UNIVERSITY OF CALIFORNIA,
IRVINE

Post-Disaster Structural Health Assessment System Using Personal Mobile-Phones

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Electrical Engineering

by

Ahmed A. Alzughaibi

Thesis Committee:
Professor Ahmed Eltawil, Chair
Professor Fadi Kurdahi
Associate Professor Ozdal Boyraz

2018

DEDICATION

This thesis is wholeheartedly dedicated to my beloved wife and daughter, who have been my source of inspiration and gave me strength when I thought of giving up, who continually provide their moral, spiritual, and emotional support.

To my father who have been supportive in every stage in my live.

To my wonderful deeply missed Mather. Forever you remain in my soul.

To my beloved Grandmother. I will always remember you.

And lastly, to my brothers, sisters, relatives, mentor, friends who shared their words of advice and encouragement to finish this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACKNOWLEDGMENTS

I would like to express the deepest appreciation and gratitude to my advisor, Professor Ahmed Eltawil, for his exceptional guidance, motivation and support throughout this academic journey. He raised my ambitions to be the best version of myself which concluded by completing this work. Without his persistent help and thoughtful insight, this thesis would not have been possible.

I would like to thank Professor Fadi Kurdahi and Professor Ozdal Boyraz for being part of my thesis committee.

Very special thanks to Ahmed Mokhtar Ibrahim, who is a good friend of mine and the lead researcher of the research group. His extensive understanding, experience and continuous help have paved the way to start this research with the best background, knowledge and experiments setup. It would be impossible to reach this point of research without his guidance.

Also special thanks to my friends Ibrahim Alquaydheb, Mohammad Almajhadi, Umar Kazmi, Ahmed Eisa, Mohamed Fouda, Michael Ayoub and Mohammed Alnemari for all of the support and encouragement that helped immensely to complete this degree.

Moreover, a message of recognition to Qassim University and to the Saudi Arabian Cultural Mission for all outstanding support. I would not be able to complete all of this achievement without their constant aid.

In addition, I consider myself very lucky and blessed to come from such an amazing and supportive family. My most sincere gratitude and thanks to my wife, daughter, parents and grandparents. Also, to my brother Khalid for his support in several stages in the project. To all of my brothers and sisters for the unconditional love and encouragement.

Finally, I want to thank all my friends Fahad Albassam, Fahad Albogami, Mohammed Alsiaifi, Waleed Almohead, Hamad Alharkan, and Ibrahim Alsadrani for their outstanding support. I am proud of being a friend for life.

ABSTRACT OF THE THESIS

Post-Disaster Structural Health Assessment System Using Personal Mobile-Phones

By

Ahmed A. Alzughaibi

Master of science in Electrical Engineering

University of California, Irvine, 2018

Professor Ahmed Eltawil, Chair

In the aftermath of a natural disaster reconnaissance is typically conducted by teams of engineers tasked with tagging buildings according to their damage state. Tagging (red, yellow, or green) conveys information about the condition of the building (unsafe, needs further evaluation, or safe, respectively). While thorough, the process can take several days to weeks to be completed. Automated assessment is an attractive alternative to manual inspection but requires deploying a dense network of sensors at the granularity of each structure. Such a network was deemed to be impractical with respect to cost or deployment time. However, with the advent of the Internet of things (IoT) era, a massive network of citizen-owned smart devices such as tablets and smart-phones that contain vibration sensors (e.g. accelerometers) is already deployed. The objective of this work is to develop a framework that can crowd-source relatively low-quality readings from distributed smart citizen owned devices and distill that information into actionable information. This information can be provided to public safety personnel within minutes of an event, in the form of a disaster map, with buildings tagged by their most likely damage state. This paper reports on the development of an application running on a mobile phone to collect readings and coupled to a cloud based server used to generate the necessary tags.

Chapter 1

Introduction

Every year, earthquakes hit largely-populated cities destroying infrastructure and killing hundreds to thousands of people depending on the severity of the event [14]. Recently, significant research attention has been devoted to studying the feasibility of applying automated systems in detecting and reporting seismic events. The two areas that are most prominently studied are Earthquake Early Warning (EEW) systems and post-disaster structural health assessment [15, 16]. Several recent efforts targeted automatic structural health monitoring by using satellite imaging [4], or by installing high accurate vibration sensors [17]. The main disadvantage in adopting satellite monitoring is that it is capable of only capturing partial to complete building collapse. On the other hand, high accuracy accelerometers are capable of detecting the building response to the event, from which its state can be inferred. However, to achieve large-scale deployment, installing a dense network of high accuracy accelerometers is needed which is not practical with respect to cost or time of deployment.

With the advent of the Internet of things (IoT) era, a dense network of citizen owned devices is already deployed, such as smart-phones, tablets, ...etc. These devices are already equipped with vibration sensors, specifically accelerometers. The objective of this work is to

develop a framework that can crowd-source relatively low-quality readings from distributed smart citizen owned devices and distill that information into actionable information. In prior work, an EEW system (called MyShake) was developed using only mobile sensors data [10]. MyShake successfully provided a 20 seconds warning for a magnitude 5 (M5) earthquake. Furthermore, CSN, another earthquake detection system used phone accelerometers to capture the s-wave in addition to the much harder to detect p-wave [18]. While prior work focused on providing early warning, in this work we focus on utilizing mobile accelerometer reading to directly assess the structural health of buildings post event.

According to the Federal Emergency Management Agency (FEMA), the relative displacement between adjacent floors is related to building health [19] via the Inter-story Drift Ratios (IDRs). IDRs can be used to classify structures as immediate occupancy (IO), life safety (LS), or collapse prevention (CP) depending on the maximum instantaneous value of the IDR.

1.1 What is the Inter-story Drift Ratio (IDR)?

Buildings react to earthquake shaking in different ways depending on the type of the building and how strong the earthquake is. Buildings shake as 1st, 2nd, 3rd, etc harmonic waves. IDR is a quantitative measure of how the structure bends in response to earthquake shaking. When a building bends over a certain threshold, it is classified as inhabitable since it's columns are most likely damaged. Figures 1.1 and 1.2 depict a building with and without earthquake shaking.

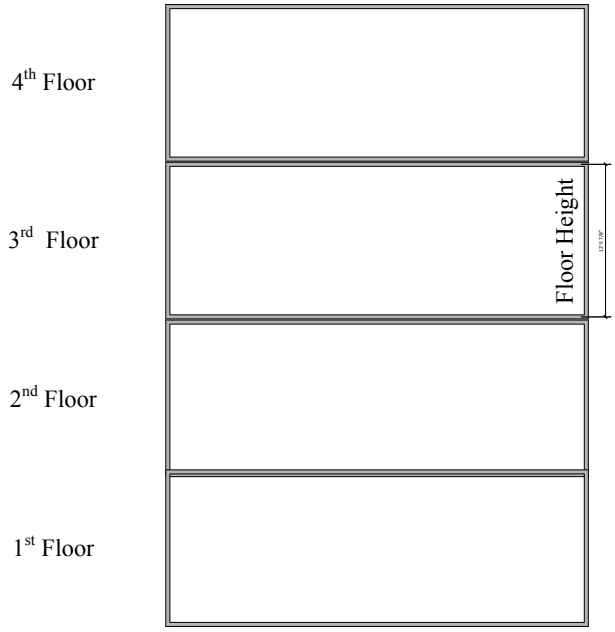


Figure 1.1: 4-story Building

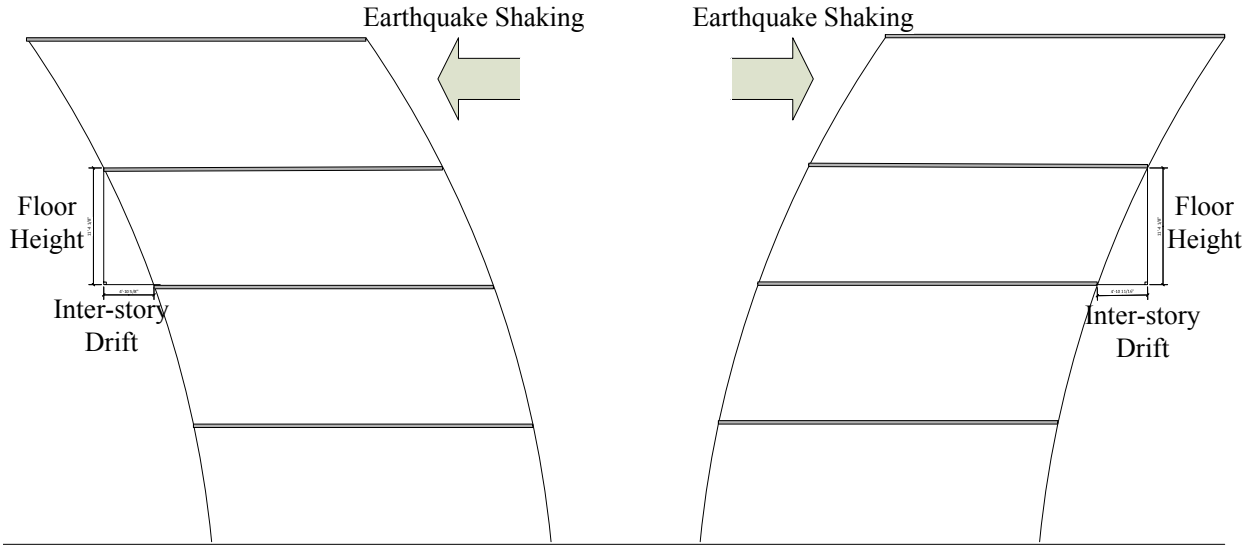


Figure 1.2: Building under earthquake shaking

In fact, IDR is defined as the drift between two adjacent floors at a moment of time as a factor of the floor height. Equation 1.1 is used to calculate the IDR in a moment of time.

$$IDR = \frac{Inter\text{-}story\ Drift}{Floor\ Height} \tag{1.1}$$

Equation 1.1 can be rewritten as fallows

$$IDR_i = \frac{Disp._{Floor_i} - Disp._{Floor_{i+1}}}{FloorHeight} \tag{1.2}$$

where $Disp._{Floor_i}$ and $Disp._{Floor_{i+1}}$ are the floor displacement of floors$i$ and $i+1$, respectively. The equivalent IDR of the building (IDR_b) is the maximum of all IDR_i in the building, the mathematical equation is given in equation 1.3

$$IDR_b = max\left\{IDR_i : i = 1, 2, ..., M-1\right\} \tag{1.3}$$

where M is number of floors in building b. After that, the structure is classified using FEMA standards. For example, Table 1.1 represents the threshold IDR of steel moment-frame buildings. Refer to section 3.3 for detailed explanations of the classification process.

