
(3.1)

, where f and p are the focal length and the coordinates of the principal point, and the subscripts x and y indicate the width and height directions, respectively. Since I consider the same viewing angle in the horizontal and vertical directions, so f_x and f_y (and p_x and p_y) are the same regardless of the direction. In Fig. 4, \overline{CD} and $\overline{(AD)} = \overline{BD}$ are f and p, respectively. Thus, f becomes:

$$f = p \arctan \theta / 2 \tag{3.2}$$

, where p is half of the size of the rectilinear image in pixels, meaning that the principal point is the center of the rectilinear images. The pairs of matched feature points are transformed by multiplying the inverse of K by their points coordinates (in the homogeneous coordinate system). Then, those points are expressed in a normalized coordinate system, denoted as x_0^n in RT_0^{\perp} , and x_c^n in RT_c^{\perp} [65]. The essential matrix E satisfies the following relationship:

$$x_c^{nT} E x_0^n = 0 (3.3)$$

The five-point algorithm based on this relationship is utilized as a hypothesis-generator (model) for RANSAC to count the number of inliers and outliers, and the inliers are used for estimating E [65, 68]. The inliers on RT_0^{\perp} and RT_c^{\perp} are called ix_0 , and ix_c , respectively. These points will be used for computing the building location in Step 7. The estimated E enables us to extract P_c if P_0 is assumed to be a canonical projection matrix $P_0 = [I\,0]$ where I is a 33 identity matrix) [65]. In this case, the origin of the camera coordinate system is the camera center (focal point) of RT_0^p (generated from PN_0). The pseudocode for step 6 is shown in Algorithm 6.

Algorithm 6

Input: x_0, x_c

Output: P_0 , P_c , ix_0 , ix_c

- 1: $x_0^n, x_c^n \leftarrow \text{Normalize } x_0 \text{ and } x_c$
- 2: E, ix_0 , $ix_c \leftarrow$ Calculate the essential matrix using x_0^n , x_c^n
- 3: $R, T \leftarrow \text{Use } ix_0, ix_c$ and E to recover relative camera rotation and translation
- 4: $P_0, P_c \leftarrow \text{Use } R$ and T to estimate the projection (camera) matrix

In Step 7, the approximate 3D locations of the building are identified in the camera coordinate system. When the direction of projection is generally aimed toward the center of the building façade, distortion on the rectilinear images can be minimized. Given the projection matrices P_0 and P_c , I can compute the 3D points in the camera coordinate system that correspond to a pair of matched image points (point correspondence) using a linear triangulation algorithm [65]. From Steps 4 and 5, I have a set of matched image points between RT_0^{\perp} and RT_c^{\perp} . However, the visual features and/or the matched sets are randomly distributed over the building region on those images, and thus it is not guaranteed that I will obtain the exact center of the building façade. Moreover, depending on the locations of PN_0 and PN_c with respect to the building and the width of the building, they may include a portion of the side of the building, meaning that the center of the bounding box may not be the center of the building façade. Thus, I reasonably define the 3D building location (in the camera coordinate system) as the 3D point generated from the matching feature closest to the $bbox_0^*$. Once I find the matching feature nearest to this point, its corresponding location in 3D space is identified in the GPS coordinate system. The estimated building location is denoted as X_B . The pseudocode for step 7 is shown in Algorithm 7.

Algorithm 7

Input: P_0 , P_c , ix_0 , ix_c

Output: X_B

- 1: $px_0 \leftarrow \text{Find a point from } \mathbf{ix_0} \text{ close to the center of } bbox_0^{\perp}$
- 2: $px_c \leftarrow \text{Find a point from from } ix_c \text{ corresponding to } px_0$
- 3: $X_B \leftarrow$ Compute a 3D location by triangulating px_0 and px_c

Step 8 is to perform a 2D similarity transformation to define X_B in the GPS coordinate system, denoted as X_G . In Step 2, each of the panoramas downloaded from street view services has its own accurate GPS information and the 3D locations of PN_0 , PN_c , and X_B are computed in the camera coordinate system using Steps 3 7. However, these two control points (the same points in two different coordinate systems) are not entirely sufficient to compute a 3D similarity transformation between the two coordinate systems (Horn,

1987). Since the positions of the camera when it is acquiring the data (panoramas) are almost co-linear (because the street view vehicle is driving along a street), adding one additional control point by considering another (3rd) camera location does not add a new equation to obtain a unique transformation. Alternatively, I choose to eliminate one of the dimensions in both coordinate systems. Because the panoramas are captured from a street view vehicle, the camera acquiring the panoramas moves along a path that is almost entirely in a single plane with a consistent height (negligible street slope exists between two panorama locations) in the camera coordinate system. Accordingly, I can assume that the variation in the camera's altitude in the GPS coordinate system is minimal and can be ignored. With two pairs of camera locations in both the camera coordinate system (considering only the image width and depth directions) and the GPS coordinate system (considering only latitude and longitude), I have a sufficient number of equations to perform the 2D similarity transformation using the two translational, two rotational, and one scaling parameters that are available. Here, the camera location in PN_0 is $[0\ 0\ 0]$ and PN_c becomes $Q^{-1}p_4$ where Q is the left 33 submatrix of P_c and p_4 is the last column of P_c [65]. GPS coordinates are typically recorded in geodetic coordinates (represented as longitude and latitude). Note that this system is not a Euclidean space, as the one used in the camera coordinate systems. The geodesic datum should thus be represented by the equivalent values in Euclidean space [69]. Since the distance between the two panoramas is relatively small compared to the radius of the Earth, I can use the flat Earth assumption in which the values are transformed into an Earth-north-up (ENU) coordinate system [70]. The ENU coordinates are formed from a plane tangent to a fixed point on the Earth's surface. Coordinates of the points are found by computing translational movements on the tangential plane to East (X) and North (Y) from the fixed point. Then, finally, I can identify the 2D similarity transformation matrix using these two pairs of control points defined in two different Euclidian coordinate systems. The transformation matrix is then applied to X_B to obtain X_G . The pseudocode for step 8 is shown in Algorithm 8.

In Step 9, the correct projection directions for $PN_i(i = -n, ..., +n)$ are computed to generate the optimal rectilinear images. PN_i have the panorama heading with respect to

Algorithm 8

Input: C_0, X_B Output: X_G

1: $X_G \leftarrow$ Transform the coord. of X_B from ENU to Geodetic with the reference C_0

North, and their locations are defined in the ENU coordinate system obtained in Step 8. Thus, as shown in Step 9 in Fig. 3, the correct projection angles (α_i^*) can be computed. The pseudocode for step 9 is shown in Algorithm 9.

```
Algorithm 9
```

```
Input: X_G, M_{-n}, \ldots, M_n

Output: \alpha_{-n}^*, \ldots, \alpha_n^*

1: for i \in [-n, \ldots, n] do

2: C_i \leftarrow Read the GPS location from M_i

3: \alpha_{relative} \leftarrow Calculate an angle between the true north and the vector \overrightarrow{X_GC_i}

4: \alpha_{reference} \leftarrow Read a panorama heading angle w.r.t.the true north from M_i

5: \alpha_i^* \leftarrow \alpha_{relative} - \alpha_{reference}
```

In Step 10, RT_i^* are generated from PN_i using α_i computed in Step 9 (i = -n, ..., +n), and these are the optimal rectilinear images. This step repeats the same process in Step 3 for all panoramas, although at this point the process is performed using the correct projection angles. Since the projection direction is aimed toward the target building, the target building is now at the center of each RT_i^* . The pseudocode for step 10 is shown in Algorithm 10.

```
Algorithm 10
```

```
Input: PN_{-n}, ..., PN_n, \alpha_{-n}^*, ..., \alpha_n^*, \theta_0 = 120
Output: RT_{-n}^*, ..., RT_n^*

1: \theta \leftarrow Set FOV of rectilinear images as \theta_0

2: for i \in [-n, ..., n] do

3: RT_i^* \leftarrow Generate a rectilinear image from PN_i using \alpha_i^* and \theta
```

In Step 11, the position of the target building in each RT_i^* is detected using the trained building classifier, which is identical to the classifier used in Step 4. The only difference compared to Step 4 is that here I detect the building within the optimal rectilinear images generated from all panoramas, which contain undistorted views of the target building. The pseudocode for step 11 is shown in Algorithm 11.

Finally, in Step 12, I obtain the set of highly localized target building images captured from various viewpoints. The detected target building images having various viewpoints

Algorithm 11

Input: RT_{-n}^*, \dots, RT_n^* Output: $bbox_{-n}^*, \dots, bbox_n^*$ 1: **for** $i \in [-n, ..., n]$ **do**

 $bbox_i^* \leftarrow \text{Detect a bounding box of a building close to the center of } RT_i^*$

are cropped from RT_i^* , which are captured from different locations. The pseudocode for step 12 is shown in Algorithm 12.

```
Algorithm 12
```

Input: $bbox_{-n}^*, \ldots, bbox_n^*, RT_{-n}^*, \ldots, RT_n^*$ Output: $bimg_{-n}, \ldots, bimg_n^*$

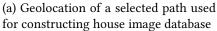
1: **for** $i \in [-n, ..., n]$ **do**

2: $bimg_i \leftarrow \text{Extract a region of } bbox_i^* \text{ from } RT_i^*$

In this study, the visual recognition of the building is used both for estimating its 3D location by computing the geometric relationship between the panoramas (Step 4 in Fig. 3) and for detecting and cropping its region on each rectilinear image (Step 11 in Fig. 3). Accurate detection and positioning of each building on the images are critical for achieving the successful extraction of pre-event building images from the panoramas. Recently several CNN based high performance object detection algorithms have been presented [58,71,72]. I incorporate a state-of-art object detection method, called faster regionbased convolutional neural network (Faster-RCNN) into this technique [58]. Faster-RCNN is an evolved version of region-based convolution neural network in terms of speed and accuracy [58, 71, 73–78]. Recently, an enhanced version of Fast R-CNN, in terms of speed and accuracy. Many other architectures have been introduced to reduce the training and testing speed as well as the accuracy, but there is always a trade-off between the accuracy and computational efficiency [72,79]. In this study, I implement the original Faster R-CNN to detect buildings on the images. To close this section, I comment on some assumptions used in the technique that one must remember for successful implementation. First, of course, the panoramas used must contain the target building. Updates on the panoramas in the street view are infrequent in remote areas, and this highly depends on the location (i.e., urban or rural regions), although data collection is generally increasing in frequency. Recently constructed buildings may not be captured in the panoramas, and renovated buildings may not always have up-to-date images. One possible solution is to explore various street view services as their data collection periods and frequencies are different. Second, the geotagged image used as the input to the technique should be collected from

a location near the target building. The GPS coordinates stored in the metadata should be close to the target building. Recall that the actual visual contents of this input image is not actually used in the technique, and its quality is not relevant. However, it is recommended that each post-disaster geo-tagged image include only one building to prevent confusions while comparing the images with pre-disaster images. For example, if the input image is acquired from a location that is closer to a different building, images of the wrong building may be generated. Third, the technique relies highly on the availability and accuracy of information in the street view service used. Here the panoramas and related metadata are directly obtained from street view services. If such information has limited availability or it is not accurate, the results will be incorrect. For example, incomplete or erroneous GPS coordinates for the panoramas would yield the wrong a building location, followed by erroneous projection direction estimation. However, GPS data is generally accurate enough for this purpose, and I have not observed any errors in the data to date. Although I successfully demonstrate the technique using Google Street View, I have not tested it using panoramas available through the other street view services. Fourth, the panoramas acquired must have sufficient spatial coverage (roughly not more than 10 meters). If the distance between adjacent panoramas is too far to contain the same target building, the technique will fail to correctly estimate the location of the building in Steps 3 to 6. Moreover, such sparse panoramas hinder the goal to obtain several high-quality building images. Fifth, there should be a reasonable distance between the panoramas and the building, enough to ensure that the panorama does exclusively contain the building in a rectilinear image. Since most of the panoramas are captured along the street, their distances from the building are often sufficiently far from the target building (e.g., more than the width of a single lane). However, when the distance between the buildings and the panorama location and/or their height or width are large, a greater FOV is needed to capture the entire view of the building. This limitation would cause a large distortion in the rectilinear images.







(b) Samples of damaged residential buildings after Hurricane Harvey in 2017 (Metz, 2017)

Fig. 3.5.: Test site for experimental validation, Holiday Beach.

3.3 Experimental Validation

3.3.1 Description of the test site

To demonstrate the performance of the technique, I use residential buildings in Holiday Beach in Rockport, Texas as

this case study. Hurricane Harvey in 2017 is the second-most costly hurricane in U.S. history, inflicting \$125 billion in damage, primary due to catastrophic rainfall-triggered flooding in the Houston metropolitan area (Smith, 2018). Harvey made landfall as a Category 4 Hurricane in southern Texas, and Rockport was directly in the path of Harvey, causing tremendous wind and storm surge damage (Metz, 2017). Holiday Beach, shown in Fig. 5(a), is a residential community and most of its residential buildings (more than 80%) (hereafter, houses) in the region were substantially damaged (Villafranca, 2017). Several reconnaissance teams were dispatched to these regions in the weeks and months after the event to evaluate structures, characterize the event, and collect data to learn from this event (FEMA, 2018). Several such teams published their data through designsafe-ci.org and weather.gov (Metz, 2017; Stark and Wooten, 2018). Geotagged images collected from Holiday Beach are also available, the selection of sample images shown in Fig. 5(b) high-lights the need for observing their pre-disaster condition. It is evident that when presented with such photos of severely damaged houses, little information is available to identify





Fig. 3.6.: Sample panorama and the corresponding optimal rectilinear image for a target house.

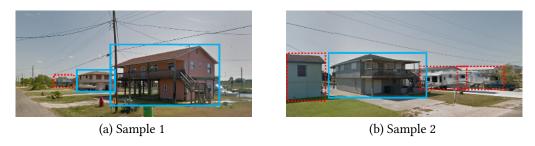


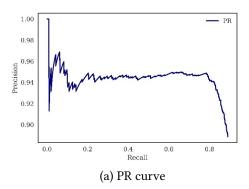
Fig. 3.7.: Sample building images used for training a residential building (house) classifier.

the vulnerabilities that may have existed. Here, these post-disaster reconnaissance images are the input to the technique and are used solely for providing GPS coordinates to automate the process of extracting useful photos of the target houses.

3.3.2 Construction of residential building image database

A large volume of ground-truth house images is prepared to train a residential building classifier using the algorithm discussed in Section 3. The house images used for this training process should be similar in appearance (architecture style) to those expected for actual testing and implementation. For instance, the goal here is to detect wooden residential buildings for single family residence. Wood is a standard construction material for houses across the Southern part of the United States. Training a classifier using

images of high-rise apartments or concrete buildings would not improve its performance unless those are expected to be present in the images collected from the testing and implementation regions. For simplicity, I prepare the ground-truth training images using representative scenes from Holiday Beach to ensure the buildings have similar styles and appearances. The distribution of the buildings used in the ground-truth dataset is quite similar to the appearance of houses across the other coastal areas of the United States. To prepare the training images for the classifier, I exploit Steps 1 to 10 from the technique developed to generate a large volume of optimal rectilinear images. Then, I manually label that large volume of images to construct the ground-truth residential building database. Note that the resulting residential building classifier is generally applicable, and is trained in advance for use across many events and regions. The database can also be expanded over time to encompass other building types, materials, architecture and styles. First, panoramas along the selected route on the waterfront in Fig. 5a (marked a red line) are downloaded from Google Street View. Google Street View does not provide a service to directly download high-resolution panoramas, but does have an API to download (show) a portion of the panorama when the user specifies a horizontal and vertical position and a size. The tool that I developed automatically downloads the tiles of each panorama and stitches them together to generate high-resolution panoramas. The resolution of each panorama is 13,312 × 6,656 pixels. Each panorama is constructed by stitching 338 tiles having a resolution of 512 × 512 pixels. A total of 128 panoramas are available along these routes. A sample of a high-resolution panorama is shown in Fig. 6(a). Second, I manually select the GPS locations of many of the houses along the chosen route in Fig. 5(a). A footprint (outline) of each house can be viewed in Google Maps, and thus, its GPS location can be easily obtained. 100 houses are considered along the route having similar house styles (e.g., number of floors, roof style, elevated house foundation). Note that for constructing the ground-truth database, this manual process is necessary and replaces the geotagged images that would be used in the implementation of the technique following Steps 3 to 8. Third, optimal rectilinear images are generated from the downloaded panoramas. Since here I know the GPS coordinates of each of the downloaded panoramas as well



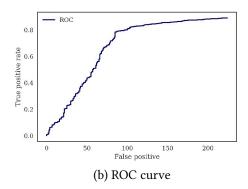


Fig. 3.8.: The performance of the residential building detector for the testing set.

as the houses, the optimal projection angles can be directly computed. Here, n is set to be five, which is the maximum allowable number of images on one side of the closest panorama location. Thus, for each house, a maximum of 11 rectilinear images are constructed from the corresponding panoramas (houses at a dead-end along the route would have fewer than 11 panoramas), and each house is positioned at the center of each of the rectilinear images. θ and \overline{AB} are 120° and 2,048 pixels, respectively. As a result, a total of 1,056 rectilinear images having a resolution of 2,048 × 2,048 pixels are generated for the database. Figure. 6(b) shows the optimal rectilinear image generated from the sample panorama in Fig. 6(a). Finally, the houses on the rectilinear images are manually labeled. I used a Python-based open source labeling tool to label a tight bounding box around each house [80]. Only houses that satisfy the following two conditions are labelled: (1) less than 70% of the front and side views of the house are obstructed by foreground objects, and (2) the height or width of the bounding box encompassing each house is larger than 200 pixels, which is around a tenth of the rectilinear image size in pixels. Samples of labeled houses are shown in Fig. 7. Here, only bounding boxes with solid blue lines are labeled as a house, while the other houses in these images are not included as ground-truth data. Those houses are marked with a red dotted line purely for purposes of illustrating non-labeled house data. The left image in Fig. 7, the red dotted bounding box around the



Fig. 3.9.: Samples of detected residential buildings using Faster R-CNN algorithm.

house on the left has height less than 200 pixels, and thus it does not contain sufficient information for this purpose. Similarly, in the right image in Fig. 7(b), all non-labeled houses are obstructed by the foreground objects (e.g., vehicle, tree, or other house) or are only partially visible. A total of 3,500 ground-truth residential buildings are labeled from the rectilinear images, and this database is utilized for training the classifier.

3.3.3 Training and testing residential building detector

An open-source library of Faster R-CNN deployed using Python is used for training and testing the residential building detector [81]. A single NVIDIA Tesla K80 is used for this computation. A total number of 3,500 residential houses have been labeled from 1,050 images. This set includes different views of 100 houses in Holiday Beach. Sets containing 60%, 10%, and 30% of the images of target houses are randomly chosen as training, validation, and testing sets, respectively. The Residual Network Model (called ResNet 101) is selected for learning robust features and increasing its efficiency [82]. I manually chose the hyper-parameters for Faster R-CNN based on trial-and-error. Then, the ConvNets and Fully-connected layers are initialized by zero-mean Gaussian with standard deviations of 0.01 and 0.001, respectively. Hyperparameters including learning rate, momentum and weight decay for training R-CNN and RPN networks are set to 0.001, 0.9, and 0.0005, respectively. The learning rate is defined as the amount the weights are adjusted with respect to the loss gradient. The decay weight is set to avoid overfitting by decaying the weight proportionally with its size. I set the momentum to 0.9 to make the gradient decent achieve a faster convergence. I used three different aspect ratios (0.5, 1, and 2) and anchor sizes (128, 256, and 512 pixels). In the test stage, in cases in which a set of the detected proposals that overlap each other with greater than a 0.3 intersection over union (IoU), an additional process is needed to obtain a precise bounding box. In each set, the proposal having the highest confidence score (probability) remains, and the others are disregarded. This process is called non-maximum suppression. Then, a threshold is set to keep only proposals having high confidence scores. In this study, this threshold is set to 0.5. I eval-

uate the performance of the trained residential building detector using the testing image set. Precision-recall (PR) and receiver operating characteristic (ROC) curves are used to evaluate the performance of the classifier quantitatively. To construct the PR curve, each detection first maps to its most overlapping ground-truth object samples. In this study, the threshold for considering a detection as successful is defined as an overlap of more than 50% IoU. True positive is defined as a detection with the highest-score (probability) mapped to each ground-truth sample, and all other detections are considered as falsepositives. Precision is defined as the proportion of true positives to all detections. Recall is the proportion of the true positives to total number of ground-truth samples. Plotting the sequence of precision and recall values yields the PR curve shown in Fig. 8(a). The typical method to evaluate the performance of the object detector is to calculate the average precision (AP) [58,76], which is the area under the PR curve [83]. The results show an AP of 85.47% for the single class of residential house detection. An ROC curve, which represents the relationship between sensitivity (recall) and specificity (not precision), is also used to evaluate the performance of the detector. As mentioned previously, each detection is considered to be positive if its IoU ratio with its corresponding ground-truth annotation is higher than 0.5. By varying the threshold of detection scores, a set of true positives and false positives will be generated which is represented as the ROC curve. From Fig. 8(b), I can interpret this result to mean that the residential building detector consistently achieves impressive performance in terms of the ROC curve by obtaining true positive rates of 81.49% and 88.07% at 100 and 200 false positives, respectively. Samples of detected houses using the optimal rectilinear images are shown in Fig. 9. Figures 9(a) and (b) shows typical successful cases in which all bounding boxes are correctly detected and tightly encompass each of house areas. Figures 9(c), (d), (e) and (f) show more challenging cases, which likely produce incorrect results. These results can be polished by adjusting the parameters used in the technique. Figures. 9(c) and (d) show two particular images which, in addition to correctly detecting houses, erroneous objects are also detected as a house. Figure. 9(c) demonstrates a case in which a boat is detected as a house with a score of 0.779. Also, in Fig. 9(d) a cargo trailer is detected as a house with a score of 0.571 which





Fig. 3.10.: Post-disaster building reconnaissance images of two different houses after Hurricane Harvey.

is not of interest in this study. As illustrated, all incorrect detections yield a low score, less than 0.8, which enables us to simply remove them using a higher threshold value for positive detection. Since in this application the target house is only likely to appear close to the center of the optimal rectilinear image, the target house would be detected with a high score, usually more than 0.95, unless either the image was captured far from the target house, or obstacles conceal the target house. Therefore, the problem of erroneous detections can be remedied by merely increasing the confidence threshold, here 0.8, to retain only the objects that receive a high score.

Figures. 9(e) and (f) illustrate cases in which all houses are correctly detected, even though they are partially occluded by foreground objects such as trees, as with the houses in the far left and middle of the images shown in Fig. 9(e) and (f), respectively. However, the occluded detected houses are not informative or useful for inspection purposes. As is mentioned in section 4.3, the appearance of the residential house to be used for inspection purposes is a critical characteristic of the object of interest. Since the ground-truth dataset is generated based on this principle, the building detector also associates a lower score with these non-informative appearances of houses. For instance,

the occluded houses shown in Fig. 9(a) and (b) are detected with scores of 0.839 and 0.507, respectively, which are considerably lower than the non-occluded houses appearing in these images. Although these objects are correctly detected as a house, the images are



Fig. 3.11.: Samples of the pre-disaster house images generated from Google Street View: (a) and (b) shows the same houses in Fig. 10 (a) and (b), respectively, before Hurricane Harvey and (c), shows a house in the same neighborhood with two others but its post-disaster geo-tagged image cannot be accessed.

not useful for inspection purposes. Considering the fact that I extract several (here, 11) optimal rectilinear images for each target house, every image need not be informative for inspection, and I can safely increase the confidence threshold to remove those non-informative house detections.

3.3.4 Sample implementation

In this section, the implementation of the technique developed is demonstrated using real-world post-event reconnaissance geotagged images collected from Holiday Beach, Texas after Hurricane Harvey in 2017. Pre-disaster images of two selected houses are automatically generated from the corresponding geotagged post-disaster images in Fig. 10. The house in Fig. 10(a) is merely a shell gutted by the hurricane and its appearance before the hurricane is hard to guess. The house on the right in Fig. 10(b) has significant shingle damage on the roof. One additional house is added, which does not have a geotagged image. The approximated GPS information near that second house is manually provided, rather than through a geotagged image. The locations of these houses are along the

selected route in Fig. 5(a). All pre-disaster images for these three different houses are automatically generated using the technique developed the proposed approach incorporates several program libraries including OpenCV [84] and PyMap3D (Hirsch, 2018), and an implementation Faster R-CNN algorithm in TensorFlow (Chen and Gupta, 2017), and is deployed as a Python script. For this testing, I utilize the same computing resources to run this script, which are used for training the house classifier. For setting up the parameters introduced in Section 3, n is set to five, so the maximum number of house images available is 11. θ and \overline{AB} are 90° and 2,048 pixels, respectively, which are the same as those used for generating training images. With this setup, at each house, the approximate processing time for conducting the four main processes explained in the first paragraph of Section 3 are approximately 130, 210, 180 and 6 seconds, respectively. As mentioned in Section 4.2, high-resolution panoramas cannot be downloaded from Google Street Views directly. It takes a considerable time to create each panorama by stitching an array of the images.



Fig. 3.12.: Reason to observe the house from multiple viewpoints: Optimal rectilinear images with the bounding boxes of the detected house from multiple views. Since the views of the house are partially or entirely obstructed by the foreground objects (here, trees), many images from different viewpoints should be considered to observe the entire front and side views of the house.

Unfortunately, this cannot be technically addressed in the front-end software, unless in the near future one is allowed to download panoramas from Google Street View directly.

However, the rest of the three processes are relatively fast and can be potentially improved by exploiting better hardware or optimizing the deployment of the Python scripts. The results in Fig. 11 show the pre-disaster house images automatically extracted from Google Street View using this technique. Figures. 11(a) and (b) show the pre-event appearance of the houses shown in Fig. 10(a) and (b), respectively, and Fig. 11(c) demonstrates the generation of pre-event images of the same house without a post-event geo-tagged image. The seven images of houses shown in Fig. 11(a) and (b) and the five images of houses shown in Fig. 11(c), from various viewpoints are detected from the 11 optimal rectilinear images, although only four representative samples of those images are provided in Fig. 11. It is clear from these samples that including images of each house from multiple viewpoints is important for enabling observation of the entire side and front façade of the house. Clearly, they provide valuable information about the pre-hurricane state of the house. For example, the pre-event appearance of the house shown in Fig.11(a) can hardly be imagined using only the post-disaster image in Fig. 10(a). Figure 12 shows the six optimal rectilinear images generated for the house in Fig. 11(c). The bounding

box of the corresponding house in each rectilinear image is marked. This example further demonstrates why multiple images from different viewpoints should be extracted (in other words, n should be set to more than two). Since large trees in front of this house block its view, multiple images are needed to observe the entire view of the house. This situation commonly occurs in cluttered scenes where various sources of foreground objects are potentially present in a street view, for instance, large street signs, parked vehicles, trees, or pedestrians. This issue can be minimized by increasing n to consider more viewpoints. 4.5 Technique validation

For further evaluation of the technique developed, I introduce a new metric, denoted Overall Practicality (*OP*). In this technique, the potential sources of error include: (1) incorrect estimation of the GPS location of the building due to a feature matching failure (Steps 3 8), (2) false building detection on the rectilinear images using the trained classifier, and (3) incorrect GPS records for the street view panoramas. The new metric, *OP*, is designed to quantitatively evaluate the likelihood of extracting a sufficient set of building images of the quality of the resulting images in Fig. 11. *OP* is formulated as:

$$OP = N_u / (N_t - N_o),$$
 (3.4)

where, N_u is the number of extracted images that contain a satisfactory view of the target building on the corresponding rectilinear images. When the extracted building is not placed near the center of the rectilinear images, due to the projection direction error, it does not contribute to N_u although it contains enough of the building's appearance. Also, if the trained classifier only detects a portion of the building or does not tightly estimate its region with a bounding box, it is also not included in N_u . N_t is the number of all building images which can be extracted from the available panorama. It can be calculated as $N_t = N_b(2n+1) - \sum_{i=1}^{N_b} m_i$, where N_b is the total number of target buildings, n is the maximum allowable number of images (see Section. 5.2), and m_i is the number of missing panoramas at building i. As mentioned in Section 3 (Step 2), if a building is located close to a dead-end or a cul-de-sac, street view panoramas are not available. N_0

is the number of occluded images. The occlusion due to foreground objects (e.g. tree, car, or fence) is inevitable, and can obstruct the view of the building. Thus, it is a clear basis for needing multi-view images. This number is not contained in either N_u or N_t . To evaluate the performance of the technique developed using OP, I randomly select 50 residential buildings along a different street (marked using a black line in Fig. 5(a)). In the actual implementation, the input of the technique is a geo-tag image which must be captured close to the building-of-interest. However, existing data sets rarely have a sufficient number of geo-tagged images available. Thus, in this evaluation, I manually provide similar GPS information for each building with the assumption that these GPS data can be obtained from geo-tagged post-disaster images during actual usage of the technique (step 1 in Fig. 3). Recall that this GPS information corresponds to a location on the street close to each building, and not the location of the buildings. Having this GPS information, I input GPS information to Step 2 and execute the rest of the steps. The performance is quite successful, and I obtain an OP of 0.838. Among 50 buildings, only one building has three missing panoramas, and thus, $N_t = 547$ and N_o and N_u are 39 and 426, respectively.