Table 1.1: Classification thresholds of steel moment-frame buildings, according to FEMA [12]

IDR %	Building State
IDR $< 2.5\%$	Immediate occupancy (IO)
$2.5\% <$ IDR $< 5\%$	Life safety (LS)
IDR $> 5\%$	Collapse prevention (CP)

1.2 Earthquake Magnitude Scale

Before designing the system, we must study the earthquake that usually affects the structural health of buildings. The United States Geological Survey (USGS) [13] categorized

earthquakes based on their magnitudes and effectiveness to structural health of buildings. Table 1.2 presents earthquake magnitudes along with their effects on buildings and frequency per year reported in [13].

Table 1.2: List of earthquake magnitudes along with their effects to buildings and frequency per year, reported in [13]

Magnitude	Earthquake Effects	Estimated Number Each Year
2.5 or less	Usually not felt, but can be recorded by seismograph.	900,000
2.5 to 5.4	Often felt, but only causes minor damage.	30,000
5.5 to 6.0	Slight damage to buildings and other structures.	500
6.1 to 6.9	May cause a lot of damage in very populated areas.	100
7.0 to 7.9	Major earthquake. Serious damage.	20
8.0 or greater	Great earthquake. Can totally destroy communities near the epicenter.	One every 5 to 10 years

1.3 An Examples of a Damaged Structure

A hotel in Van Nuys, California has been damaged during the Northridge earthquake in 1994. Although there is no noticeable damage when seen from a distance (Figure 1.3), the main columns of the building have been severely damaged (Figure 1.4).

Figure 1.3: Holday Inn Van Nuys, California seen from distance after 1994 Northridge earthquake.

(a) Exterior up-close view of the column damage cased by the earthquake

(b) Interior up-close view of the column damage cased by the earthquake

Figure 1.4: Holday Inn Van Nuys, California after 1994 Northridge earthquake.

1.4 Current methods for post-disaster building assessment

There are several methods for structural health assessment in the aftermath of an earthquake. The most popular method is conducting an on-site inspection by teams of engineers. However, having engineers visit every building is impractical. This process takes weeks to months which could result in additional loss of lives. Another method used for structural health assessment is using the satellite maps to tag the collapsed building. The main issue with using satellites is that they can only detect partially or completely collapsed buildings.

The most accurate automated method to classify the structural health of buildings is using seismic-grade accelerometers to calculate the IDR. However, it is expensive to implement seismic-grade accelerometers on every floor. Pros and cons of each inspection method are summarized in Table 1.3.

1.5 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 contains the literature review. Chapter 3 provides an overview of the system architecture. Chapter 4 reviews the experimental validation of the system. Finally, the conclusion is drawn in Chapter 4.4.

Table 1.3: Current methods used for post-disaster building assessment

Inspection Method	**Definition**	**Advantages**	**Drawbacks**
Physical Inspection	on-site door-by-door inspection by teams of engineers	accurate decisions made by professional	long process with possibility of lost lives due to unchecked collapsed buildings; very expensive
Satellite Monitoring	comparing before- and after-disaster satellite maps to tag partially or completely collapsed buildings	relatively fast detection	detects only partially or completely collapsed building
Seismic-grade Accelerometers	installing seismic-grade accelerometers in every floor in every building	accurate accelerometer reading means accurate relative displacement calculation	very expensive to set up in every building

Chapter 2

Literature Review

The literature review is divided into two thematic sections in addition to a concluding section. Section 2.1 reviews the current community-based earthquake monitoring systems. Section 2.2 discusses the structural health assessment systems in the aftermath of an earthquake.

Each section contains chronologically ordered publications in the same theme. The title of the publications subsection contains the publication titles, authors and years of publishing.

2.1 Community-based Automated Earthquake Monitoring Systems

In This section, a list of systems to utilize community-based sensors in seismology is presented. There are two major systems that are currently working. First, Community Sensing Network (CSN) which was developed in 2011[1]. Second, MyShake, originally iShake, introduced in 2013[1].

[1]This is the year of the first known publication

2.1.1 Community Sensing Network (CSN), 2011 -

Community Seismic Network, R Clayton et al, 2011 [8]

This article reports the design of CSN. CSN is a dense open seismic network based on cheap MEMS sensors owned by volunteers from the community which are connected to their personal computers or sensors located in mobile-phone devices. The readings are then sent to a cloud server to be processed.

The main output of CSN is a map of peak acceleration, generated seconds after the event. It was suggested that sensors in buildings can be useful for monitoring the state-of-health of the structure after major earthquakes. The vision of this work is replacing the USGS system *Did You Feel It (DYFI)?* which is a subjective measure of the earthquake intensity i.e. asking people if they have noticed the shaking and how strong the shaking was in a scale from 1 to 10. CSN was intended as a supplement to increase the resolution of ground shaking measurements of traditional networks, not as a replacement.

The paper lists some of the advantages of dense seismic networks over traditional seismic networks. First, the real-time processing is generally simpler by easily generating the hypocenter using the first few readings. Second, a map of shaking is produced directly from the readings rather than from a model-based approach that depends on knowing the epicenter of the event as in the traditional networks. Figure 2.1 shows a comparison between maps generated using traditional seismic network (top) vs. using the proposed dense network (bottom).

Third is the relatively low cost of dense networks. In fact, CSN uses Phidget sensors, that cost about $100, rather than the extraordinarily expensive seismic-grade accelerometers, which are in the order of thousands of dollars. As a trade-off, Phidgets have a noise level that is much higher than seismic-grade accelerometers. However, the noise level of Phidgets is 1/4 of the noise level of sensors that were equipped in smartphones in 2011.

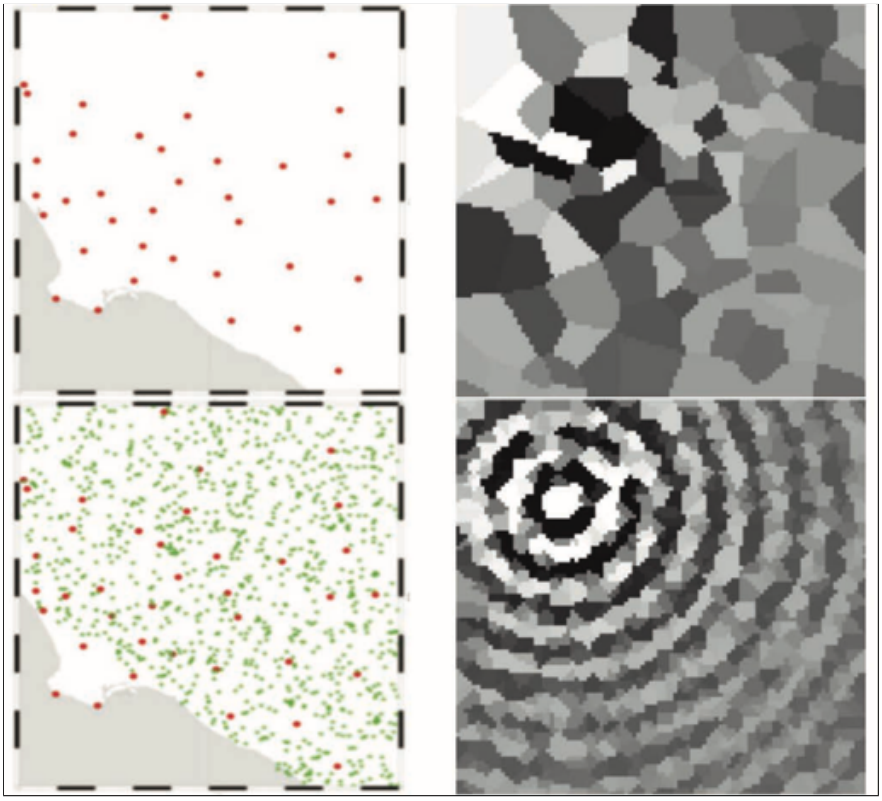


Figure 2.1: A comparison between maps generated using traditional seismic network (top) vs. using the proposed dense network (bottom), proposed in [8].

A client software is used to retrieve the sampled data from the sensor and apply minor data processing to the signal to be sent through the connected computer to the server. The detection algorithm uses a standard ratio of short-term-average over long-term average. The information that is sent when an earthquake is detected are the peak amplitude, time and location of the event. The sampling rate that was used in the system was 50 samples per second.

The installation of the proposed system was planned in Pasadena, CA in 2011. The major challenge of the suggested system is convincing the public to buy and install the sensors.

The Next Big One: Detecting Earthquakes and other Rare Events from Community-based Sensors, M Faulkner et al., 2011 [20]

In this publication, M Faulkner et al. (2011), proposed a statistical approach to initiate an early warning system using CSN. In addition, they introduced the use of mobile-phones int the CSN network.

M Faulkner et al. conducted an experiment to assess the sensitivity of the sensors, estimating and evaluating sensors' ROC curves using experiments involving historical earthquakes data. For instance, it was claimed that approximately 100 Android clients or 20 Phidgets per 20 km by 20 km area may be sufficient to achieve roughly 99% correct detection for events magnitude 5 and above.

Community Sense and Response Systems: Your Phone as Quake Detector, M Faulkner et al., 2014 [18]

This work proposed using decentralized event detection in the CSN system. It was claimed that introducing the decentralized event detection in addition to the server level detection would minimize the error alerts.

In addition, an ML detection algorithm was introduced to the system. The researchers suggest that using a more sophisticated ML algorithm will reduce the probability error.

The team conducted an experiment on the suggested system and found that approximately 50 phones or 10 Phidgets are enough to detect a nearby magnitude 5 or larger event with about 99% success.

Dense building instrumentation application for city-wide structural health monitoring, 2017 [21]

In this work, a practical real experiment was conducted. CSN initiated a structural monitoring program for NASA Jet Propulsion Laboratory (JPL) campus buildings. Researchers claimed that evaluation of data during and after events using the CSN system shows the usefulness of higher density networks more than methodw that use traditional sparser deployment.

The system still requires the installation of MEMS sensors, connected to active computers during the event, in all buildings and areas being monitored which is impractical.

2.1.2 iShake and MyShake, 2013 -

Mobile phones as seismologic sensors: automating data extraction for the iShake system, J Reilly, 2013 [22]

iShake is a seismic sensing system designed by a team at UC Berkeley. iShake uses sensors in mobile-phone devices to measure earthquake ground shaking. The aim of the iShake project is to provide the public with a way to contribute more quantitative data, than USGS's *Did You Feel It (DYFI) ?* to earthquake research by automating the sensor reading collection via the iShake mobile application.

Shake table testing was conducted on iPhones to determine if they are sensitive enough to sense earthquake shaking. iShake focuses on recovering earthquake-related parameters, such as Arias intensity and spectral ordinates rather than an early-warning system.

To solve the high power hungriness of the app, the device has to be charging at the time of the earthquake. In addition, the device has to enter a steady mode, by not exceeding a certain

threshold for a predefined period of time, before it could be triggered. The mechanism used to detect earthquake is crossing a predefined threshold i.e. when the acceleration value is larger than a predefined value, the system can be triggered. After that, the application sends the detected earthquake readings to a server along with the timing and GPS location.

On the server end, signal processing is done to the received data to ensure accurate results. First, detected earthquakes have to be verified with USGS earthquake detection database to be considered real earthquakes. Second, fall detection is applied by detecting the large change in Arias Intensity. In addition, specific phone resonance needs to be filtered out of the reading for unbiased results. In the end, a map using the strengths of the readings along with their locations is presented.

A virtual earthquake field test is conducted to test the feasibility of the proposed system visualization map. This work does not focus on detecting the structural damage caused by earthquakes.

MyShake: A smartphone seismic network for earthquake early warning and beyond, Q Kong, 2016 [10]

The goal of the proposed system in the paper (MyShake, initially iShake) has been changed to an earthquake early warning (EEW) system. At the beginning of this work, noise floors of several lately released phones, at the time of publication, were calculated and compared to currently used sensors in traditional networks and lately developed MEMS sensor, that are too expensive to be integrated into mobile phones. The study found that mobile phone sensors can detect earthquakes that are magnitude 5 or stronger. In addition, the study found an improvement of the noise floor over the years.

MyShake introduced intelligent detection algorithms instead of using the threshold mechanism. Researchers have used three classifiers to classify earthquake shaking from movements

applied to the device. The three classifiers used are Zero-Crossing (ZC), which is a frequency measure, IQR, which is an amplitude measure, and cumulative absolute velocity (CAV).

MyShake successfully provided a 20 seconds warning for a magnitude 5 (M5) earthquakes. The system still does not focusing on detecting the structural damage caused by earthquakes.

2.2 Post-Earthquake Structural Health Assessment

Instead of conducting reconnaissance by teams of engineers tasked with tagging buildings according to their damage state, several automated structural health assessment methodologies were proposed over the past few years. In this section, some of the purposed methodologies are listed.

2.2.1 Continuous GPS monitoring of structural deformation at Pacoima dam, California, K Hudnut et al., 1998 [1]

The first discussed methodology is using GPS to monitor the structural health of civil structures. Pacoima Dam, built in 1928, suffered structural deformation after 1971 San Fernando and 1994 Northridge earthquakes. In 1995, a system of three GPS receivers was deployed in the dam. The proposed system used the GPS receivers to monitor the momentary displacements of the 70-year-old dam relative to a stable station in a nearby station.

After two years of continually monitoring the dam movement, time series to the GPS receiver started to indicate downstream motion of the dam during the fall and winter, followed by upstream deflection during the spring and summer. The deflection was nearly $2cm$ peak-to-peak amplitude with an approximately 1-year period.

The present system at Pacoima Dam was operating in near real-time manner. There was a time lag of several days. The suggested system required installing GPS sensors in the monitored structure.

2.2.2 Determination of building damage due to earthquakes using aerial television images, H Hasegawa et al., 2000 [2]

In this work, researchers suggested the methodology of using the already available high definition television images to detect the damage in buildings in the aftermath of a disaster. The proposed approach used oblique angle (bird's eye) images taken by helicopters after natural disasters. The results of the interpretation were compared with the results of the ground survey to determine the feasibility of the system. For wooden buildings, moderate and severe damage was mostly recognized in the helicopter photos. However, only collapsed and severely damaged buildings were detected in non-wooden buildings.

The main drawbacks of this approach are the delay of having helicopters take the images and analyzing them in addition to the non-availability of helicopter images in rural areas.

2.2.3 A wireless modular health monitoring system for civil structures, J Lynch et al., 2002 [3]

This proposed method uses wireless sensing units for real-time structural response measurements via using available technologies in the marketplace. J Lynch et al. claimed that the suggested system is a low-cost alternative to traditional wire-based sensing systems. The readings are then sent wirelessly to a centralized data storage system. In the end, data processing is done to study the structural health of the structure.

In order to prove the concept, a small-scale model building was instrumented and excited with modal analysis performed on the time history response. Although the proposed system is claimed to minimize the installation cost, the cost is still too high to be used in all structures.

2.2.4 Use of satellite SAR intensity imagery for detecting building areas damaged due to earthquakes, M Matsuoka et al., 2004 [4]

This proposed method uses Synthetic aperture radar (SAR) Satellite because of its ability to record the backscattering coefficient regardless of weather condition or sun illumination and consequently can be used for better understanding damaged areas after disasters such as earthquakes and floods.

The suggested system compares before and after event satellite images to calculate physical earth coefficients. The researchers found that the backscattering coefficient and intensity correlation between the two attained values were much lowered in hard-hit areas. In addition, they evaluated building damage using the difference in the backscattering coefficient and correlation coefficient of the pre- and post-event ERS images. Figure 2.2 shows a schematic explanation of the observation process.

In conclusion, the investigators proposed that they developed an automated method to detect hard-hit areas based on the discriminant analysis, and compared the results that were found with a damage survey result. However, the biggest limitation of this approach is the fact that it can only detect partially or completely collapsed buildings.

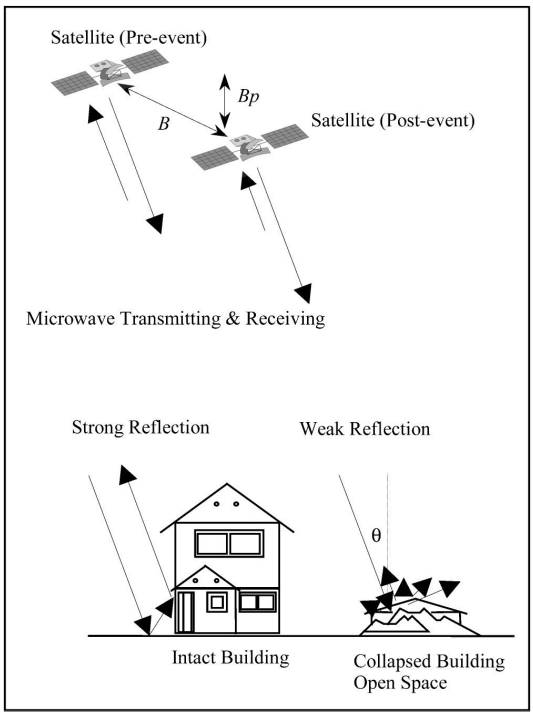


Figure 2.2: A schematic explanation of the observation process proposed in [4].

2.2.5 Application of a web-enabled real-time structural health monitoring system for civil infrastructure systems, S F Masri et al., 2004 [5]

This suggested method discussed in this paper targeted infrastructure systems, such as bridges. As a proof of concept, the researchers deployed the system into Vincent Thomas Bridge in Los Angeles, California. The bridge had been instrumented with 26 strong motion accelerometers to monitor its movement. It was shown that the analysis results of a recent earthquake correlate well with similar results previously obtained by other investigators who studied the bridge.

Because the sensors were installed in the 1970s, the sensor network was very sparse for applications that need to have an accurate estimate of the complex, three-dimensional motion. The researcher suggested using a broad spectrum of sensors to capture different measure-

ments of monitored infrastructure systems such as bridges. For instance, corrosion sensors, vision-based approaches, strain gauges, and direct displacement meters can provide a comprehensive diagnosis of the structure. In addition, the proposed system requires installing sensors in the monitored infrastructure system.

2.2.6 An advanced vision-based system for real-time displacement measurement of high-rise buildings, J Lee et al., 2012 [6]

This paper proposes a vision-based method for the real-time displacement measurement of high-rise buildings based on the successive estimation of relative displacements and rotational angles at several floors using a multiple vision-based displacement measurement systems. vision-based monitoring is very attractive since it calculates the displacement directly, avoiding the relatively large error of converting acceleration to displacement in the case of using accelerometers.

J Lee et al. claimed that the proposed vision-based system can successfully measure the horizontal displacement of a high-rise building by comparing the results that were collected for the experiment with reference readings from accurate sensors that were mounted in the same structure.

The system faces two main challenges. First, the need to installing cameras integrated buildings in advanced in addition to all the implemented wires and routers. Second, the system requires huge bandwidth for image processing.

2.2.7 Intelligent monitoring system on prediction of building damage index using neural-network, 2012 [7]

The aim of this work is to design a structural analysis software to classify buildings by their structural health status. This work proposes an intelligent monitoring system utilizing artificial neural network (ANN) for structural health assessment. In addition, the system provides an alert and notifications system to inform the public about the status of the damage. Figure 2.3 shows the suggested system framework.

The technique that was used in the ANN system is supervised learning, which means dividing the data set into 2 sets, one for learning and the other for testing. The dataset was a combination of previous earthquake readings and readings from accelerometers where no earthquake had occurred.

The proposed system added an intelligent approach for structural health detection to an already developed structural health assessment system. The main drawbacks of this approach are the high cost and the need to pre-install accelerometer in the targeted buildings.

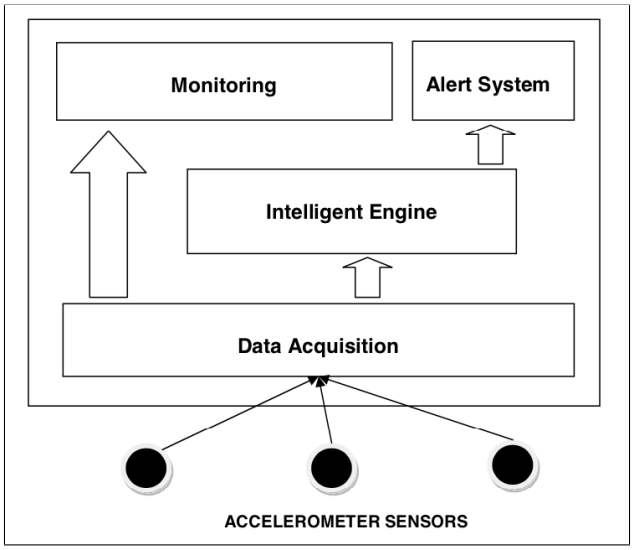


Figure 2.3: An ANN structural health monitoring framework proposed in [7]

2.3 Summary

Previous work in automating earthquake monitoring has been studied in this chapter. Table 2.1 shows summarizes major automated earthquake monitoring systems currently working.

Table 2.1: Summary of major automated earthquake monitoring systems

	Year Lunched	Sensor Type	Requires Installation?	Detection Type	Goal
Community Seismic Network (CSN)	2011	cheap MEMS sensors, suggested using phones sensors	yes, sensors connected to user computers	Threshold	Early Warning System (EEW)
updated CSN	2017	MEMS sensors	yes, packaged box contains a sensor, a computers, and backup battery	ML algorithm	EEW System and study buildings behavior
iSkake	2013	phones sensors	no	Threshold	study earthquakes
MyShake	2016	phones sensors	no	ML algorithm	EEW System

In addition, Several methodologies used for structural health monitoring have been studied. Table 2.2 gives a hint on the most common methodologies.