4. PRE-EVENT AND POST-EVENT DATA COLLECTION AND ANALYSIS USING ARTIFICIAL INTELLIGENCE

Here the objective is to develop and validate an automated technique to directly support engineers and architects as they conduct a preliminary survey. The physical and structural attributes of the buildings are of the critical features for predicting the response of the building to disruptive events [41,85–87]. I establish the ability to categorize buildings based on their physical and structural characteristics and on their overall post-event structural condition. An enabling factor in the proposed method is the availability of powerful convolutional neural network algorithms (CNNs) that are implemented for scene (image) classification to identify the structural characteristics and condition of the target buildings on the images. The technique is developed by dividing it into pre-event and post-event streams, each with the goal to first extract all possible information about the target buildings from pre-event images and post-event images. I verify and validate the proposed technique using post-event images captured during hurricane Harvey and Irma reconnaissance missions collected by a structural wind and coastal engineering reconnaissance team, NSF Structural Extreme Events Reconnaissance (StEER) Network.

The significant contribution of this research is to develop a technique capable to overcome real-world challenges to fuse the extracted information from multiple pre-event and post-event images to reliably categorize buildings based on their key physical attributes and rapidly determine their post-event overall buildings condition. Recent studies have conducted to facilitate the particular applications of damage detection using images that were collected with the intention of using them for that specific purpose. This technique is supposed to be used in real-world applications rather than focusing on image classification. So, I designed the technique be able to deal with the typical unorganized images captured during preliminary reconnaissance missions in real-world post-event circumstances, including relevant and irreverent images, and ultimately classify the buildings

based on the field engineers' desirable goals. Using the visual contents of the images, this technique classifies the images and selects only the images that potentially can be used for the overall structural condition of the building. I fine tune a decision fusion algorithm to integrate the result of the CNN-based classifiers of all images properly and determine the physical attributes and overall structural condition. I also consider the quality and completeness of the data collected in the field after the real-events which highly bias the results to make the ultimate decisions. Considering the fact that the damage is not visible on all the post-event images, the result of all images of each building will automatically be fused to make the final decision about the overall structural condition of the building. However, to detect the physical attributes of the buildings, an agreement between all points of view is the key to make the final decision.

The remainder of this chapter is organized as follows: Sec. 4.1 provides the problem formulation. Sec. 4.2 provides a demonstration and validation of its effectiveness. Please see the archived [43] and published versions [88] for more details.

4.1 Technical approach

A general diagram of the technique developed is shown in, Fig.4.1. The input is a collection of geo-tagged, post-event images of the residential buildings in a region. The output is the information needed for an assessment of each residential building, including automatically generated physical and structural attributes plus post-event condition information. Certain necessary physical and structural attributes are best obtained from the pre-event condition, so multiple pre-event images are automatically extracted from existing street view databases. Post-event building condition information is obtained directly from post-event images.

The technique is implemented through two branches of data analysis, conducted independently. I call these two branches the *post-event data analysis stream* and the *preevent data analysis stream*. The post-event stream detects assesses the overall damage condition of the building after the event based on the images collected during the pre-

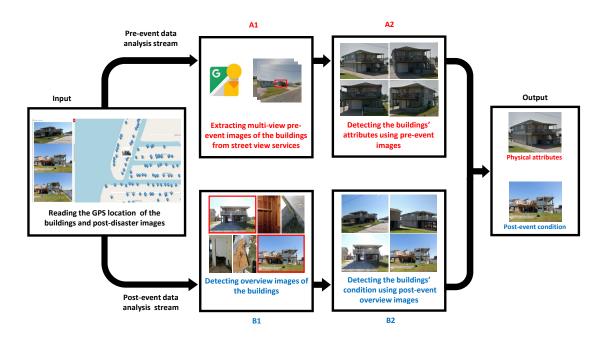


Fig. 4.1.: Diagram showing the steps in the automated procedure.

liminary survey. The pre-event stream extracts building physical attributes to be used for the preliminary screening, as well as several pre-event views of the building from various perspectives. These two sets of complementary information are organized in a way that assists the decision-making process of human inspectors regarding where to focus resources during a detailed survey. For clarity, I design a classification schema specific to post-event preliminary surveys. The schema can be easily extended to support other applications. In the subsequent paragraphs, I discuss the process use to develop each data analysis stream. The detailed definitions for the classification schema are provided in Sec. 4.1.1.

The post-event data analysis stream requires the design and training of two image classifiers which are implemented sequentially. The first classifier is intended to filter out images that contain useful information about the condition of the building, step *B1*. The best images for detecting the overall condition of the building for hurricane assessment are images that provide a view of the entire building. However, the data collected for a given target building may include close images of components or details, or even irrelevant images (e.g., cars, trees, windows, doors, etc). Including these in the dataset to be automatically analyzed may bias the results, or increase the processing time. The filtered data are passed to the next classifier, which is trained to detect the overall condition of the structure, step *B2*, see Sec. 4.1.1.

The pre-event data analysis stream automatically detects certain physical attributes of each building that are useful in a preliminary post-event survey using image classification. Since post-event images of buildings that have experienced severe damage cannot reliably be used to determine the original physical attributes, it is more appropriate to use pre-event images for this purpose. To this end, I developed a fully automated technique to extract pre-event images from street view imagery services, step *A1*. These pre-event images along with the ground truth labels, provided by the field engineers [89], are used to design and train a set of image classifiers, that can detect certain physical attributes, explained in Sec. 4.1.1, step *A2*.

In some cases, reliable determination of a physical attribute or even the condition of the building requires that classification results from several images containing multiple views of the building be used. For instance, if several post-event images are collected from a building, and only one of those images provides a view of the damaged region, the classifier will only detect damage in that one specific image; The specific image containing the damage cannot be known in advance. Therefore, the relevant images available must be used collectively to make a determination. I have developed an approach to fuse the information from several images to make such decisions. The problem formulation is provided in Sec. 4.1.2 and the demonstration is included in Sec. 4.2.

4.1.1 Design of the classification schema

The classification schema designed to support preliminary hurricane surveys is shown in, Fig. 4.2, (the abbreviations are defined later). Classifiers are much more effective when clear boundaries exist to distinguish the visual features of the images in different classes. This is especially true to achieve robust classification in the real world when using such unstructured and complex data, as is often the case in reconnaissance datasets. Thus, a clear definition for each class is needed to establish consistent ground-truth data that are suitable for training. The definitions for those comprising the post-event and pre-event streams are discussed in the following sections.

Classifiers used in the post-event stream

The procedure used in the post-event data analysis stream is shown in, Fig. 4.3. Two classifiers are used for classification of the post-event data, one to filter out less valuable images from the larger set, and a second to determine the condition of the building. These are applied to the dataset sequentially, as shown in, Fig. 4.2b.

The first classifier needed for post-event data analysis is called the *Overview classi- fier*. This is a binary classifier that filters flags images that show a sufficient view of the

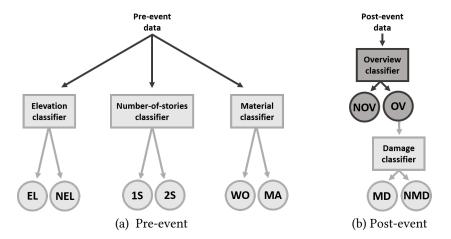


Fig. 4.2.: Hierarchy of classifiers used in pre-event and post-event data analysis streams.

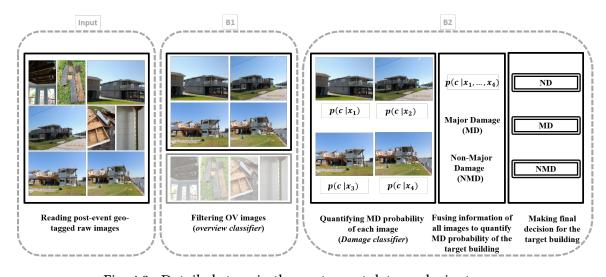


Fig. 4.3.: Detailed steps in the post-event data analysis stream.



Fig. 4.4.: Samples of images classified as overview (OV) and non-overview (NOV).

building. Each post-event image is classified as either "Overview" or "Non-Overview," as indicated in Step *2A*.

The Overview classifier is defined as:

• Overview (hereafter, OV): Images classified as OV show the entire building, irrespective of whether it is damaged or not, in the sense that they contain more than 70% of the facade (with either a front view or a side view) and they include portion of the roof. To include the possibility of severe damage, an image with some standing columns, or a pile of debris which can clearly be identified as a collapsed building, is also classified as OV. Examples of the latter include images of the general overall view of standing structural members or a collapsed roof. An additional restriction of OV images is that no more than 20% of the image area shows the surrounding



Fig. 4.5.: Samples of images classified as major damage (MD) and non-major damage (NMD).

buildings. In some cases, partial obstruction, by trees, cars, and other buildings, is an inevitable challenge. However, if the obstruction hides less than 30% of the building facade, I still consider the image as an OV.

 Non-overview (hereafter, NOV): Images that are not OV are NOV. Examples of NOV include images of the interior of the building, measurements, GPS devices, drawings, multiple buildings, building facades occluded by trees, cars or other buildings.

Samples of images defined as OV and NOV are shown in Figs. 4.4a and 4.4b, respectively.

Next, as shown in, Fig. 4.3, the subset of images classified as OV are analyzed collectively to determine the overall building condition. A classifier is trained to determine whether a single OV image should be labeled as "Major damage" or "Non-major damage," which includes both minor and no damage. I call this binary classifier the *Damage classifier*. Note that a single image is not sufficient to characterize a building as it may be

showing a side from which damage is not visible. Therefore, after classifying the damage in each OV image of a given building, the overall condition must be decided by fusing all available information (this will be discussed in Sec. 4.1.2). The Damage classifier is defined as:

- Major damage (hereafter, MD): Images classified as MD contain visual evidence of severe damaged by wind, wind-driven rain, or flood. Specific examples include signs of roof collapse, and column, wall or exterior door failure. In the case of severe water intrusion/damage, I also classify the image as MD. Considerable damage to the roof or exterior doors or windows or garage doors, either from flooding or water intrusion in the case of a hurricane, are also interpreted as major damage.
- Non-major damage (hereafter, NMD): Images that are not MD are NMD. No damage, or minor damage, such as cracked, curling, lifted, or missing shingles, missing flashing, or dents on the doors, are all considered as NMD.

Samples of images defined as MD and NMD are shown in Figs. 4.5a and 4.5b, respectively.

Classifiers used in the pre-event stream

The sequence of steps used to perform the pre-event data analysis stream is shown in, Fig. 4.6. In the pre-event stream, multiple external views of each building, collected before the event, are required. I employ an automated method I previously developed to extract suitable pre-event residential building images from typical street view panoramas [44,90].

I design three independent classifiers, shown in, Fig. 4.2a, to label the scenes containing each view of the pre-event target building. These classifiers detect: first floor elevation, number of stories, and construction material. To successfully train the classifiers to detect building attributes, I need a clear definition of each class. In what follows, I describe these definition in detail.

One important physical attribute of a residential building is first floor elevation, which is defined as the elevation of the top of the lowest finished floor, which must be an enclosed

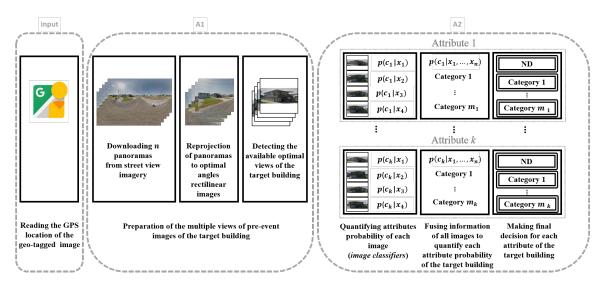


Fig. 4.6.: Detailed steps in the pre-event data analysis stream.

area, of a building. I train a classifier to determine whether a single building image should be classified as "Elevated" or "Non-elevated". The *Elevation classifier* is defined as:

- Elevated (hereafter, EL): This class includes buildings with a first floor that appears to be elevated more than 5 feet (or, half a story). Buildings are considered as EL when their ground floor, below the first finished floor, is not covered by walls or cladding and is thus visually distinguishable from an occupied floor. The lack of coverings or walls is present to potentially allow water to pass through in case of flood to reduce hydrodynamic impact loads. In a typical elevated building, the first floor only contains supporting columns (sometimes referred to as slits) which are visually identifiable in the images. Fig 4.7a shows samples of EL images.
- Non-elevated (hereafter, NEL): This class has the opposite meaning as the elevated class. It includes images of buildings without first floor elevation, or with a first floor elevation of less than 5 feet. Any images of buildings with a first floor that is covered by walls or cladding are classified as NEL. Fig (4.7b) shows samples of NEL images.

Another useful physical attribute is the number of stories. Because I focus on residential buildings here, the vast majority of the images will contain buildings that have either one or two stories. So, I train a two-class classifier to classify each of the images as either as "One-story" or as "Two-stories." This classifier does not consider any floors that are not visible, for instance in a case where a floor may be below grade. This classifier is the *Number-of-stories classifier*, and these two classes are defined as follows:

- One-story (hereafter, 1S): This class includes images of buildings which can be considered to have one-story from a structural engineering point of view (i.e., dynamically, it behaves like a single story). If any elevation is present in the image, it must not be enough to be classified as EL (i.e., less than about half a story). Fig 4.8a shows samples of *One-story* images.
- Two-stories (hereafter, 2S): This class includes images of buildings which can be considered to have two-stories, from the structural engineering viewpoint. Either a

two story building with no first floor elevation, or a one story building with greater than 5 feet of elevation at the first floor is included in the *Two-stories* category. Fig 4.8b shows samples of *Two-stories* images.

The third classifier applied to the pre-event images is trained to detect the construction material of the building. In a preliminary survey, it is important to know if wood is the main construction material, or if there is an abundance of other materials present, for instance masonry structural components or veneers. Based on the common construction practices in this geographical region, wood is the main material used for residential construction [91,92]. The *Material classifier*, distinguishing between "Wood" and "Masonry," is defined as:

- Wood (hereafter WO): Images in this class provide visible evidence that wood is the main construction material in the building. Note that all materials may not be visible in each image (or even in any image). If all visible parts of the building in the image, including columns, posts, roof structure, exterior load-bearing walls, beams, and girders, are made of wood, the image is classified as WO. Fig 4.9a shows samples of WO images.
- Masonry (hereafter, MA): When more than 70% of the visible portions of the exterior of the building in the image consists of masonry, the image is classified as MA. Fig 4.9b shows samples of *Masonry* images. Note that sloped roof buildings with masonry walls generally have wooden roofs.

4.1.2 Information fusion

I discuss how to make decisions using a probabilistic approach that fuses the classification results from several images. Let C be the random variable (r.v.) corresponding to a given physical building attribute taking values in the set C. Now consider n images x_1, \ldots, x_n of the same building and let C_1, \ldots, C_n be the set of r.v.'s corresponding to the detection of the physical attribute each one of the images. The C_i 's also take values in C,



Fig. 4.7.: Samples of images classified as Elevated and Non-elevated building images.



Fig. 4.8.: Samples of images classified as (a) One-story and (b) Two-stories.



Fig. 4.9.: Samples of images classified as (a) Wood and (b) Masonry.

but they are distinctly different. The former, C_i , only tells us which attribute was detected in image i, whereas the latter, C, which attribute was detected in the entire building. The two are different because an attribute may not be visible in all images. Since C_i depends only on the i-th image, I have:

$$p(C_i = c_i | x_1, \dots, x_n) = p(C_i = c_i | x_i) := f_{\text{CNN}, c_i}(x_i), \tag{4.1}$$

where $f_{\text{CNN},c}(x)$ is the CNN-based classifier corresponding to the attribute. How can I use the classification of each image (C_i) to classify the entire building (C)? I have:

$$p(C = c | x_{1}, ..., x_{n}) = \sum_{c_{1},...,c_{n} \in C} p(C = c | C_{i} = c_{1}, ..., C_{n} = c_{n}, x_{1}, ..., x_{n}) \cdot$$

$$p(C_{1} = c_{1}, ..., C_{n} = c_{n} | x_{1}, ..., x_{n})$$

$$= \sum_{c_{1},...,c_{n} \in C} p(C = c | C_{i} = c_{1}, ..., C_{n} = c_{n}) \cdot$$

$$p(C_{1} = c_{1}, ..., C_{n} = c_{n} | x_{1}, ..., x_{n})$$

$$= \sum_{c_{1},...,c_{n} \in C} p(C = c | C_{i} = c_{1}, ..., C_{n} = c_{n}) \cdot$$

$$\prod_{i=1}^{n} p(C_{i} = c_{i} | x_{1}, ..., x_{n})$$

$$= \sum_{c_{1},...,c_{n} \in C} p(C = c | C_{i} = c_{1}, ..., C_{n} = c_{n}) \cdot$$

$$\prod_{i=1}^{n} p(C_{i} = c_{i} | x_{i}) .$$

$$(4.2)$$

Here, going from the first to the second step I assumed that the raw data x_1, \ldots, x_n do not provide any additional information about the building label C if image labels C_1, \ldots, C_n are known. This assumption is discussed again in Sec. 4.1.2. For the next steps, I use the sum rule of probability, and observe that the C_i 's are independent conditional on the images, and then apply Eq. (4.1), w The term $p(C = c|C_1 = c_1, \ldots, C_n = c_n)$ gives the probability that the target building is labeled c, given the available images are labeled as c_1, \ldots, c_n . This *fusion* probability is attribute-specific, as discussed in Secs. 4.1.2 and 4.1.2 for post-event and pre-event attributes, respectively. Note that, in this case, the set of possible classes C always contains two elements. Without loss of generality, in what

follows, I am going to denote it with $C = \{0, 1\}$ with c = 1 corresponding to the positive detection of an attribute and c = 0 to detection of the alternative.

Finally, let \mathcal{D} be the set of possible decisions that are available to us with regard to a given building, and one void class, here called No Decision (ND), added to skip making a decision when a confident decision is not available. For example, in case of predicting the overall damage condition, it will include MD, NMD and ND. Define a loss function denoted $\ell(d,c)$ which represents the resulting loss if I choose decision d in \mathcal{D} when the true attribute is c in C. Ignoring risk preferences, the rational decision is the one minimizing the expected loss:

$$d^{*}(x_{1},...,x_{n}) = \arg\min_{d \in \mathcal{D}} \sum_{c \in C} \ell(d,c) p(C = c | x_{1},...,x_{n}).$$
 (4.3)

Here, the loss represents the threshold for making a decision about the building or leaving it as ND. The loss function parameters can be tuned by the reconnaissance teams for a specific reconnaissance goal, such as to either make the best possible decision about all cases, or to make decision only when it is highly confident. The loss function is structured to handle the trade-off between the accuracy and informativeness of the results through adding ND class to skip making a decision in case of not being sufficiently confident.

Post-event

The case of the post-event stream, and in particular the MD (C=1) vs NMD (C=0) problem, is inherently asymmetric. On one hand, one must consider the whether or not the set of images shows the building from all sides. For example, a single image classified as NMD is not sufficient to conclude that the building is indeed NMD since the damage may simply not be visible from the viewpoint of that image. So, to classify a given building as NMD, I need to ensure that all sides of the building are shown in the set of images (in this case, I say that *the building is covered*). If all of these individual images are classified as NMD, only then can the building be categorized as NMD. On the other hand, to classify a building as MD, it is sufficient to have a single image classified as MD.

Define a binary r.v. Z taking values $\{0,1\}$ indicating that the building is not covered and is covered, respectively. Let $p(Z=1|C_1=c_1,\ldots,C_n=c_n,x_1,\ldots,x_n)$ be probability that the available images sufficiently cover the target building, hereafter *coverage probability*. This dataset does not provide any information about Z (the images do not include sufficient geolocation information). Therefore, I may write:

$$p(Z = 1 | C_1 = c_1, \dots, C_n = c_n, x_1, \dots, x_n) =$$

$$p(Z = 1 | C_1 = c_1, \dots, C_n = c_n) = q_n,$$
(4.4)

where in the last step I used the observation that only the number of images are affects the state of knowledge about Z, i.e., the labels themselves are uninformative about Z. Obviously, $q_1 = 0$ and $q_2 = 0$ since one or two images cannot cover the building. Furthermore, I should have that $0 \le q_i \le q_{i+1} \le 1$. The specific numerical choice of this series of probabilities depends on the state of knowledge about how the data were collected. For example, if I knew that any three images cover the building, then I would set $q_1 = q_2 = 0$ and $q_n = 1$ for $n \ge 3$.

Now, I use the sum rule on the fusion probability:

$$p(C = 1 | C_1 = c_1, ..., C_n = c_n) = p(C = c | C_1 = c_1, ..., C_n = c_n, Z = 1)$$

$$\cdot p(Z = 1 | C_1 = c_1, ..., C_n = c_n)$$

$$+ p(C = 1 | C_1 = c_1, ..., C_n = c_n, Z = 0)$$

$$\cdot p(Z = 0 | C_1 = c_1, ..., C_n = c_n)$$

$$= p(C = 1 | C_1 = c_1, ..., C_n = c_n, Z = 1)q_n$$

$$+ p(C = 1 | C_1 = c_1, ..., C_n = c_n, Z = 0)(1 - q_n).$$
(4.5)

The two terms that I need to specify are the probabilities of labeling the building as MD (C = 1) given the image labels and whether or not the building is covered. For the covered case, I set:

$$p(C=1|C_1=c_1...,C_n=c_n,Z=1) = \left[\frac{\sum_{i=1}^n c_i}{n}\right],$$
(4.6)

where $\lceil \cdot \rceil$ is the first integer greater than its argument. This means that there is at least one image labeled as MD, then the entire building is labeled MD. For a covered building to be labeled NMD, all images must be labeled NMD. There are no intermediate cases. For the uncovered case, I set:

$$p(C = 1 | C_1 = c_1, \dots, C_n = c_n, Z = 0) = \max\left\{ \left[\frac{\sum_{i=1}^n c_i}{n} \right], \theta_n \right\},$$
(4.7)

where θ_n represents the probability that the building is MD but the damage is not visible in n images. Again, θ_n depends on what I know about data collection. In general, I must have $0 \le \theta_i \le \theta_{i+1} \le 1$. In the case studies, I simply pick $\theta_n = 0.5$ for all n. So, for the uncovered case, a single MD labeled image is sufficient to characterize the building as MD. However, if all images are labeled NMD, there is still a probability, θ_n , that the building is MD but the damage is not visible.

Pre-event

In the pre-event stream, I detect binary physical attributes, i.e., EL vs NEL, 1S vs 2S, and WO vs MA. All these cases are similar in nature. The more often an attribute is detected in the images the more likely it is really there. The simplest model that encodes this intuition is:

$$p(C=1|C_1=c_1,\ldots,C_n=c_n)=\frac{\sum_{i=1}^n c_i}{n}.$$
 (4.8)

Here, I exploit the 0-1 encoding of the binary class. The probability on the right hand side is simply the average number of ones in the n images. Essentially, the r.v. C conditional on the r.v.'s C_1, \ldots, C_n has a Bernoulli distribution. The approach can be trivially generalized, using a Categorical distribution, to the case where C contains more than two options.

Classifier name	Learning rate	Momentum	Weight decay coefficient
Overview	1×10^{-5}	9×10^{-1}	1×10^{-6}
Damage	5×10^{-6}	9×10^{-1}	1×10^{-6}
Elevation	1×10^{-6}	9×10^{-1}	1×10^{-6}
Number-of-stories	5×10^{-7}	9×10^{-1}	1×10^{-7}
Material	1×10^{-7}	9×10^{-1}	1×10^{-7}

Table 4.1.: Hyper-parameters used to train the classifiers

4.2 Experimental validation

I verify the individual classifiers and validate the overall technique using a high-quality published and curated post-event dataset. These perishable information were captured during reconnaissance missions that took place shortly after hurricanes Harvey and Irma, led by the NSF-funded Structural Extreme Events Reconnaissance (StEER) Network, with data collection supported by the Fulcrum App [89]. CNNs-based algorithms are the most powerful image classifiers that have recently been used successfully in a wide range of applications [93,94]. I have tried three networks, Inception v3 [95], InceptionResNetV2 [96], and Xception [97], as the image classifiers, and Xception network slightly outperformed the two others. I implemented Xception with Depthwise Separable Convolutions network, in Keras [98].

In this implementation I used Stochastic Gradient Descent(SGD) optimizer. The SGD hyper-parameters used for the classifiers were fine-tuned using grid search to train each of the classifiers. I tuned the hyper-parameters, particularly the learning rate which is the most important hyper-parameter [99], carefully to improve the performance of the classifiers. I set the grid to search for 1) learning rate in $\{1 \times 10^{-1}, 5 \times 10^{-2}, \dots, 5 \times 10^{-9}, 1 \times 10^{-10}\}$, 2) momentum in $\{1 \times 10^{-1}, \dots, 9 \times 10^{-1} \text{ and } 99 \times 10^{-2}\}$, 3) weight decay coefficient in $\{1 \times 10^{-1}, \dots, 1 \times 10^{-10}\}$. I randomly separate the train and test set with 70% and 30%, respectively, of the data for each classifier. To avoid over-fitting, I randomly sample out 10% of the train set to use for hyperparameters fine-tuning. Table 4.1, shows the hyperparameters used to train these five required classifiers.

The StEER network was formed to document the damage induced and enable research to understand the effects of a series natural hazard events [100, 101], including hurricanes Harvey, Irma and Maria in 2017 [102, 103], and hurricane Florence and Michael in 2018 [104–106], on the built environment. An overview of the dataset [101, 107] is shown in, Fig. 4.10. Detailed damage surveys of more than 4,000 buildings were conducted door-to-door [89, 108]. The data include assessments of the post-event condition of most of the buildings. Other documentation includes primary structural typologies, construction materials, and certain component damage levels. The documentation available for this data also includes both building attributes plus observations of the overall damage condition of the building after the hurricane. Thus, these data are well-suited for validation of the technique developed.

For training the classifiers I used data from 3,141 buildings, including 2,020 buildings collected after hurricane Harvey in Texas, and 1,121 building collected after hurricane Irma in Florida. The data vary greatly from building to building in terms of completeness and number of images collected. Thus, not all the data collected from these 3,141 buildings are useful. I pre-process the dataset as follows. I made adjustments to the pre-event attributes documented in the original dataset that were necessary to conform with the definitions. The first floor elevation is reported as an estimated height of elevation in the original documentation. Here, I use the threshold of 5 feet to manually label the data for training, testing and validation. Then, if the building is elevated, I also add one to the number of stories reported to conform to the definition. Regarding the construction material, I make use of the attribute in the original data called structural framing. However, most of these building actually use wood for the structural framing, or the load bearing elements, and thus I redefine it as the main construction materials visible on the exterior of each building as explained in Sec. 4.1.1. When multiple items are provided in the original data, I simply use the first material listed.

Because the data I use for validation do not contain geo-location information, I only consider the number of available images (see Sec. 4.1.2). In Sec. 4.1.2, I defined the probability that n images are sufficient to cover the building as $p(Z = 1|C_1 = c_1, ..., C_n =$

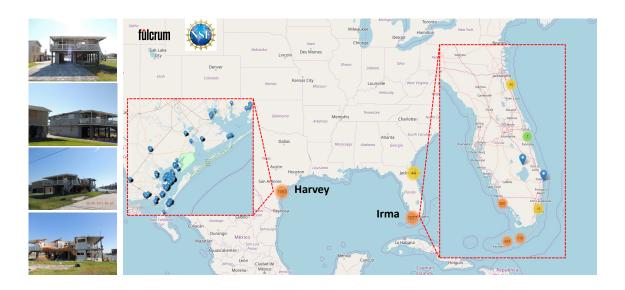


Fig. 4.10.: Post-event reconnaissance dataset collected after Harvey and Irma and published on DesignSafe-CI and Fulcrum [89, 107, 108].

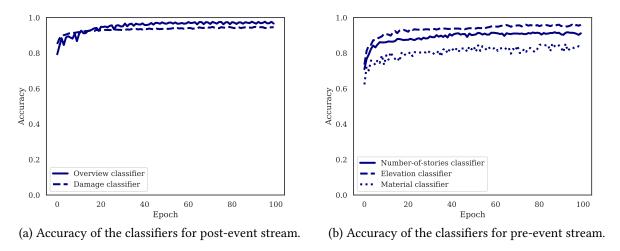


Fig. 4.11.: Accuracy plots of classifiers.

 c_n) = q_n . Currently the typical number of images captured in wind-event reconnaissance missions is quite small. Furthermore, there is a certain bias in the collection process since the data collector is, typically, interested in collecting images of damage. For example, I observe that data collectors take fewer images of buildings that have no damage or only minor damage. In these circumstances, if only one image is captured, then I may conclude that the building is sufficiently covered, i.e., $q_n = 1$ for all $n \ge 1$. In a more objective data collection process, one has to adjust coverage probability accordingly, see Sec. 4.2.1.

I evaluate the performance of the pre-event and post-event data analysis streams independently. The validation of the method involves first evaluating the performance of the individual steps in each branch (i.e., of each classifier), as well as considering the end-to-end performance of each data analysis branch. Fig.4.11a and 4.11b show only the accuracy of the classifiers used for post-event and pre-event stream, respectively. However I evaluate the end-to-end performance of the method developed in Sec. 4.2.1 and Sec. 4.2.2. The input to each branch is the set of geo-tagged raw images of the buildings. To validate each of these, I use raw available data from all of the 1,121 buildings collected after hurricane Irma. Here I explain both the post-event and pre-event data analysis streams validation results. In the post-event stream, first I demonstrate the results for an example loss func-

Table 4.2.: Loss function.

_			Decisi	on
label		ND	MD	NMD
ue l	MD	α_1	0	1
Irc	NMD	α_2	1	0

tion assuming all buildings are sufficiently covered. Then, I discuss how the results can be improved if I refine the coverage probability, q_n in Eq. 4.4. Subsequently, I study the effect of the loss function parameters on the trade-off between *accuracy* and *ND rate*, rate of ND predictions over all permissible predictions. In the post-event stream, I illustrate the results for an example loss function, and then the procedure for tuning of the loss function parameters is discussed.