Table 2.2: Summary of the most common structural health monitoring methods

	Year Lunched	Sensor Type	Requires Installation?	Goal
GPS Monitoring	1995	GPS receivers	yes, GPS receiver attached to structure	Structural health Monitoring (SHM)
Arial Images Monitoring	2000	-	no	SHM
Warless Local Sensors Monitoring	2002	Accelerometers	yes	SHM
Satellite Monitoring	2004	-	no	SHM
Infrastructure Monitoring	2004	Already deployed sensor	yes	SHM
vision-based Monitoring	2012	cameras	yes	SHM
Intelligent algorithms Monitoring	2012	Accelerometers	yes	SHM

Chapter 3

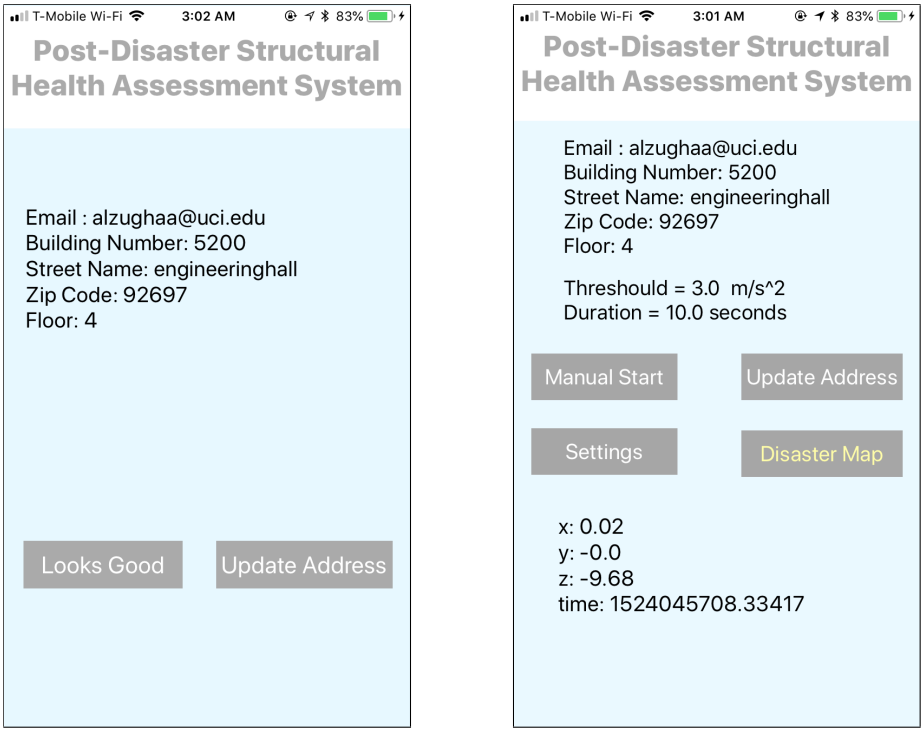
System Overview

The system was designed using client-server architecture. The user application is used to detect, store and then send recorded sensor data to a centralized cloud server. Then, the server sorts, organizes and stores data in a database. When requested by the user, the server computes the IDR value for each building and classifies the building to IO, LS or CP. Then, the server generates a map that displays buildings tagged with their structural health status. The rest of the chapter is organized as follows. Section 3.1 gives an overview of the mobile-phone application and the earthquake detection criteria. Section 3.2 covers server's architecture. Finally, the classification process is presented in section 3.3.

3.1 Client Mobile-phone Application

As part of this work, we developed a mobile-phone application that detects an earthquake and sends sensor readings to a centralized cloud computing server along with the exact time, device ID, user email address, building's street address, and floor number. The first time the app is installed, the user is asked to manually enter an email address and the location

information to be saved in a local database for future use. That process not only reduces the app power consumption by not requesting GPS location but also avoids any additional errors due to mis-geolocating adjacent buildings. When the app is restarted, the stored information will be displayed and the user will have the option to update it. Screen-shot of the iOS application are in Figure 3.1.



(a) Welcome page (b) Main UI

Figure 3.1: Screenshots of the Client Application. The seismic sensing is performed silently in the background. The information is fed back to the user in the form of a disaster map.

The app consists of 3 modes: steady mode, trigger mode, and streaming mode. The app has to enter a steady state before it can be active and available to get excited by an earthquake to avoid any additional noise in the reading. The app enters the steady mode when the absolute sensed acceleration in the x-y direction is below a certain threshold for certain time window. Trigger stage starts once an earthquake is detected,i.e. predetermined threshold ($0.1g$) has been crossed in the x-y direction. This is the same technique that has been used in iShake [22]. The app stores readings for a specific duration. After that, the app begins

sending the recorded event information and readings to the centralized cloud-server. An overview of how the app works is shown in Figure 3.2.

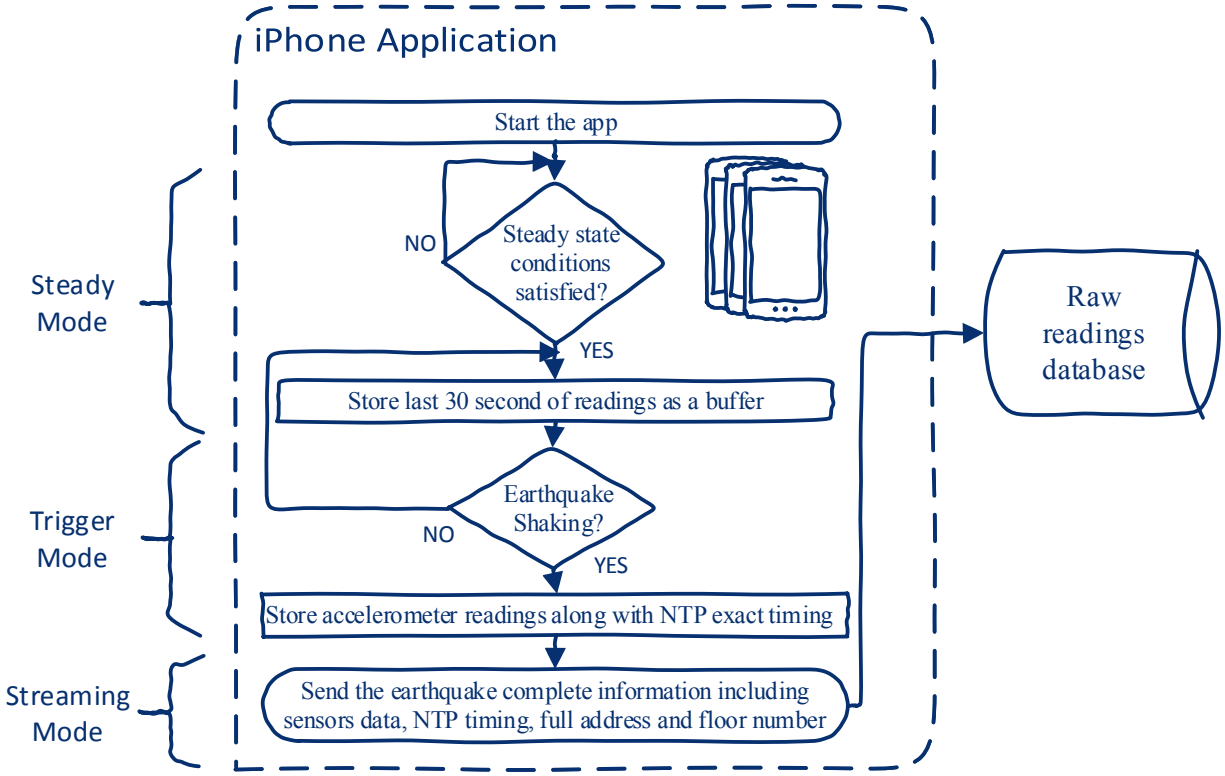


Figure 3.2: Mobile-phone Application Flowchart. The left hand side shows different application modes.

The detailed detection mechanism and signal processing of the mobile-phone application is presented in Algorithm 1.

3.1.1 Sampling Rate

It is well known that the sampling rate of the accelerometer affects displacement calculation. The error decreases with the increase of the sampling frequency. According to Apple developers documentation, The maximum frequency at which you can request updates is

Algorithm 1 Mobile-phone Application

INPUT *shaking, and location information*
OUTPUT *acceleration as a function of time*
for $i \leftarrow 1, 2, ..., f_s * T_{steady}$ **do** ▷ f_s is the sampling rate, and T_{steady} is the pre-defined duration of the steady state condition
    $a_{x_i}, a_{y_i}, and\ a_{z_i} \leftarrow$ accelerometer reading in the x-axis, y-axis, and z-axis, *respectively*
    $a_{xy_i} \leftarrow \sqrt{a_{x_i}^2 + a_{y_i}^2}$
end for
while $max\{a_{xy}\} < \Theta_{steady}$ **AND** $a_{z_i} > 9.0$ **do** ▷ Θ_{steady} is the pre-defined threshold of the steady state condition
    if $a_{xy} > \Theta_{detection}$ **then** ▷ $\Theta_{detection}$ is the pre-defined earthquake detection threshold
        for $i \leftarrow 1, 2, ..., f_s * T_{storing}$ **do** ▷ T_{steady} is the pre-defined duration of the earthquake readings
            $a_{x_i}, a_{y_i}, and\ a_{z_i} \leftarrow$ accelerometer reading in the x-axis, y-axis, and z-axis, *respectively*
            $a_{t_i} \leftarrow$ current NTP epoch time
        end for
    else
        Check steady state conditions again
    end if
end while
Raw readings database $\leftarrow a_x$, a_y, a_z, a_t, and location information

hardware-dependent [23]. In our experiments, the available phones were iPhone 5 and 5S. The highest sampling frequency for both devices is 100 Hz, so this was used in the experiment. This frequency is twice as much as what was used in MyShake [10]. Less error is expected when more advanced recently developed phones, such as iPhone X are used.

3.1.2 Pre-Trigger

If the data is recorded at the start of the trigger time, the prior data will be lost. It is worth noting that an error at the beginning of the acceleration time window is amplified by double integration, which is needed to calculate the displacement (see section 3.3.2 for the details). The error in the displacement without adding pre-trigger data compared with including pre-trigger data is discussed in the section 4.1.3. For that reason, a moving buffer was added to store accelerometer data for 30 seconds. The stored data is then sent to the server when the app is triggered. The first 25 seconds of the pre-trigger data could be used for noise characterization which will be helpful for the aggregation of the readings by giving more weight to less noisy readings.