4.2.1 Post-event stream validation

As described earlier, each OV post-event image is passed through the damage classifier. Predicting the overall condition of the building, based only on images, is subject to error, see Sec. 4.1.2. Even if the building is covered, it may still be difficult to make the decision based entirely on the images. For example, the damage shown in the image may not be sufficiently severe to be labeled MD, nor minor enough to confidently labeled as NMD. Under these circumstances, even human inspectors face difficulties and the situation calls for a more detailed inspection.

The general form of the loss function is shown in, Table 4.2. Without loss of generality, I can set the loss of correct predictions to zero. The cost of mistakenly characterizing an MD (NMD) building as NMD (MD) is 1. The cost of labeling as ND when the building state is MD (NMD) is α_1 (α_2). These parameters are selected to reflect the goals of the preliminary survey, see Sec. 4.2.1.

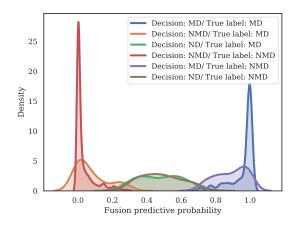


Fig. 4.12.: Density of the fusion predictive probabilities corresponding to each different decision, assuming all buildings are sufficiently covered (with $q_n = 1$ for $n \ge 1$).

Table 4.3.: Confusion matrix using a loss function with parameters ($\alpha_1 = \alpha_2 = 0.3$); assuming all buildings are sufficiently covered.

		De	cision			
		No OV	ND	MD	NMD	All
_	No label	26	6	5	17	54
abe	MD	44	16	151	39	250
rue label	NMD	109	71	71	566	817
Tr	All	179	93	227	622	1,121



Fig. 4.13.: Sample of a correct MD detection.

Sample results

First, consider the case in which all of the buildings are assumed to be captured adequately with the images available, $q_n = 1$ for all $n \ge 1$, and pick a loss function with $\alpha_1 = \alpha_2 = 0.3$. This choice of the loss function making mistakes has a unit cost, while not deciding costs thirty percent of the mistake cost. In Fig. 4.12, I visualize the density of the fusion predictive probabilities corresponding to each different decision and true label, i.e., density of decisions made at a given fusion probability. It shows six combination of the two true labels, MD and NMD, and three possible decisions, MD, NMD and ND. The correct decisions for the buildings with NMD (MD) true labels, depicted in red (blue), show low-variance right (left)-skewed density with a mode close to 0 (1). However, the densities of the incorrect decisions for both MD and NMD buildings, have more variance. Table 4.3 provides the confusion matrix, table of true labels versus predicted, for the results of the demonstration of the end-to-end post-event stream data analysis. Out of a total of 1,121 buildings visited after hurricane Irma, the dataset includes 54 buildings with no true label, and 179 buildings with no OV images. Also, 26 buildings are not distinct and those data are merged into one building set. Therefore I have 914 labeled buildings with OV images. The results show that 717 buildings are correctly categorized, 110 buildings are classified incorrectly, and 87 buildings labeled ND.

To understand the limitations of the approach, it is informative to examine some specific building examples of correct (incorrect) decisions as well as ND. Figure 4.13 shows four images of a representative case in which a building is correctly categorized as MD. In this case, the first three raw images, numbered as 1, 2, and 3, do not show any evidence of



Fig. 4.14.: Sample of a ND building categorization.



Fig. 4.15.: Sample of an incorrect building categorization.

damage. However, image number 4 does show the damage clearly, and the CNN classifies it as MD with a high probability. The fusion formula, Eq. (4.6), categorizes the building as MD with high probability.

Figure 4.14 includes six images corresponding to an ND case. The true label of the building is MD. The three images in top row, numbered as 1, 2, and 3 are each individually classified as NOV with a high probability. However, image number 4 does show signs of damage on the roof, albeit with a 51.79% probability. Images 5 and 6 do not show any evidence of damage. The fusion formula, also gives an almost fifty-fifty chance of MD.

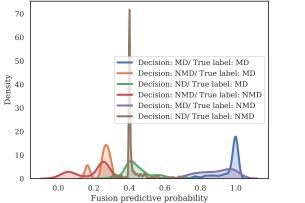
Figure 4.15 corresponds to a case that is incorrectly categorized as NMD due to a shortage of informative images. In particular, there is not an adequate number of images to cover the building (remember that in this case study I have set $q_n = 1$, i.e., the framework mistakenly "thinks" that the building is covered). Only one image (front view of the building facade, numbered 1) is classified as OV. Image number 2 shows canonical view of the building and potentially could capture the damage, but is highly obstructed by trees. Therefore, image 2 is classified as NOV and is not used for building categorization. Thus, image number 1 is the only image available for detecting the overall damage condition which does not have any evidence that the building should be categorized as having major damage, and is not classified as damaged. However, image number 3, which is the top view of the building capture through aerial imagery, which is not part of the data collected in preliminary survey, does show the damage on the back side of the building clearly. Note that this image would have been filtered out automatically by the overview classifier. It is included manually here for demonstrating the true building label. Investigating the case shown in 4.15 reveals that the need for capturing multiple post-event images that cover all around the building is critical for correct building categorization, see Sec. 4.4.

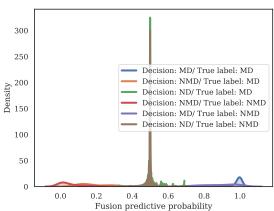
Discussion on selecting the coverage probability

The results presented in Table 4.3 are based on the assumption that each given building is sufficiently covered, and human data collectors may have taken only 1 or 2 images of the

Table 4.4.: Confusion matrix using a loss function with parameters ($\alpha_1 = \alpha_2 = 0.3$); considering a sample coverage probability, $q_1 = 0.2, q_2 = 0.5, q_3 = 0.9, q_n = 1$ for $n \ge 4$.

Decision						
		No OV	ND	MD	NMD	All
-	No label	26	19	5	4	54
abe	MD	44	45	151	10	250
rue label	NMD	109	355	71	282	817
Tru	All	179	419	227	296	1,121





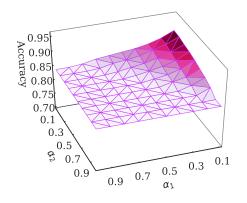
(a) Fusion probability with coverage probability of q_1 =(b) Fusion probability with coverage probability of q_1 = 0.2, q_2 = 0.5, q_3 = 0.9, q_n = 1 for $n \ge 4$. $q_2 = q_3 = 0$, $q_n = 1$ for $n \ge 4$.

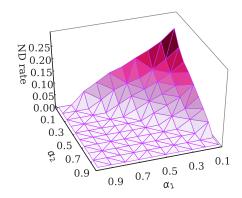
Fig. 4.16.: Fusion probabilities.

NMD buildings. However, the method is capable of dealing with unbiased data collected automatically. This is possible through proper setup of $p(C = c | C_1 = c_1, ..., C_n = c_n)$, introduced in Sec. 4.1.2. In Table 4.4 I illustrate the results of considering a sample coverage probability, $q_1 = 0.2$, $q_2 = 0.5$, $q_3 = 0.9$, $q_n = 1$ for $n \ge 4$. The results in Table 4.4 show that the number of MD buildings which are incorrectly characterized as NMD is reduced by almost 75%, compared with Table 4.3. These building are moved to the ND class. For example, the case discussed in Fig. 4.4 is characterized as ND after modifying the coverage probability. Figure 4.16a shows the density of the fusion predictive probabilities corresponding to different decision. However, since one or two images are deemed insufficient to consider the building covered, the number of correctly detected NMD buildings also decreases by about 50%, and again these are moved to the ND class. These consequences of incorporating coverage information can be interpreted as an indication that human data collectors typically have an inherent bias to take fewer images of buildings with no or minor damages, or NMD buildings. The human collectors see things that are not depicted in the images they take. For future utilization of this method, assuming the collected dataset contains more images of the target buildings, it is recommended to use realistic choice of coverage probability, e.g., $q_1=q_2=q_3=0, q_n=1$ for $n\geq 4$. Density of the fusion predictive probabilities corresponding to different decisions are depicted in Fig. 4.16b

Discussion on tuning the loss function

In Tables 4.3 and 4.4, the ratio of the correct, incorrect and ND prediction is highly dependent on the loss function parameters. The choice of these parameters should reflect the objectives of the reconnaissance team. To develop some intuition about these parameters, I investigate their effect on the results, I change α_1 and α_2 from 0.1 to 1 and calculate the results for all combination sets of the parameters. Figure 4.17a demonstrates the effect of loss function parameters on the *accuracy* of the post-event buildings overall damage categorization.





- (a) Loss function parameters on the accuracy.
- (b) Loss function parameters on the ND rate.

Fig. 4.17.: Accuracy vs ND rate.

According to, Fig. 4.17a, decreasing both the parameters α_1 and α_2 , results in higher accuracy. However according to, Fig. 4.17b, decreasing α_1 and α_2 , results in a high *ND rate*, rate of ND predictions over all permissible predictions. To explain it more clearly, I describe two scenarios corresponding to two teams with different goals. The first scenario refers to a team that has limited but sufficient resources to visit all potential MD buildings, and prefers to not miss any of the MD buildings. In this scenario, high accuracy is not critical, albeit they want avoid a high ND rate which may lead to missing some MD cases. They can encode this objective in the loss function by picking the α_1 and α_2 very high, e.g., 0.9. The second scenario refers to a team that has a limited resources and prefers to spend it more conservatively and only visit the buildings that have high probability of falling into MD category. In this scenario, the goal is to increase the accuracy, however, having high ND rate is not a big concern. They can encode this objective by picking the α_1 and α_2 very small, e.g., 0.1.

4.2.2 Pre-event stream validation

In the pre-event stream, images of 807 of the 1,121 buildings visited after hurricane Irma are successfully extracted from street view panoramas. The 314 buildings excluded from the pre-event images extraction are not available because (1) the building's address is not available, (2) the street view panoramas are not available, (3) the building facade maybe occluded by other objects, e.g., trees, cars or other buildings, (4) in some geographical regions street view images are not up to date and have a very low resolution. So I set the pre-event image extraction tool to filter out those images. Here, all of these 807 buildings are assumed to be captured adequately with the images available.

The general form of the loss function for determining pre-event attributes is shown in, Fig. 4.5. Similar to the post-event loss function, the loss of a correct prediction is set to zero, but the loss of making mistakes, 1, or labeling as ND, α_1 and α_2 , represents the relative penalties.

Table 4.5.: Loss function.

-			Decisio	n
ıe label		ND	Attribute 1	Attribute 2
le]	Attribute 1	α_1	0	1
Tru	Attribute 2	$lpha_2$	1	0

Table 4.6.: Confusion matrix of first floor elevation using a loss function with parameters ($\alpha_1 = \alpha_2 = 0.3$).

			Decision				
-		ND	Elevated	Not Elevated	All		
abe	Elevated	111	136	35	282		
rue labe	Not Elevated	143	20	362	525		
Tr	All	254	156	397	807		

Table 4.7.: Confusion matrix of number of stories using a loss function with parameters ($\alpha_1 = \alpha_2 = 0.3$).

	Decision				
		ND	One	Two	All
7	One	137	226	34	397
rue label	Two	67	19	209	295
ıe l	Unknown or more than Two	16	12	87	115
Tr	All	220	257	330	807

Table 4.8.: Confusion matrix of construction material using a loss function with parameters ($\alpha_1 = \alpha_2 = 0.3$).

			Decision		
		ND	Masonry	Wood	All
-	Masonry	27	102	10	139
ue label	Wood	119	28	116	263
le J	Unknown or Others	164	136	105	405
Trc	All	310	266	231	807

Tables 4.6, 4.7, and 4.8 provide the confusion matrix for the results of the demonstration of the end-to-end, pre-event stream data analysis. These results are obtained with a loss function with $\alpha_1 = \alpha_2 = 0.3$. Table 4.6 provides the confusion matrix for the results of the demonstration of the end-to-end, pre-event stream data analysis for first floor elevation attribute. Out of a total of 807 buildings, 498 buildings in the dataset posted are correctly categorized, 55 buildings are classified incorrectly, and 253 buildings labeled ND. Table 4.7 provides the confusion matrix for the results for number of stories attribute. Out of a total of 807 buildings, 115 buildings in the posted dataset have an unknown or more than two stories true label. Therefore data from the 692 one and two story labeled buildings are used here. The results show that 435 buildings are correctly categorized, 53 buildings are classified incorrectly, and 204 buildings labeled ND. Table 4.8 shows the confusion matrix for the results for construction material attribute. Out of a total of 807 buildings, 405 buildings have unknown or other types of material, and 402 buildings are labeled as either wood or masonry buildings. Out of these 402 buildings, the automated data analysis procedure results show 218 buildings are correctly categorized, 38 buildings are classified incorrectly, and 146 buildings are labeled ND.

5. AUTOMATING DATA-DRIVEN PROBABILISTIC VULNERABILITY MODEL GENERATION

Probabilistic vulnerability models are a key component of predictive models.

The term "vulnerability model" refers to the conditional probability of an undesirable phenomenon, e.g., an specific level of damage, given the intensity of an excitation, e.g., wind or seismic load [109]. Vulnerability models are classified into three categories: empirical models, models based on experts judgment, and analytical models [85].

One category of the vulnerability models entirely relies on the judgment of the experts. The designated are estimating the probability of exceeding certain levels of damage of an element at risk for different given levels of loading. [85]. For example, the vulnerability models of the roads and tunnels used in HAZUS for are developed using the expert method described in the ATC-13 documents [110, 111].

Analytical methods to develop these functions are typically based on simulations. In these methods, the damage is estimated based on the simulated response of the element at risk under various loadings. Due to the necessary levels of complexity and accuracy, several analytical methodologies have been developed [41,112–114].

Empirical vulnerability models are based on observational data of the real-world events [39]. Empirical methods benefit from the real event data and can account for various local effects, e.g., construction quality and geographical site impacts. In case of an earthquake, this may be interpreted as a shortcoming that the empirical vulnerability models remain specific to a given region with particular conditions and event parameters. However, in the case of an event like a hurricane that may happen with a reasonably short return period in the same/similar geographical region, the empirical vulnerability models can take into account the associated uncertainties with the site and event [85, 110].

The resulting prediction of damage or loss form different selection of vulnerability models would deviate considerably, and likely bias the decision-making. Currently, there is no consensus concerning an ideal method as all the described approaches have their strengths and shortcomings [41,110]. Further, the available vulnerability models are validated merely through comparison with other existing models or against relatively small real-world datasets. The outcome of analysis could be highly biased when the insufficiently validated vulnerability models are used in real-world applications [41]. This gap in validation originated from the burden of on-site real-world data collection and data cleaning. Due to the considerable improvement of data collection platforms and computational resources, this burden can be considerably mitigated through automating the bottlenecks in the process.

The objective here is to develop an automated algorithm for hurricane loss prediction to support the national, state and local planners and decision-makers. The contribution here is to automate the process of empirical vulnerability generation using postevent building images. Using a dataset that was collected after hurricane Harvey in 2017, I generate input-output pairs for each observed asset consisting of structural and non-structural characteristics before the hurricane (input) and its damage state after the hurricane (output). Using these input-output pairs, I create a vulnerability model that predict damage level of structures. Taking advantage of the automated data preparation techniques [60,88], the credibility and comprehensiveness of the empirical vulnerability model can be improved as I collect more data over the newly occurred events.