3.1.3 NTP-Time Synchronization

Calculating relative displacements (IDR) requires millisecond accuracy for precision phasing. Therefore, a synchronization technique is required across phones to avoid clock drift. At first, the internal mobile-phone's internal clock was used for reading synchronization. However, the error in relative displacement calculation was slightly more than expected. For that reason, the proposed application uses *Network Time Protocol* (NTP) timing to ensure an accurate simultaneous reference for all devices [24], [25].

NTP is the Internet protocol used to synchronize the clocks of computers to some time reference. The time reference used in the proposed system is epoch time which is the total number of seconds since midnight of January 1, 1970. The error in calculating the displacement using the mobile-phone internal clock for Synchronization compared with using NTP clock is discussed in the section 4.2.3.

3.1.4 Running in the Background

One key feature of the application is the ability to detect earthquakes while running in the background. In other words, even if the user is not directly interacting with the application interface the app is still sensing the acceleration and can switch between the internal modes.

3.2 EC2 Cloud Server

Using a cloud server has major advantages over using a standard server. One advantage is the scalability of the cloud server which is crucial property in seismic related systems because it is hyperactive for a short period of time during and after an event [26]. Low operational cost is another attractive feature of cloud servers compared to standard servers. In addition, in cloud services maintenance and back up are usually offered by the service provider.

Amazon Web Services (AWS) is one of the major cloud computing service providers which make it suitable for the proposed system. An Amazon *Elastic Compute Cloud* (EC2) is used as a base for all computations and processes needed. An overview of how the EC2 server is presented in Figure 3.3.

A quick look at procedures done by the server is shown in Algorithm 2.

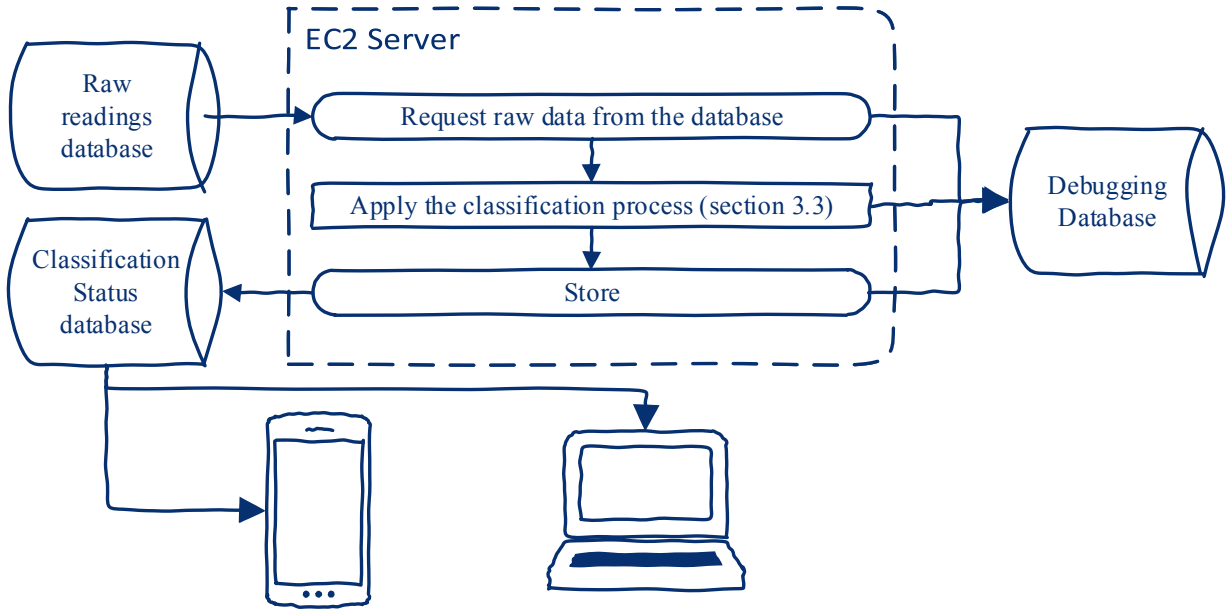


Figure 3.3: The EC2 Cloud Server Flowchart

Algorithm 2 The EC2 Cloud Server

1: **INPUT** *accelerometer readings, and building heights*
2: **OUTPUT** *detailed disaster map*
3: *Local memory* ← *Raw readings database*
4: **for** *Buildings in the targeted area* **do**
5: *Structucal health status* ← *Classification algorithm, Raw readings* ▷ see section 3.3
6: *Disaster map annotation* ← *Structural health status, Building's location*
7: **end for**

3.2.1 MySQL Database

The system uses the MySQL database because of its ability to store and organize thousands of readings to be called back when needed. The data is organized by zip code, street address, and floor number. The system consists of 3 different databases. One database is used to store raw sensor data received from the app along with the location information. Another database used to store the resultant IDRs for each building along with the exact time and date of the incident (earthquake). The last database is used as checkpoints for system debugging.

Raw Readings Database

The raw readings database is used to store raw sensor data received from the app along with the NTP timing of the readings. Location, date, and time of the incident are registered as well. A sample of the raw readings database is given in Figure 3.4.

ID	deviceID	buildingNumber	streetName	zipCode	floor	x	y	z	rtime	reading	readingDate	readingTime
25	ahmed1	5200	engineeringhall	92697	4	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-17	13:01:32
26	ahmed2	5200	engineeringhall	92697	3	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-17	13:01:38
27	ahmed6	916	engineeringtower	92697	10	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-18	00:16:17
28	ahmed7	916	engineeringtower	92697	9	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-18	00:16:21
29	ahmed10	6210	donaldbrenhall	92697	2	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-18	00:36:19
30	ahmed11	6210	donaldbrenhall	92697	3	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-18	00:36:16
31	ahmed22	1200	rowlandhall	92697	5	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-18	00:45:57
32	ahmed20	1200	rowlandhall	92697	4	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-18	00:45:57
33	alzughaa@uci.edu	5200	engineeringhall	92697	4	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-18	11:07:08
34	ahmed30	20	pacifica	92618	2	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-18	07:10:19
35	ahmed31	20	pacifica	92618	3	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-18	07:10:17
36	ahmed41	40	pacifica	92618	2	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-18	07:15:53
37	ahmed40	40	pacifica	92618	3	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-18	07:15:51

Figure 3.4: Raw Readings Database. BLOB means a very long array is stored in this location.

Classification Status Database

The classification status database is used to store the resultant IDRs for each building along with the exact time and date of the incident (earthquake). A sample of the classification status database is given in Figure 3.5.

ID	Address	IDR	IDRIndex	incidentDate	incidentTime
8703	5200engineeringh...	0.6034	849	2018-04-17	12:31:37
8704	5200engineeringh...	1.4441	969	2018-04-17	12:31:48
8705	5200engineeringh...	-1.0147	709	2018-04-17	12:31:58
8706	5200engineeringh...	-0.5624	964	2018-04-17	12:32:07
8707	5200engineeringh...	-2.3466	910	2018-04-17	12:32:17
8708	5200engineeringh...	0.5817	280	2018-04-17	12:32:27
8709	5200engineeringh...	0.8836	913	2018-04-17	12:32:38

Figure 3.5: Classification Status Database

Debugging Database

The debugging database provides checkpoints. Debugging database A is used to compare the received data to the data that was sent by the app. Debugging database B is used to ensure that the displacement calculation was done correctly. Finally, debugging database C is used to check the synchronization and relative displacement processes. These databases are displayed in Figures 3.6, 3.7 and 3.8, respectively.

ID	Address	floor	readings	incidentDate	incidentTime
33114	16201pintado92618	1	BLOB	2018-04-01	09:22:35
33115	16201pintado92618	2	BLOB	2018-04-01	09:22:35
33116	162pintado92618	2	BLOB	2018-04-01	09:22:35
33117	5200engineeringhall92697	4	BLOB	2018-04-01	09:22:35
33118	5200engineeringhall92697	3	BLOB	2018-04-01	09:22:35
33119	16201pintado92618	1	BLOB	2018-04-01	09:22:41

Figure 3.6: Debugging Database A

ID	Address	floor	readingsAfterCumsum	netDisplacmentX	netDisplacmentY	maxDisplacmentX	maxDisplacmentY	incidentDate	incidentTime
33015	162pintado92618	2	BLOB	-0.175737189579642	-0.0694187250230462	0.176395202821	0.0737178407433	2018-04-01	09:21:08
33016	5200engineeringhall92697	4	BLOB	0.00508359960227302	0.00643587520445081	0.00508359960227	0.0064377618219	2018-04-01	09:21:08
33017	5200engineeringhall92697	3	BLOB	-0.0378356023481559	-0.00935588886638018	0.0378356023482	0.0093616005024	2018-04-01	09:21:08
33018	16201pintado92618	1	BLOB	-0.050036023181155	-0.0580493883924776	0.050075462847	0.0582931273126	2018-04-01	09:21:12
33020	162pintado92618	2	BLOB	-0.175737189579642	-0.0694187250230462	0.176395202821	0.0737178407433	2018-04-01	09:21:12
33021	5200engineeringhall92697	4	BLOB	0.00508359960227302	0.00643587520445081	0.00508359960227	0.0064377618219	2018-04-01	09:21:12
33022	5200engineeringhall92697	3	BLOB	-1.64993311407645	-0.615561157979429	1.64993311408	0.615561157979	2018-04-01	09:21:12
33023	16201pintado92618	1	BLOB	-0.050036023181155	-0.0580493883924776	0.050075462847	0.0582931273126	2018-04-01	09:21:25
33025	162pintado92618	2	BLOB	-0.175737189579642	-0.0694187250230462	0.176395202821	0.0737178407433	2018-04-01	09:21:26
33026	5200engineeringhall92697	4	BLOB	-0.314685778298452	-0.537327618499871	0.314685778298	0.5373276185	2018-04-01	09:21:26
33027	5200engineeringhall92697	3	BLOB	-0.0807921362402602	-0.0192160153140474	0.0807970372427	0.0192215673796	2018-04-01	09:21:26

Figure 3.7: Debugging Database B

ID	Address	Delay	floor1	reading1	reading1x	reading1y	reading1z	reading1t	floor2	reading2	reading2x	reading2y	reading2z	reading2t	difference	incidentDate	incidentTime
15646	16201pintado92618	[0]	1	BLOB	BLOB	BLOB	BLOB	BLOB	2	BLOB	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-01	09:22:05
15647	5200engineeringhall92697	[0]	3	BLOB	BLOB	BLOB	BLOB	BLOB	4	BLOB	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-01	09:22:11
15648	16201pintado92618	[0]	1	BLOB	BLOB	BLOB	BLOB	BLOB	2	BLOB	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-01	09:22:11
15649	5200engineeringhall92697	[0]	3	BLOB	BLOB	BLOB	BLOB	BLOB	4	BLOB	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-01	09:22:17
15650	16201pintado92618	[0]	1	BLOB	BLOB	BLOB	BLOB	BLOB	2	BLOB	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-01	09:22:17
15651	5200engineeringhall92697	[0]	3	BLOB	BLOB	BLOB	BLOB	BLOB	4	BLOB	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-01	09:22:23
15652	16201pintado92618	[0]	1	BLOB	BLOB	BLOB	BLOB	BLOB	2	BLOB	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-01	09:22:23
15653	5200engineeringhall92697	[0]	3	BLOB	BLOB	BLOB	BLOB	BLOB	4	BLOB	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-01	09:22:29
15654	16201pintado92618	[0]	1	BLOB	BLOB	BLOB	BLOB	BLOB	2	BLOB	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-01	09:22:29
15655	5200engineeringhall92697	[0]	3	BLOB	BLOB	BLOB	BLOB	BLOB	4	BLOB	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-01	09:22:35
15656	16201pintado92618	[0]	1	BLOB	BLOB	BLOB	BLOB	BLOB	2	BLOB	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-01	09:22:35
15657	5200engineeringhall92697	[0]	3	BLOB	BLOB	BLOB	BLOB	BLOB	4	BLOB	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-01	09:22:41
15658	16201pintado92618	[0]	1	BLOB	BLOB	BLOB	BLOB	BLOB	2	BLOB	BLOB	BLOB	BLOB	BLOB	BLOB	2018-04-01	09:22:41

Figure 3.8: Debugging Database C

3.2.2 Web page

A list of the tested buildings along with their classification status is displayed on the system's website which is also hosted by the EC2 server. In addition, the website includes a map of the buildings tagged with their structural health state making it easier for the public to check their buildings in the aftermath of an event. A screen-shot of the web page is shown in Figure 3.9. Furthermore, the list and the map are available in the app for even simpler access.

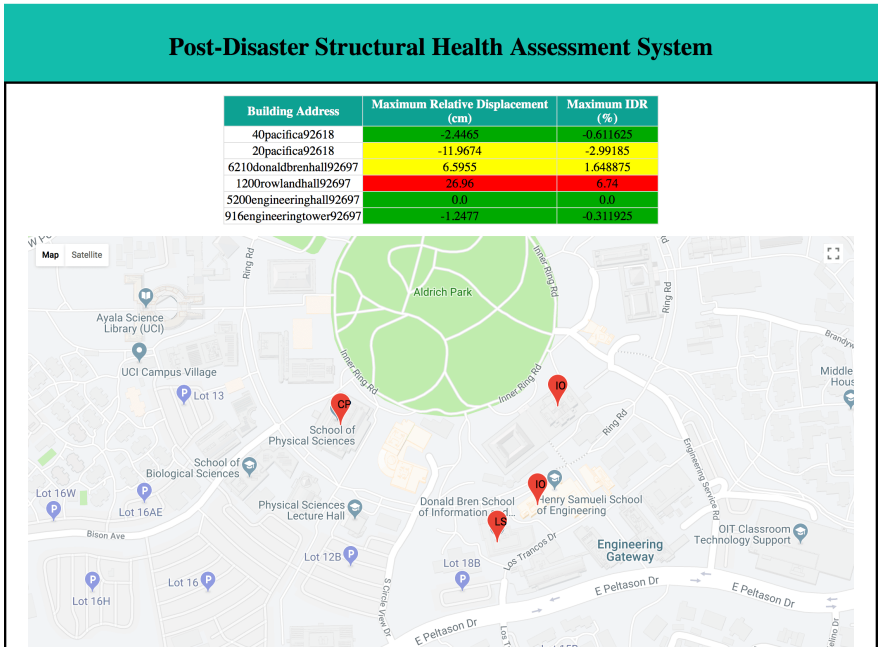


Figure 3.9: Website Screenshot. A detailed map is shown containing buildings tagged with their most likely structural health status.

Knowing the terrain surrounding the targeted building helps safety personnel on planing evacuation. A satellite view showing the terrain of the building surroundings is also provided in the webpage, as in Figure 3.10

The street-view is another option provided in the website to make it easier for the public to search for their building's structural health status. Figures 3.11, 3.12 present the street-view feature in the website and the phone's application, respectively.

Figure 3.10: Satellite view showing the terrain surrounding the targeted buildings.

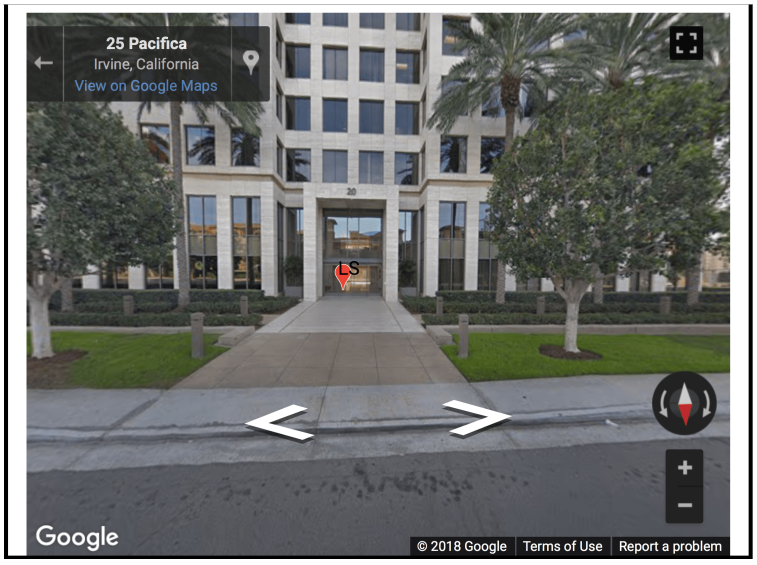
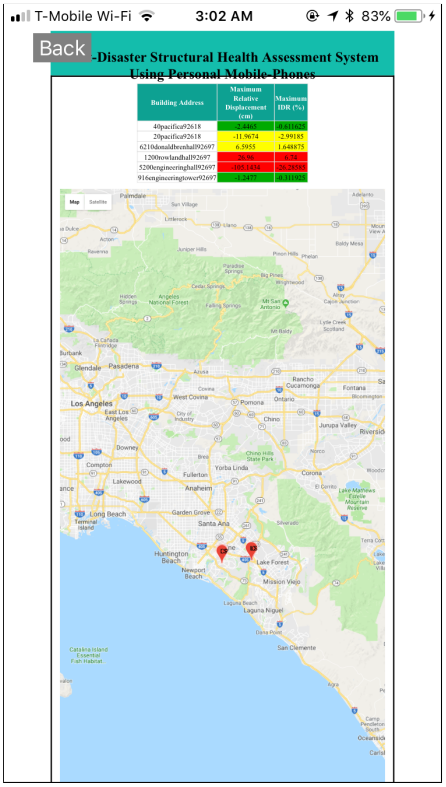


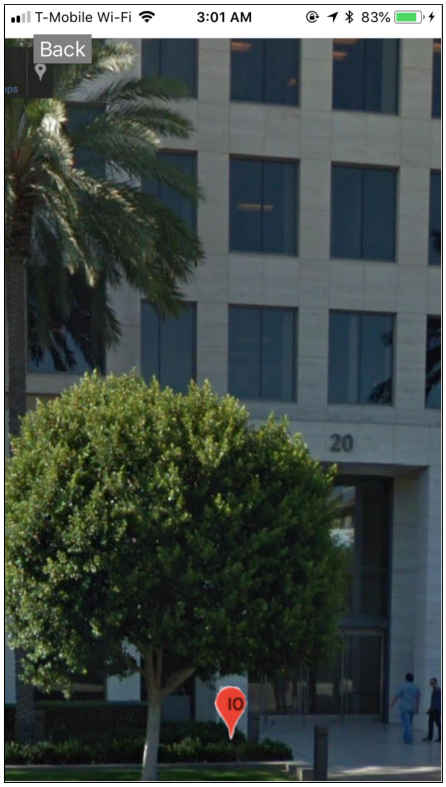
Figure 3.11: Street-view feature makes searching for buildings even easier for the public

3.3 Classification Process

The system loops over all the buildings that have active users during the disaster to categorize them into IO, LS or CP using FEMA standard [12]. For instance, an IDR over 4% for steel-moment frame buildings is considered CP. IDRs over 1% and below 4% are classified as LS, while IDRs less than 1% stated as being IO. Classifying a building is done through several stages. These stages are:



(a) a list of the tested buildings search-able by street address or zip code in addition to the detailed map containing buildings tagged with their structural health status



(b) the street view feature allows the user to use phone's orientation to clearly point to the building

Figure 3.12: Screenshots of the Client Application Disaster Map. The information is fed back to the user in the form of a disaster map.

3.3.1 Removing the Bias

Phone accelerometers usually have a constant bias. The first step is removing the bias for each axis of the acceleration which is estimated by long-term averaging. First the long-term mean (bias) needs to be calculated for each axis using equations 3.1 and 3.2.

$$\mu_x = \frac{\sum_{i=1}^{N} a_{x_{row}}[i]}{N} \tag{3.1}$$

$$\mu_y = \frac{\sum_{i=1}^{N} a_{y_{row}}[i]}{N} \tag{3.2}$$

Where N is the total number of readings i the given period, $a_{x_{row}}$ is the raw acceleration x-axis readings that were sent from the mobile-phone, and $a_{y_{row}}$ is the raw acceleration y-axis readings that were sent from the mobile-phone.

After that, the long-term mean (bias) has to be subtracted from each reading in every axis using formulas 3.3 and 3.4.

$$a_x[i] = a_{x_{row}}[i] - \mu_x \qquad \text{for } i = 1, 2, ..., N \tag{3.3}$$

$$a_y[i] = a_{y_{row}}[i] - \mu_y \qquad \text{for } i = 1, 2, ..., N \tag{3.4}$$

Where $a_x[n]$ is the x-axis acceleration reading after filtering out the bias, and $a_y[n]$ is the y-axis acceleration reading after filtering out the bias. The pseudo-code for removing the bias process is given in Algorithm 3.