This vulnerability model is generated based on the past data can be used to plan for future events. I generate synthetic events scenarios using the hazard maps of the region, and use the vulnerability model to predict the probability of a specific damage level conditional on the asset characteristics and the intensity of the event. Each damage level is associated with an approximated percentage of the building net value. I estimate the net value of the assets using commercial real estate tools, e.g. Zestimate [115], to predict the regional loss.

The remainder of this chapter is structured as follows. Section (5.1) presents the general methodology and problem formulation. Section (5.2) provides an illustrative example of hurricane Harvey to demonstrate and validate the methodology and discuss the results.

5.1 Methodology

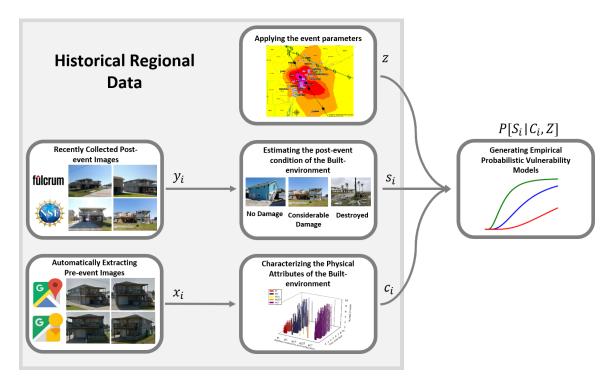


Fig. 5.1.: Schematic of the process developed to generate empirical vulnerability models

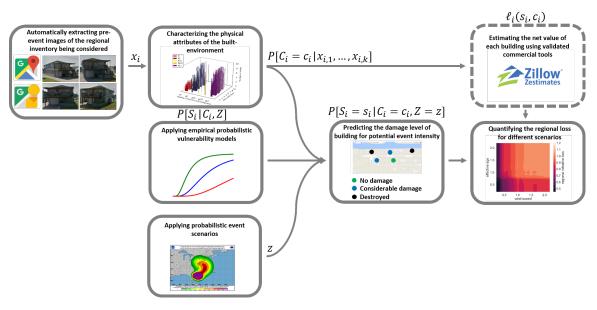


Fig. 5.2.: Predicting regional loss of a future event with a regional inventory and potential range of event intensity

Figure 5.1 demonstrates the process of generating probabilistic vulnerability models, I provide the details in Sec. 5.1.1. Figure 5.2 visualizes the application of the vulnerability models for regional loss prediction in hypothetical future events, see Sec. 5.1.2 for the details. In the phase of vulnerability model generation I am using the historical regional data, and in the loss prediction phase I am using the data of the regional inventory being considered.

5.1.1 Automated vulnerability model generating

I use the common mathematical notation in which random variables (r.v.) are indicated by upper case whereas their values by the same lower case letters. For example, the variable X represents a r.v. and the variable x a possible value of X. With p(X = x) I denote the probability mass function of X, if X is discrete, or the probability density function (PDF), if X is continuous. When there is no possibility of ambiguity, I am going to write p(x) instead of p(X = x). Abusing a little bit the terminology, I will be referring to p(x) as the "probability" of x irrespective of whether the r.v. X is discrete or continuous. If Y is another random variable, then p(X = x, Y = y) (or p(x, y)) will denote the joint PDF of X and Y and p(X = x|Y = y) will denote the PDF of X conditioned on observing Y = y.

Let $i=1,\ldots,n$ represent the index of assets in a built environment with n assets. The characteristics of asset i are denoted by C_i . These characteristics could be structural, e.g., number of stories and first floor elevation, or non-structural, e.g., floor area and zip code. The characteristics C_i are random variables because they are not directly observable, i.e., there is epistemic uncertainty regarding their value. With the nonnegative r.v. Z I capture the event intensity, e.g., maximum wind speed or inundation depth. The discrete r.v. S_i represents the damage level of asset i after an event. I assume that S_i takes values in $\{0,1,2\}$ that $\{0,1,2\}$ corresponding to no damage, considerable damage, and destroyed, respectively. Using the notation, the conditional probability that building i is at damage

level $S_i = s_i$ after a hazardous event with intensity Z = z is denoted by $p(S_i = s_i | C_i = c_i, Z = z)$ or, more simply, by $p(s_i | c_i, z)$. The function $p(s_i | c_i, z)$ is the vulnerability model.

The preliminary steps to automatically generate the vulnerability models are discussed in [60, 88]. In [60] I developed the algorithm to automatically extract multi-view pre-event images of the built environment from street-view imagery. In [88] I describe a methodology for inferring the physical attributes C_i , using k_i^{pre} pre-event images denoted as $x_i^{\text{pre}} = \left(x_{i,1}^{\text{pre}}, \dots, x_{i,k_i^{\text{pre}}}^{\text{pre}}\right)$, and the damage level S_i of the buildings, using k_i^{ost} post-event images denoted as $x_i^{\text{post}} = \left(x_{i,1}^{\text{post}}, \dots, x_{i,k_i^{\text{post}}}^{\text{post}}\right)$. These procedures essentially yield the posterior probabilities $p(c_i|x_i^{\text{pre}})$ and $p(s_i|x_i^{\text{post}})$.

5.1.2 Regional loss prediction

The direct monetary loss that building i with characteristics of c_i that experience the damage level s_i , denoted as $l(s_i, c_i)$.

Then, to quantify the future regional loss of an inventory under consideration L, I have:

$$L(Z = z) = \sum_{i=1}^{N} \sum_{s \in S} \sum_{c \in C} \ell_i(s_i, c_i) \cdot$$

$$p(S_i = s_i | c_i, Z = z) p(C_i = c_i | x_{i,1}, \dots, x_{i,k}), \quad (5.1)$$

where the $p(C_i = c_i | x_{i,1}, ..., x_{i,k})$ can be determined through fusing the extracted information of multiple pre-event images to detect the characteristics of the building [60,88]. Here I briefly describe this step which are used both in vulnerability generation and regional loss prediction phases. I consider k images $x_{i,1}, ..., x_{i,k}$ of the building i and let $C_{i,1}, ..., C_{i,k}$ be the set of r.v.'s corresponding to the detection of the physical attribute each one of the images. The $C_{i,j}$'s also take values in C, but they are distinctly different. The former, $C_{i,j}$, only tells us which attribute was detected in image j, whereas the latter, $C_{i,j}$, which attribute was detected in the entire building. The two are different because

an attribute may not be visible in all images. Since $C_{i,j}$ depends only on the j-th image, I have:

$$p(C_{i,j} = c_{i,j}|x_{i,j}) := f_{\text{CNN},c_{i,j}}(x_{i,j}), \tag{5.2}$$

where $f_{\text{CNN},c}(x)$ is the CNN-based classifier corresponding to the attribute. How can I use the classification of each image $(C_{i,j})$ to classify the entire building (C_i) ? I have:

$$p(C_{i} = c_{i} | x_{i,1}, \dots, x_{i,k}) = \sum_{c_{i,1}, \dots, c_{i,k} \in C} p(C_{i} = c_{i} | C_{i,j} = c_{1}, \dots, C_{i,k} = c_{i,k}) \cdot \prod_{i=1}^{k} p(C_{i,j} = c_{i,j} | x_{i,j}), \quad (5.3)$$

The $l_i(s_i, c_i)$ can be estimated using commercial tools, e.g. Zestimate.

5.2 Illustrative Example

I use the real-world data collected post hurricane Harvey reconnaissance mission by StEER Network to demonstrate and validate the empirical vulnerability model generation. The validation of the presented technique can be break down in three core parts: 1) automated exposure detection, 2) Assets net value estimation, and 3) empirical predictive vulnerability model generation. Part 1 is already demonstrated and validate in [60,88]. Part 2 is relying on the commercial tools that are already validated. Here I focus on demonstration and validation of the part 3.

The post hurricane Harvey dataset contains the key characteristics of the 1988 buildings in Texas. Figure 5.3 shows the statistics of the data. The *construction material*, Fig. 5.3a, the *occupancy group*, Fig. 5.3b, and the *number of stories*, Fig. 5.3c, of the collected data is highly unbalanced. I develop predictive vulnerability models only for the intersection of the dominant categories, wood structure and single family residence, and one and

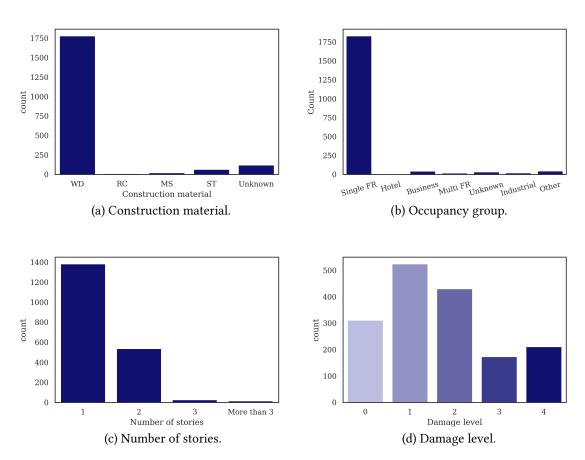
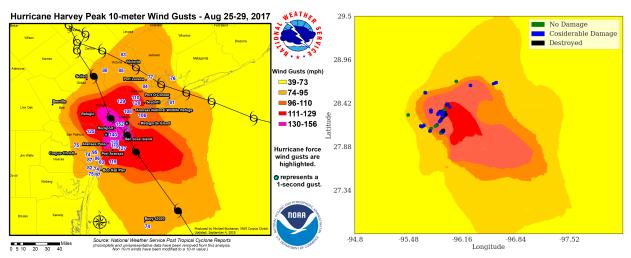


Fig. 5.3.: Taxonomy of the raw data collected after hurricane Harvey [89, 101, 107].



(a) Original wind map of the hurricane Harvey published(b) Geographical distribution of the visited buildings by by National Weather Service.

Steer team on the retrieved wind map.

Fig. 5.4.: Assigning the approximate wind speed to the building inventory using the hurricane Harvey wind map [116].

Damage state	Qualitative damage description	Roof	Window	Roof deck	Missile	Roof	Wall
		cover failure	door failures		on walls	structure failure	failure
0	No damage or very minor damage Little or no visible damage	$\leqslant 2\%$	No	No	No	No	No
	from the outside. No broken windows, or failed roof deck. Minimal loss of						
	roof over, with no or very limited water penetration.						
1	${f Minor\ damage}$ Maximum of one broken window, door, or garage door.	>2%	One	No	< 5	No	No
	Moderate roof cover loss that can be covered to prevent additional water	and	window,		impacts		
	entering the building. Marks or dents on walls requiring painting or	$\leqslant 15\%$	door, or				
	patching for repair.		garage door				
			failure				
2	Moderate damage Major roof cover damage, moderate window	>15%	> one and	1 to 3	Typically	No	No
	breakage. Minor roof sheathing failure. Some resulting damage to interior $% \left(1\right) =\left(1\right) \left(1\right) $	and	\leqslant the	panels	5 to 10		
	of building from water.	$\leqslant 50\%$	larger of		impacts		
			20% and 3				
3	${\bf Severe} \ \ {\bf damage} \ {\tt Major} \ {\tt window} \ {\tt damage} \ {\tt or} \ {\tt roof} \ {\tt sheathing} \ {\tt loss}. \ {\tt Major}$	>50%	> the	> 3	Typically	No	No
	roof cover loss. Extensive damage to interior from water.		larger of	and	10 to 20		
			20% and 3	$\leqslant 25$	impacts		
			and $\leqslant 50\%$	%			
4	$\mathbf{Destruction}$ Complete roof failure and/or failure of wall frame. Loss of	Typically	>50%	> 25	Typically	Yes	Yes
	more than 50% of roof sheathing.	>50%		%	> 20		
					impacts		

Fig. 5.5.: Damage states for residential construction classes [117]

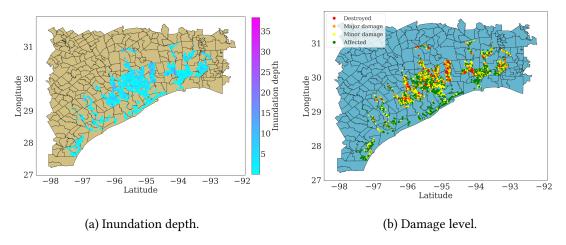


Fig. 5.6.: Visualization of the 156,099 buildings visited by FEMA after hurricane Harvey [118].

two stories buildings. Filtering out the minority categories, in total I use 1649 building samples.

Figure 5.3d demonstrate the distribution of the damage level, define by [117], in the dataset. Referring to this definitions, here I define the recovery cost into three levels of: a) 0% of the net value corresponded to damage level 0, b) 50% of the net value corresponded to damage levels 1,2,and 3, c) 100% of the net value corresponded to damage level 4.

Typically hurricanes produce heavy precipitations. Wind speed and inundation depth are the hazard parameters to be considered for loss prediction. However the dataset does not include the wind speed and inundation depth. I need to assign the estimated values from other available sources using the geo-spatial information. Estimating the wind speed after a hurricane is a complicated task [119,120] and typically are presented in wind map contours [116]. Due to inaccessibility of the original wind speed data, I retrieved this information from the hurricane Harvey wind map contours published by National Weather Service [116], and then assigned the estimated wind speed to each building, See Fig. 5.4.

An inundation depth of more than 156,000 buildings are estimated by FEMA [118]. However the visited buildings by FEMA has no overlap with the buildings visited by StEER team. To estimate the inundation depth of the buildings in StEER data, I find the closest building visited by FEMA for each building, See Fig. 5.6.

To find the conditional probability of $\mathbb{P}[S_i = s_i | c_i, Z = z]$, I use the extreme gradient boosting (XGBoost) algorithm [121, 122] to train the vulnerability model in a form of a classifier. I implement a grid search and compute the cross-validation loss function for each combination of the hyperparameters, e.g., learning rate, maximum depth, minimum child weight, gamma, lambda, alpha, sub-sample, column sample.

The receiver operating characteristic (ROC) curve is a performance metric demonstrating the relationship between true positive rate and false positive rate for different threshold settings. The ROC curve is typically used to investigate the effect of threshold setting for binary classification problems. Although the problem is a multi-class classification I binaries each class to demonstrate the performance of model for each class. The ROC curve of each class and the micro and macro average, Fig. 5.7a, demonstrates the

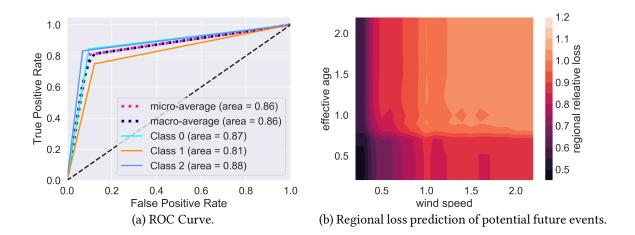


Fig. 5.7.: Validation and prediction results.

promising result of the classifier. However, the more data of the real-world event I collect I can improve the credibility of the vulnerability models for wider range of events and geographical regions.

Here I demonstrate how this technique can be used to support planning for the future events using a sample of 675 buildings. To keep the demonstration simple I assume the net value of the all buildings are equal to 1. However for a real application I can use the commercial tools, e.g., Zestimate that is provided by Zillow Inc., to estimate the net value for the buildings more realistically. Considering the true labels of the buildings collected by field engineers after hurricane Harvey, the total recovery cost of the 675 buildings sample is 332.5. To predict the loss for future events I generate synthetic data with multiplying a variational ratio to the event parameters, e.g., wind speed, and the key characteristics of the buildings, e.g., effective age, and calculate the total relative loss, total predicted loss divided by estimated loss using true labels. Figure 5.7b showes the variation of the predicted relative total loss of the sample set of buildings with respect to multiplied ratio to the wind speed and the effective age of the buildings. One application of this analysis is that the decision makers and insurance adjusters can evaluate the effectiveness of retrofitting the buildings, decreasing the effective age, of the buildings on the projected regional loss in various possible event scenarios.

6. SUMMARY AND CONCLUSION

Communities aiming to be resilient need tools at their disposal that empower them to both respond to and prepare for extreme events. With minimal inventory data, communities can set desired objectives, and plan based on those desired objectives to cope with future disruptive events. One of the strategic decisions to prepare for extreme events is to determine an appropriate budget, based on desired objectives, that the community can set aside to prepare for building inspections after a potential future event. Here an approach and the associated computational techniques are developed that can be used for various types of disruptions, and with varying levels of detailed data. This approach is broken down into four modules, as shown in Fig. 6.1.

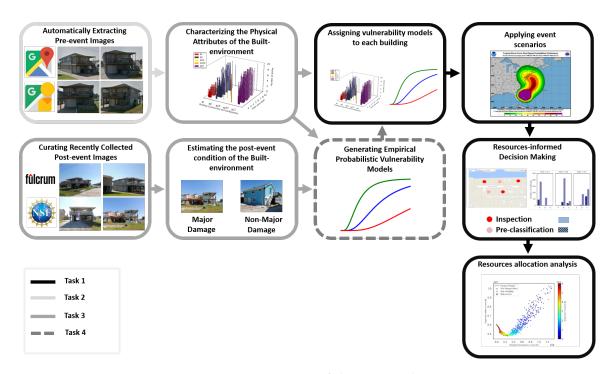


Fig. 6.1.: Overview of the approach.

The method developed in module 1, discussed in chapter 2, is intended for rapid and effective planning of building inspections while also minimizing the expected cost of this process to the community. The key idea behind prioritizing the structures for inspection is that the post-event safety level of some structures can be predicted reliably using available information. These predicted safety levels can be adopted with minimum consequences for the community. This approach goes beyond past projects focused on developing urban or regional risk models. This approach is relatively simple and supplies necessary guidance for decisions in the case of a disruptive event. Furthermore, this study represents an example of how the typical outcome of numerous urban or regional risk analysis initiatives can be used in practice, as well as assessing the impact of using eventually inaccurate information of that sort.

This approach is illustrated using a realistic building inventory, manually collected data, to demonstrate its use to determine field inspection priorities for hypothetical events. Not all of the preliminary data to implement this approach may be readily available in all communities. Two other techniques are developed here to prepare the required preliminary information. Both of the associated techniques are fully automated, which enables communities around the world to readily use this approach.

The first automated technique which is developed as module 2, discussed in chapter 3, presents an automated technique to extract pre-event building images from street view panoramas, publicly available online. The technique developed herein automates the process of extracting building images from existing street view panoramas to support the building reconnaissance process. Once an approximate location of a buildings are provided, several high-quality external views of the entire building are rapidly generated for use. The performance of the technique developed here is successfully demonstrated using actual reconnaissance images as well as Google Street View panoramas collected from Holiday Beach and Rockport, Texas, which suffered significant damage during Hurricane Harvey in 2017. These images were collected by field investigators during a post-disaster reconnaissance mission. All of the pre-disaster images, corresponding to the house in each post-disaster image, are successfully extracted from Google Street View panoramas.

The technique represents a promising example of how to easily and automatically add value to both newly collected and existing (legacy) volumes of visual data.

The second automated technique which is developed as module 3, discussed in chapter 4, is focused on categorizing buildings based on their key physical attributes and assess their post-event overall condition. This technique is intended to use pre-event and postevent images and rapidly provide the reliable and sufficiently comprehensive information required to support inspection planning. The input to the technique is a collection of post-event images collected from residential buildings in the effected region. The output of the technique is the building attributes, and the damage classification for the buildings in that region. The technique incorporates several steps designed to automate this process. The technique is divided into pre-event and post-event streams, each intending first to extract all possible information about the target buildings using both pre-event and post-event images. The technology enabling this research is convolutional neural network algorithms (CNNs) that are implemented for scene (image) classification. A hierarchical classification schema is designed to organize the data. Robust scene classifiers are designed for specific scene classification tasks. Information fusion methods are developed to combine the results from multiple images, yielding a result that collectively considers the individual results of multiple images. Valuable lessons on how to achieve robust classification are also discussed. The technique is validated using a publicly-available, real-world data set collected by the NSF-funded StEER teams during the 2017 and 2018 hurricanes. The technique provides automated capabilities, reducing effort, improving consistency, and accelerating decisions after a major event.

The last part of this research, module 4, focuses on developing the building-specific empirically-calibrated probabilistic vulnerability models automatically. The key contribution of this research is to automate the process of developing empirical probabilistic vulnerability models using post-event images. The vulnerability models are generated and validated using field data, including key structural and nonstructural characteristics of the built environment. These empirically calibrated vulnerability models are used to demonstrate the application of vulnerability models in regional loss prediction for the

similar/same geographical regions. The data collected after hurricane Harvey, in 2017, in Texas, is used as the reference dataset. This dataset is captured during reconnaissance missions led by the NSF-funded Structural Extreme Events Reconnaissance (StEER) Network, with data collection supported by the Fulcrum App [89].

As demonstrated in Fig. 6.1, the associated computational techniques developed along with this approach can be used by policymakers in determining a suitable field inspection budget in advance. Different communities, with different perspectives regarding budget and what level of risk is tolerable, can adopt this approach and use the results to support their decisions in case of a disruptive event. With their relevant input data, communities can use this approach to determine a suitable budget and plan for a range of resilience goals based on risk tolerance. One of the significant advantages of this approach is the applicability of the technique as wide as the street view imagery coverage. The application of the approach is highly affordable because of the availability of the commercial street view services and the historical post-disaster data repositories. When resources are limited in the aftermath of a disruptive event, the results support the community to perform the field inspection by prioritizing certain buildings and pre-classifying less critical ones based on expected performance levels. This capability will reduce overall costs and support a faster start to the rest of the recovery processes. This approach is shown to support informed decision-making at a community-level, but also at a national or individual level, to prepare for and mitigate the impact of disruptive events.



REFERENCES

- [1] R. J. Klein, R. J. Nicholls, and F. Thomalla, "Resilience to natural hazards: How useful is this concept?" *Global Environmental Change Part B: Environmental Hazards*, vol. 5, no. 1, pp. 35–45, 2003.
- [2] A. Rose, "Economic resilience to natural and man-made disasters: Multidisciplinary origins and contextual dimensions," *Environmental Hazards*, vol. 7, no. 4, pp. 383–398, 2007.
- [3] A. Lenjani, I. Bilionis, S. J. Dyke, C. M. Yeum, and R. Monteiro, "A resilience-based method for prioritizing post-event building inspections," *Natural Hazards*, pp. 1–20, 2020.
- [4] A. Rose, "Defining and measuring economic resilience to disasters," *Disaster Prevention and Management: An International Journal*, vol. 13, no. 4, pp. 307–314, 2004.
- [5] S. L. Cutter, L. Barnes, M. Berry, C. Burton, E. Evans, E. Tate, and J. Webb, "A place-based model for understanding community resilience to natural disasters," *Global environmental change*, vol. 18, no. 4, pp. 598–606, 2008.
- [6] L. Gunderson, "Ecological and human community resilience in response to natural disasters," *Ecology and society*, vol. 15, no. 2, 2010.
- [7] G. Hutter, C. Kuhlicke, T. Glade, and C. Felgentreff, "Natural hazards and resilience: exploring institutional and organizational dimensions of social resilience," *Natural hazards*, vol. 67, no. 1, pp. 1–6, 2013.
- [8] M. Mieler, B. Stojadinovic, R. Budnitz, S. Mahin, and M. Comerio, "Towards resilient communities: A performance-based engineering framework for design and evaluation of the built environment," *PEER*, vol. 2013, 2013.
- [9] M. Mieler, B. Stojadinovic, R. Budnitz, M. Comerio, and S. Mahin, "A framework for linking community-resilience goals to specific performance targets for the built environment," *Earthquake Spectra*, vol. 31, no. 3, pp. 1267–1283, 2015.
- [10] R. T. Eguchi, J. D. Goltz, C. E. Taylor, S. E. Chang, P. J. Flores, L. A. Johnson, H. A. Seligson, and N. C. Blais, "Direct economic losses in the northridge earthquake: A three-year post-event perspective," *Earthquake Spectra*, vol. 14, no. 2, pp. 245–264, 1998.
- [11] J. C. Ingram, G. Franco, C. Rumbaitis-del Rio, and B. Khazai, "Post-disaster recovery dilemmas: challenges in balancing short-term and long-term needs for vulnerability reduction," *Environmental Science & Policy*, vol. 9, no. 7-8, pp. 607–613, 2006.
- [12] F. Cavalieri, P. Franchin, P. Gehl, and D. D'Ayala, "Bayesian networks and infrastructure systems: Computational and methodological challenges," in *Risk and Reliability Analysis: Theory and Applications.* Springer, 2017, pp. 385–415.

- [13] M. Bruneau, S. E. Chang, R. T. Eguchi, G. C. Lee, T. D. O'Rourke, A. M. Reinhorn, M. Shinozuka, K. Tierney, W. A. Wallace, and D. Von Winterfeldt, "A framework to quantitatively assess and enhance the seismic resilience of communities," *Earth-quake spectra*, vol. 19, no. 4, pp. 733–752, 2003.
- [14] K. Tierney and M. Bruneau, "Conceptualizing and measuring resilience: A key to disaster loss reduction," *TR news*, no. 250, 2007.
- [15] M. C. Comerio, Disaster hits home: New policy for urban housing recovery. Univ of California Press, 1998.
- [16] L. Pearce, "Disaster management and community planning, and public participation: how to achieve sustainable hazard mitigation," *Natural hazards*, vol. 28, no. 2-3, pp. 211–228, 2003.
- [17] S. B. Miles and S. E. Chang, "Resilus: A community based disaster resilience model," *Cartography and Geographic Information Science*, vol. 38, no. 1, pp. 36–51, 2011.
- [18] Applied Technology Council, "ATC-20," 1982. [Online]. Available: https://www.atcouncil.org/atc-20
- [19] J.-A. Goulet, C. Michel, and A. D. Kiureghian, "Data-driven post-earthquake rapid structural safety assessment," *Earthquake Engineering & Structural Dynamics*, vol. 44, no. 4, pp. 549–562, 2015.
- [20] M. Fernandez, L. Alvarez, and R. Nixon, "Still waiting for FEMA in texas and florida after hurricanes," 2017. [Online]. Available: https://www.nytimes.com/2017/10/22/us/fema-texas-florida-delays-.html
- [21] D. M. Frangopol and A. C. Estes, "Optimum lifetime planning of bridge inspection and repair programs," *Structural Engineering International*, vol. 9, no. 3, pp. 219–223, 1999.
- [22] D. Straub and M. H. Faber, "Risk based inspection planning for structural systems," *Structural safety*, vol. 27, no. 4, pp. 335–355, 2005.
- [23] D. T. Phan and Y. Zhu, "Multi-stage optimization for periodic inspection planning of geo-distributed infrastructure systems," *European Journal of Operational Research*, vol. 245, no. 3, pp. 797–804, 2015.
- [24] D. M. Frangopol and M. Soliman, "Life-cycle of structural systems: recent achievements and future directions," *Structure and infrastructure engineering*, vol. 12, no. 1, pp. 1–20, 2016.
- [25] G. Hearn, *Bridge inspection practices*. Transportation Research Board, 2007, vol. 375.
- [26] D. Alexander, "Planning for post-disaster reconstruction," in *I-Rec 2004 International Conference Improving Post-Disaster Reconstruction in Developing Countries*, 2004.
- [27] J. A. Ramirez, R. J. Frosch, M. A. Sozen, and A. M. Turk, "Handbook for the post-earthquake safety evaluation of bridges and roads," 2000.

- [28] M. Bensi, A. D. Kiureghian, and D. Straub, "Framework for post-earthquake risk assessment and decision making for infrastructure systems," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, vol. 1, no. 1, p. 04014003, 2014.
- [29] U. Rosenthal and A. Kouzmin, "Crises and crisis management: Toward comprehensive government decision making," *Journal of Public Administration Research and Theory*, vol. 7, no. 2, pp. 277–304, 1997.
- [30] P. Alipour, S. Mukherjee, and R. Nateghi, "Assessing climate sensitivity of peak electricity load for resilient power systems planning and operation: A study applied to the texas region," *Energy*, vol. 185, pp. 1143–1153, 2019.
- [31] S. Safarkhani, I. Bilionis, and J. Panchal, "Understanding the effect of task complexity and problem-solving skills on the design performance of agents in systems engineering," in *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.* American Society of Mechanical Engineers Digital Collection, 2018.
- [32] S. Safarkhani, I. Bilionis, and J. H. Panchal, "Toward a theory of systems engineering processes: A principal–agent model of a one-shot, shallow process," *IEEE Systems Journal*, 2020.
- [33] I. Grigoratos, R. Monteiro, P. Ceresa, A. Di Meo, M. Faravelli, and B. Borzi, "Crowd-sourcing exposure data for seismic vulnerability assessment in developing countries," *Journal of Earthquake Engineering*, pp. 1–18, 2018.
- [34] A. Lenjani, I. Bilionis, S. Dyke, C. M. Yeum, and R. Monteiro, "A resilience-based method for prioritizing post-event building inspections," *arXiv* preprint *arXiv*:1906.12319, 2019.
- [35] C. Baggio, A. Bernardini, R. Colozza, L. Corazza, M. Della Bella, G. Di Pasquale, M. Dolce, A. Goretti, A. Martinelli, G. Orsini et al., "Field manual for post-earthquake damage and safety assessment and short term countermeasures (aedes)," European Commission—Joint Research Centre—Institute for the Protection and Security of the Citizen, EUR, vol. 22868, 2007.
- [36] J. D. Marshall, K. Jaiswal, N. Gould, F. Turner, B. Lizundia, and J. C. Barnes, "Postearthquake building safety inspection: lessons from the canterbury, new zealand, earthquakes," *Earthquake Spectra*, vol. 29, no. 3, pp. 1091–1107, 2013.
- [37] H. Kellerer, U. Pferschy, and D. Pisinger, "Introduction to np-completeness of knapsack problems," in *Knapsack problems*. Springer, 2004, pp. 483–493.
- [38] Google optimization tools, "Google OR-Tools," 2019. [Online]. Available: https://developers.google.com/optimization/
- [39] K. A. Porter and A. S. Kiremidjian, Assembly-based vulnerability of buildings and its uses in seismic performance evaluation and risk-management decision-making. SPA Risk LLC, 2000.
- [40] J. W. Baker, "Efficient analytical fragility function fitting using dynamic structural analysis," *Earthquake Spectra*, vol. 31, no. 1, pp. 579–599, 2015.