Algorithm 3 Removing the Bias

1: **INPUT** $a_{x_{row}}[n]$, $a_{y_{row}}[n]$
2: **OUTPUT** $a_x[n]$, $a_y[n]$
3: $\mu_x \leftarrow mean(a_{x_{row}}[n])$
4: $\mu_y \leftarrow mean(a_{y_{row}}[n])$
5: **for** $i \leftarrow 1, 2, ..., N$ **do**
6: $\quad a_x[i] \leftarrow a_{x_{row}}[i] - \mu_x$ ▷ removing the long-term average from every reading in x-axis
7: $\quad a_y[i] \leftarrow a_{y_{row}}[i] - \mu_x$ ▷ removing the long-term average from every reading in y-axis
8: **end for**

3.3.2 Displacement Calculation

Finding velocity as a function of time $v(t)$ from raw accelerometer data $a(t)$ is done using equation 3.5

$$v(t) = v_0 + \int_0^t a(t)\ dt \tag{3.5}$$

The initial velocity $(v_0) = 0$ because of the steady mode conditions. Furthermore, displacement $d(t)$ is found using equation 3.6

$$d(t) = d_0 + \int_0^t v(t)\ dt \tag{3.6}$$

The initial displacement $(d_0) = 0$ because of the steady mode conditions. In short, integrating the acceleration results in velocity while double integrating the acceleration calculates displacement as in equation 3.7.

$$d(t) = \int_0^t \left(\int_0^t a(t)\ dt \right)\ dt \tag{3.7}$$

Integration is done numerically by summation in the case of digital (discrete) signals. Equations 3.5, 3.6, and 3.7 can be rewritten as follows

$$v[n] = dt\ \sum_0^n a[n] \tag{3.8}$$

$$d[n] = dt\ \sum_0^n v[n] \tag{3.9}$$

$$d[n] = (dt)^2 \sum_{0}^{n} \sum_{0}^{n} a[n] \tag{3.10}$$

where dt is the inverse of the sampling rate (f_s) ; $dt = 1/f_s$. $d[n]$ is the displacement at time n. To plot the displacement as a function of discrete time (n), the function *cumsum* (cumulative summation) is used to approximate integration. *cumsum* is defined in 3.11.

$$cumsum(g[n])[i] = \sum_{k=1}^{i} g[k] \qquad \text{for } i = 1, 2, ..., N \tag{3.11}$$

Where $g[n]$ is a discrete sequence. For example, the cumulative sums of the sequence $\{a, b, c, d, e, ...\}$, are $\{a, a + b, a + b + c, a + b + c + d, a + b + c + d + e, ...\}$. Furthermore, the double cumulative sums (double integration) of the sequence $\{a, b, c, d, e, ...\}$, are $\{a, 2a+b, 3a+2b+c, 4a+3b+2c+d, 5a+4b+3c+2d+e, ...\}$. As a result, the contribution of the beginning of the sequence is much higher than its tail. Similarly, the effect of the noise in the beginning of a signal distorts the resultant signal much more severely than the noise at the middle or end of it. Note that the result of the operation *cumsum* is a sequence. That is different than the operation *sum* which results in a single value.

The pseudo-code for displacement calculation procedures is given in Algorithm 4.

3.3.3 Data Synchronization

Data must be aligned before relative displacement calculation. NTP time that was sent from the app, as mentioned in section 3.1.3, is used as the base of alignment since the error

Algorithm 4 Displacement Calculation

INPUT f_s, $t[n]$, $a[n]$
OUTPUT $d[n]$
$f_s \leftarrow round(\frac{t[N]-t[1]}{N})$
$dt \leftarrow 1/f_s$
for $i \leftarrow 1, 2, ..., N$ **do**
    $cumsum(a[n])[i] = \sum_{k=1}^{i} a[k]$
    $cumsum(v[n])[i] = \sum_{k=1}^{i} v[k]$
end for
$v[n] \leftarrow dt * cumsum(a[n])$ ▷ $v_0 = 0$ because of the steady mode conditions
$d[n] \leftarrow dt * cumsum(v[n])$ ▷ $d_0 = 0$ because of the steady mode conditions

of the device's clock is irrelevant. The pseudo-code for reading synchronization is given in Algorithm 5.

Algorithm 5 Data Synchronization

INPUT $t[n]$, $d[n]$ ▷ $t[n]$ is the NTP epoch time for $d[n]$
OUTPUT $d[n]_S$ ▷ $d[n]_S$ is a synchronized version of $d[n]$
for $i \leftarrow 1, 2, ..., M-1$ **do**
    $Delay_i \leftarrow t[n]_i - t[n]_{i+1}$
    $d[n]_{S_i} \leftarrow d[n]_i, Delay_i$
end for

Figure 3.13 shows an example of the effect of using mobile-phone's internal-clock time for synchronization to relative displacement calculation. It is clear that mis-aligning the two readings has led to a large unreal relative displacement. In the the example in Figure 3.13, even though one reading is lagged by less than 15 ms, it has led to a 2 cm error, which is intolerable in IDR calculations.

On the other hand, the NTP alignment has led to a 2 ms mis-alignment. This 2 ms shift between the two readings has led to 0.2 cm relative displacement error as presented in Figure 3.14.

The improvement of using NTP time over the phone's clock as a function of the stored data duration is provided in Fig 3.15. It makes sense that the contribution of the synchronization

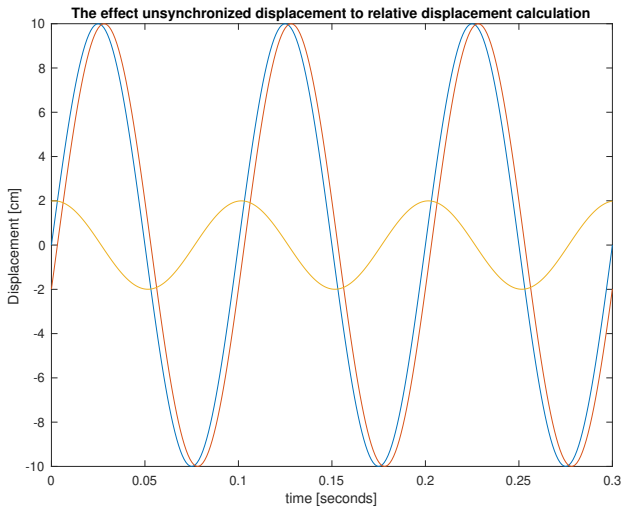


Figure 3.13: An example of the effect of using mobile-phone's internal-clock time for synchronization to relative displacement calculation.

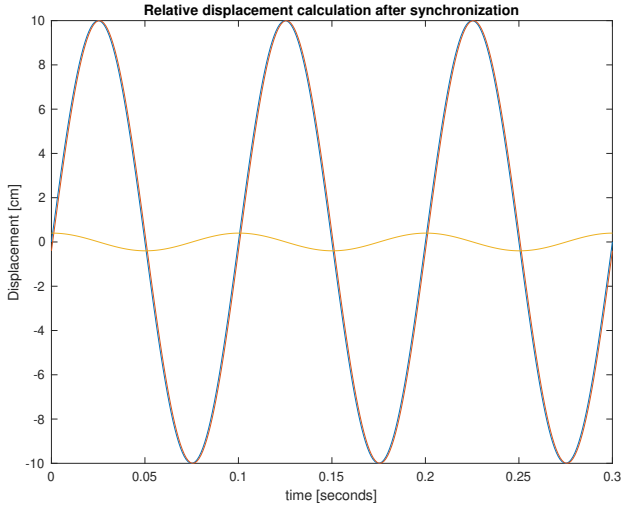


Figure 3.14: An example of the effect of using NTP time for synchronization to relative displacement calculation.

error to the total error is decreases with increasing duration since the error due to the accelerometer noise accumulates fast for long duration which minimizes the synchronization error portion of the total error.

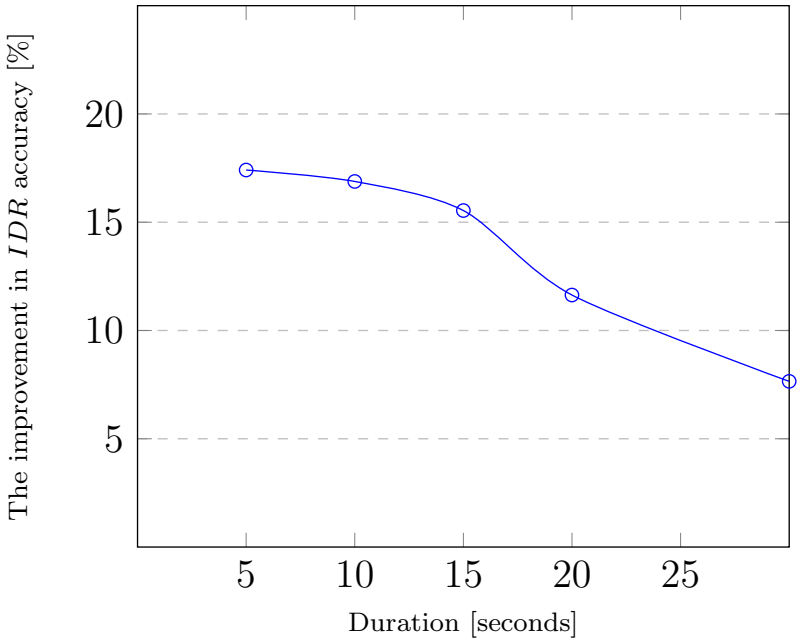


Figure 3.15: The improvement in The IDR accuracy of using NTP time over internal-clock time for synchronization

3.3.4 Relative Displacement Computation

The server loops over every adjacent floor in every building containing active devices during an earthquake. Then, the IDR is computed using equation (3.12).

$$IDR_i = max\left\{\left|\frac{d(t)_i - d(t)_{i+1}}{h}\right|\right\} \tag{3.12}$$

Where IDR_i is the maximum IDR value between floor i and floor $i+1$, $d(t)_i$ and $d(t)_{i+1}$ are the calculated displacements in meters for floor i and floor $i+1$, respectively, and h is the floor height in meters (typically, 4m for steel-moment frame buildings). In Digital signals, formula 3.12 can be rewritten as

$$IDR_i = max\left\{\left|\frac{d[n]_i - d[n]_{i+1}}{h}\right|\right\} \tag{3.13}$$

After that, an equivalent IDR value is calculated for every building using equation (3.14).

$$IDR_b = max\left\{IDR_i : i = 1, 2, ..., M-1\right\} \tag{3.14}$$

Where IDR_b is the maximum IDR value recorded between any adjacent floors in building b and M is number of floors in building b. Finally, a class is given to the building by comparing IDR_b with the FEMA reference values. The pseudo-code for relative displacement and IDR_b calculation procedures is given in Algorithm 6.