- [41] V. Silva, S. Akkar, J. Baker, P. Bazzurro, J. M. Castro, H. Crowley, M. Dolsek, C. Galasso, S. Lagomarsino, R. Monteiro *et al.*, "Current challenges and future trends in analytical fragility and vulnerability modelling," *Earthquake Spectra*, 2019.
- [42] C. M. Yeum, A. Lenjani, S. J. Dyke, and I. Bilionis, "Automated detection of predisaster building images from google street view," *arXiv preprint arXiv:1902.10816*, 2019.
- [43] A. Lenjani, S. J. Dyke, I. Bilionis, C. M. Yeum, K. Kamiya, J. Choi, X. Liu, and A. G. Chowdhury, "Towards fully automated post-event data collection and analysis: preevent and post-event information fusion," *arXiv preprint.*, 2019.
- [44] A. Lenjani, C. M. Yeum, S. Dyke, and I. Bilionis, "Automated building image extraction from 360-degree panoramas for post-disaster evaluation," *arXiv preprint arXiv:1905.01524*, 2019.
- [45] A. Di Meo, B. Borzi, M. Faravelli, M. Pagano, P. Ceresa, R. Monteiro, and J. Al-Dabbeek, "Seismic vulnerability assessment of the urban building environment in nablus, palestine," *International Journal of Architectural Heritage*, vol. 12, no. 7-8, pp. 1196–1215, 2018.
- [46] B. Borzi, H. Crowley, and R. Pinho, "Simplified pushover-based earthquake loss assessment (sp-bela) method for masonry buildings," *International Journal of Architectural Heritage*, vol. 2, no. 4, pp. 353–376, 2008.
- [47] B. Borzi, R. Pinho, and H. Crowley, "Simplified pushover-based vulnerability analysis for large-scale assessment of rc buildings," *Engineering Structures*, vol. 30, no. 3, pp. 804–820, 2008.
- [48] D. Bolognini, B. Borzi, and R. Pinho, "Simplified pushover-based vulnerability analysis of traditional italian rc precast structures," in *Proceedings of the 14th world conference on earthquake engineering, Beijing, China*, 2008.
- [49] G. Grünthal, "European macroseismic scale 1998," European Seismological Commission (ESC), Tech. Rep., 1998.
- [50] M. Faravelli, B. Borzi, A. Di Meo, and D. Polli, "A mechanic-based model for definition of seismic risk and real time damage scenario of buildings," in *Proceedings of the 6th International Conference on Computational Methods in Structural Dynamics and Earthquake Engineering, Rhodes Island, Greece*, 2017.
- [51] Home Advisor, "Learn how much it costs to hire a structural engineer." 2018. [Online]. Available: https://www.homeadvisor.com/cost/architects-and-engineers/hire-an-engineer/
- [52] Angie's List, "How much does a structural engineer cost?" 2018. [Online]. Available: https://www.angieslist.com/articles/how-much-does-structural-engineer-cost.htm
- [53] W. K. Viscusi and J. E. Aldy, "The value of a statistical life: a critical review of market estimates throughout the world," *Journal of risk and uncertainty*, vol. 27, no. 1, pp. 5–76, 2003.
- [54] H. Adeyemo and M. Ahmed, "Solving 0/1 knapsack problem using metaheuristic techniques," in 2017 9th IEEE-GCC Conference and Exhibition (GCCCE). IEEE, 2017, pp. 1–6.

- [55] T. Hajilounezhad, S. Safari, and M. Aliehyaei, "Multi-objective optimization of solid oxide fuel cell/gt combined heat and power system: A comparison between particle swarm and genetic algorithms," 2020.
- [56] P. Alipour, P. Damodaran, and C. Nguyen, "Using grasp approach and path relinking to minimise total number of tardy jobs on a single batch processing machine," *International Journal of Industrial and Systems Engineering*, vol. 35, no. 1, pp. 80–99, 2020.
- [57] H. M. Markowitz, "Foundations of portfolio theory," *The journal of finance*, vol. 46, no. 2, pp. 469–477, 1991.
- [58] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.
- [59] V. Shet, "Go back in time with street view," 2014.
- [60] A. Lenjani, C. M. Yeum, S. Dyke, and I. Bilionis, "Automated building image extraction from 360 panoramas for postdisaster evaluation," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 3, pp. 241–257, 2020.
- [61] G.-I. Kweon, "Panoramic image composed of multiple rectilinear images generated from a single fisheye image," *Journal of the Optical Society of Korea*, vol. 14, no. 2, pp. 109–120, 2010.
- [62] M. Ruder, A. Dosovitskiy, and T. Brox, "Artistic style transfer for videos and spherical images," *International Journal of Computer Vision*, vol. 126, no. 11, pp. 1199–1219, 2018.
- [63] C. M. Yeum, S. J. Dyke, B. Benes, T. Hacker, J. Ramirez, A. Lund, and S. Pujol, "Postevent reconnaissance image documentation using automated classification," *Journal of Performance of Constructed Facilities*, vol. 33, no. 1, p. 04018103, 2018.
- [64] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [65] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [66] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [67] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 6, pp. 0756–777, 2004.
- [68] H. Li and R. Hartley, "Five-point motion estimation made easy," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 1. IEEE, 2006, pp. 630–633.
- [69] S. P. Drake, "Converting gps coordinates [phi, lambda, h] to navigation coordinates (enu)," 2002.
- [70] G. Pandey, J. R. McBride, and R. M. Eustice, "Ford campus vision and lidar data set," *The International Journal of Robotics Research*, vol. 30, no. 13, pp. 1543–1552, 2011.

- [71] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.
- [72] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.
- [73] H. Harzallah, F. Jurie, and C. Schmid, "Combining efficient object localization and image classification," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 237–244.
- [74] G. E. Hinton and R. R. Salakhutdinov, "Replicated softmax: an undirected topic model," in *Advances in neural information processing systems*, 2009, pp. 1607–1614.
- [75] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.
- [76] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
- [77] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.
- [78] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [79] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.
- [80] L. Tzutalin, "Git code (2015)."
- [81] X. Chen and A. Gupta, "An implementation of faster rcnn with study for region sampling," *arXiv preprint arXiv:1702.02138*, 2017.
- [82] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [83] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [84] G. Bradski, "The opency library," *Dr Dobb's J. Software Tools*, vol. 25, pp. 120–125, 2000.
- [85] K. Porter, "A beginner's guide to fragility, vulnerability, and risk," *Encyclopedia of earthquake engineering*, vol. 2015, pp. 235–260, 2015.
- [86] M. Roohi, "Performance-based seismic monitoring of instrumented buildings," 2019.
- [87] M. Roohi and E. M. Hernandez, "Performance-based post-earthquake decision-making for instrumented buildings," *arXiv preprint arXiv:2002.11702*, 2020.

- [88] A. Lenjani, S. J. Dyke, I. Bilionis, C. M. Yeum, K. Kamiya, J. Choi, X. Liu, and A. G. Chowdhury, "Towards fully automated post-event data collection and analysis: preevent and post-event information fusion," *Engineering Structures*, p. 109884, 2020.
- [89] StEER Network. (2018 (accessed: 22.09.2018)) NSF Structural Extreme Events Reconnaissance (StEER) Network. [Online]. Available: https://web.fulcrumapp.com/communities/nsf-rapid
- [90] C. M. Yeum, A. Lenjani, S. J. Dyke, and I. Bilionis, "Automated detection of predisaster building images from google street view," in *Proceedings of The 7th World Conference on Structural Control and Monitoring, (7WCSCM 2018),* 2018.
- [91] M. Roohi, E. M. Hernandez, and D. Rosowsky, "Nonlinear seismic response reconstruction and performance assessment of instrumented wood-frame buildings—validation using neeswood capstone full-scale tests," *Structural Control and Health Monitoring*, vol. 26, no. 9, p. e2373, 2019.
- [92] M. Roohi, E. Hernandez, and D. Rosowsky, "Nonlinear seismic response reconstruction in minimally instrumented buildings—validation using neeswood capstone full-scale tests," *Structural Health Monitoring 2019*, 2019.
- [93] A. Lenjani, S. Dyke, I. Bilionis, C. M. Yeum, J. Choi, A. Lund, and A. Maghareh, "Hierarchical convolutional neural networks information fusion for activity source detection in smart buildings," *Structural Health Monitoring 2019*, 2019.
- [94] T. Hajilounezhad, Z. A. Oraibi, R. Surya, F. Bunyak, M. R. Maschmann, P. Calyam, and K. Palaniappan, "Exploration of carbon nanotube forest synthesis-structure relationships using physics-based simulation and machine learning," 2020.
- [95] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.
- [96] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [97] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.
- [98] F. Chollet et al., "Keras," https://keras.io, 2015.
- [99] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT press, 2016.
- [100] NSF Award 1841667. (2016 (accessed: 22.09.2018)) EAGER: Operationalization of the Structural Extreme Events Reconnaissance (StEER) Network. [Online]. Available: {https://www.nsf.gov/awardsearch/showAward?AWD_ID=1841667&HistoricalAwards=false}
- [101] StEER Network. (2018 (accessed: 25.04.2019)) Structural Extreme Events Reconnaissance Network. [Online]. Available: https://www.steer.network

- [102] T. Kijewski-Correa, A. Taflanidis, A. Kennedy, A. Womble, D. Liang, M. Starek, K. Peterman, C. Sun, F. Leite, R. Wood *et al.*, "Hurricane harvey (texas) supplement–collaborative research: Geotechnical extreme events reconnaissance (geer) association: Turning disaster into knowledge," 2018.
- [103] J.-P. Pinelli, D. Roueche, T. Kijewski-Correa, F. Plaz, D. Prevatt, I. Zisis, A. Elawady, F. Haan, S. Pei, K. Gurley *et al.*, "Overview of damage observed in regional construction during the passage of hurricane irma over the state of florida," *Proc., ASCE Forensic*, vol. 18, 2018.
- [104] T. Kijewski-Correa, D. Prevatt, M. Musetich, D. Roueche, K. Mosalam, F. Hu, A. Salman, H. Peng, C. Gonzalez, and I. Robertson, "Hurricane florence: Field assessment team 1 (fat-1) early access reconnaissance report (earr)," 2018.
- [105] D. Roueche, J. Cleary, K. Gurley, J. Marshall, J.-P. Pinelli, D. Prevatt, D. Smith, A. Alipour, K. Angeles, B. Davis, C. Gonzalez, A. Lenjani, H. Mulchandani, M. Musetich, A. Salman, T. Kijewski-Correa, I. Robertson, and K. Mosalam, "Steerhurricane michael: Field assessment team 1 (fat-1) early access reconnaissance report (earr)," 2018.
- [106] T. Kijewski-Correa, D. Prevatt, I. Robertson, D. Smith, K. Mosalam, D. Roueche, B. Lichty, M. Gutierrez Soto, A. M. Aly, A. Lenjani, A. Salman, A. Alipour, B. Davis, E. Sutley, and N. Miner, "Steer-hurricane michael: Preliminary virtual assessment team (p-vat) report," 2018.
- [107] Spatial Networks, Inc. (2018 (accessed: 22.09.2018)) Fulcrum|Mobile Form Builder & Data Collection App. [Online]. Available: https://www.fulcrumapp.com
- [108] DesignSafe-CI. (2018 (accessed: 22.02.2019)) DesignSafe-CI: A Comprehensive Cyberinfrastructure Environment for Research in Natural Hazards Engineering. [Online]. Available: https://www.designsafe-ci.org/
- [109] D. B. Roueche, D. O. Prevatt, and F. T. Lombardo, "Epistemic uncertainties in fragility functions derived from post-disaster damage assessments," *ASCE-ASME J. Risk Uncertainty Eng. Syst. A Civ. Eng.*, vol. 4, no. 2, p. 04018015, 2018.
- [110] K. Pitilakis, H. Crowley, and A. M. Kaynia, "Syner-g: typology definition and fragility functions for physical elements at seismic risk," *Geotechnical, Geological and Earthquake Engineering*, vol. 27, 2014.
- [111] Applied Technology Council, *Earthquake damage evaluation data for California*. Applied technology council, 1985.
- [112] D. D'ayala, A. Meslem, D. Vamvatsikos, K. Porter, T. Rossetto, H. Crowley, and V. Silva, "Guidelines for analytical vulnerability assessment of low/mid-rise buildings—methodology. vulnerability global component project," 2014.
- [113] K. Pitilakis, S. Karapetrou, and S. Fotopoulou, "Consideration of aging and ssi effects on seismic vulnerability assessment of rc buildings," *Bulletin of Earthquake Engineering*, vol. 12, no. 4, pp. 1755–1776, 2014.
- [114] F. Peña, I. Bilionis, and S. Dyke, "Model selection and uncertainty quantification of seismic fragility functions," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, vol. 5, no. 3, p. 04019009, 2019.

- [115] Zillow. (2018 (accessed: 23.05.2020)) How much is my home worth? [Online]. Available: https://www.zillow.com/how-much-is-my-home-worth/
- [116] National Weather Service. (2020) Major Hurricane Harvey August 25-29, 2017. [Online]. Available: https://www.weather.gov/crp/hurricaneharvey
- [117] P. J. Vickery, P. F. Skerlj, J. Lin, L. A. Twisdale Jr, M. A. Young, and F. M. Lavelle, "Hazus-mh hurricane model methodology. ii: Damage and loss estimation," *Natural Hazards Review*, vol. 7, no. 2, pp. 94–103, 2006.
- [118] Federal Geographic Data Committee (FGDC). (2020) Whole community geospatial data, tools, and resources for disaster response and resiliency. [Online]. Available: http://disasters.geoplatform.gov
- [119] P. J. Vickery, D. Wadhera, L. A. Twisdale Jr, and F. M. Lavelle, "Us hurricane wind speed risk and uncertainty," *Journal of structural engineering*, vol. 135, no. 3, pp. 301–320, 2009.
- [120] P. J. Vickery, F. J. Masters, M. D. Powell, and D. Wadhera, "Hurricane hazard modeling: The past, present, and future," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 97, no. 7-8, pp. 392–405, 2009.
- [121] T. Chen, T. He, M. Benesty, V. Khotilovich, and Y. Tang, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, pp. 1–4, 2015.
- [122] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings* of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.