Algorithm 6 IDR calculation

INPUT h, $d[n]_S$
OUTPUT IDR_b
for $i \leftarrow 1, 2, ..., M-1$ **do**
    for $j \leftarrow 1, 2, ..., N$ **do**
        $d[j]_{i_{relative}} \leftarrow \frac{d[j]_{S_i} - d[j]_{S_{i+1}}}{h}$ ▷ where $d[j]_{i_{relative}}$ is the relative displacement between floors i and $i+1$
    end for
    $IDR_i \leftarrow max\left\{|d[n]_{i_{relative}}|\right\}$
end for
$IDR_b \leftarrow max\left\{IDR_m\right\}$ ▷ m = 1, 2, ..., M-1

The variance of the error in relative displacement is given in equation 3.15.

$$Var[d_i - d_{i+1}] = Var[d_i] + Var[d_{i+1}] - 2\ Cov[d_i, d_{i+1}] \tag{3.15}$$

Since the two devices are independent, Covariance ($Cov[d_i, d_{i+1}]$) = 0. Furthermore, $Var[d_i] = Var[d_{i+1}]$ assuming the cell-phone accelerometers are identical. Therefore,

$$Var[d_i - d_{i+1}] = 2\ Var[d_i]$$

and the standard deviation of the error in relative displacement $Std[d_i - d_{i+1}]$ is given in equation (3.16).

$$Std[d_i - d_{i+1}] = \sqrt{2}\ \ Std[d_i] \qquad for\ i = 1, 2, ..., n-1 \tag{3.16}$$

An overview of how the server classification process is given in Figure 3.16

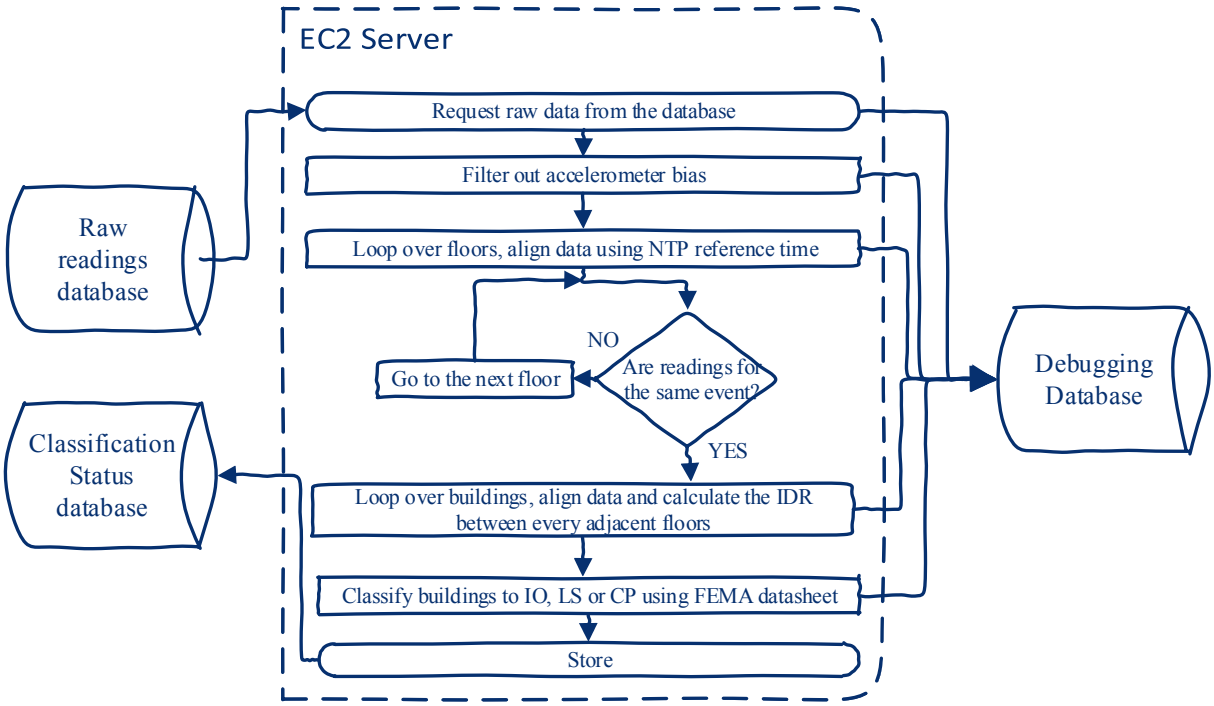


Figure 3.16: The EC2 Cloud Server Flowchart. The chart shows signal processing of received phone acceleration signals.

The pseudo-code for the classification process is given in Algorithm 6.

Algorithm 7 Structural health classification process

INPUT *accelerometer readings, and building heights*
OUTPUT *structural health status*
for *Building in the targeted area* **do**
    for *floors in the building* **do**
        $a_i[n] \leftarrow a_{i_{row}}[n]$ ▷ Filtering out the bias using Algorithm 3
        $d_i[n] \leftarrow a_i[n]$ ▷ the displacement is calculated for every floor using Algorithm 4
        $d[n]_{S_i} \leftarrow t[n]_i, d[n]_i$ ▷ reading synchronization using Algorithm 5
        $d[n]_{i_{relative}} \leftarrow d[n]_{S_i}, d[n]_{S_{i+1}}$ ▷ Relative displacement between every adjacent floors is calculated using Algorithm 6
        $IDR_i \leftarrow d[n]_{i_{relative}}$ ▷ Local IDR is calculated for every adjacent floors using Algorithm 6
    end for
    $IDR_b \leftarrow IDR_i$ ▷ Building IDR_b is calculated for The building using Algorithm 6
    $IO, LS, or\ CP \leftarrow IDR_b$ ▷ A classification status is given to the building using FEMA standards
end for

3.4 Summary

A post-earthquake automated assessment system was built using client-server architecture. For the client side, an iPhone application was designed for earthquake detection. Once the earthquake is detected, the application saves the accelerations of the shaking for a predefined duration. When the recording is finished, the detailed information about the earthquake is then sent to a database used to register the unprocessed application readings. The detailed information includes accelerometer readings, the NTP precise timing and location details.

On the other side, an Amazon cloud server was built for 4 purposes. First, storing raw readings that were sent through the application in a database. Second, implementing all the signal processing needed for structural health assessment.Third, saving building status along with location information in a database. Finally, visualizing the resultant classification into a detailed interactive map including buildings tagged with the most likely structural health status.

The system flowchart is presented in Figure 3.17

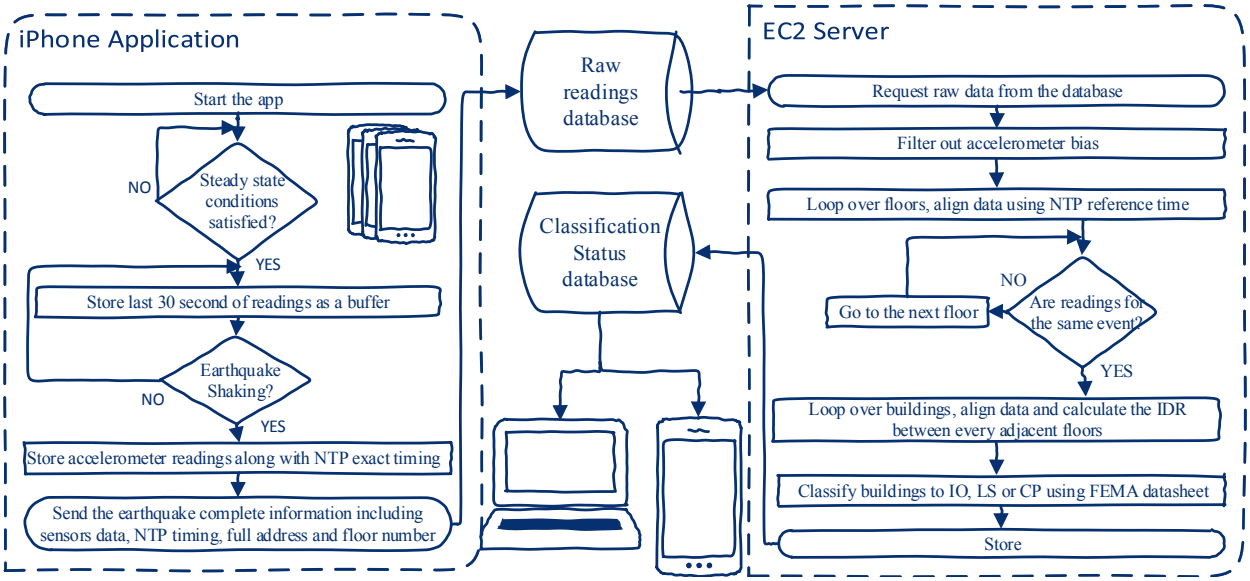


Figure 3.17: System Flowchart. The left hand side shows the mobile-phone application modes. The right hand side describes the signal processing of received phone acceleration signals.

Chapter 4

Experimental Validation

Several tests were conducted to determine the phone's ability to track phone displacement. A few more experiments were done in which different features were added to the system to test for improvement. The experiments were repeated for durations ranging from 5 to 30 seconds and for different settings. These settings are described below.

4.1 Preliminary Experiments

A few experiments were conducted to discover the mobile-phone accelerometer feasibility in earthquake detection and analysis.

4.1.1 Noise Characterization

Before building the system, a full characterization of mobile-phone accelerometer noise is needed to check the ability of the phone's accelerometer to sense an earthquake. First, In order to determine the noise level, the devices were left in a quit room to store accelerometer

readings overnight. The registered reading is the total noise which consists of the accelerometer's noise plus the everyday vibrations due to walking, door closing, etc. Figure 4.1 presents a 10-minute sample of the accelerometer's readings.

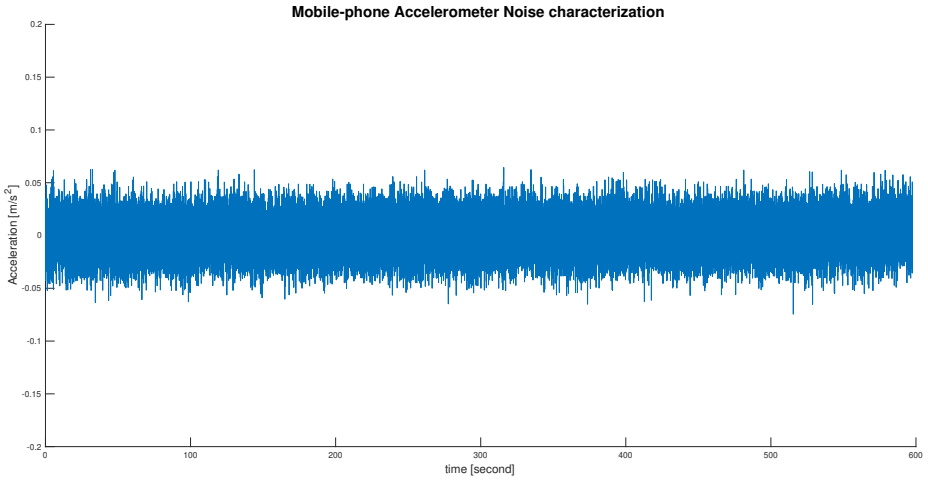


Figure 4.1: Pure Mobile-phone Accelerometer Noise for 10 minutes

Figure 4.2 shows the power spectral density of the stored acceleration. The noise level of the accelerometer of the tested phone was about -52 dB.

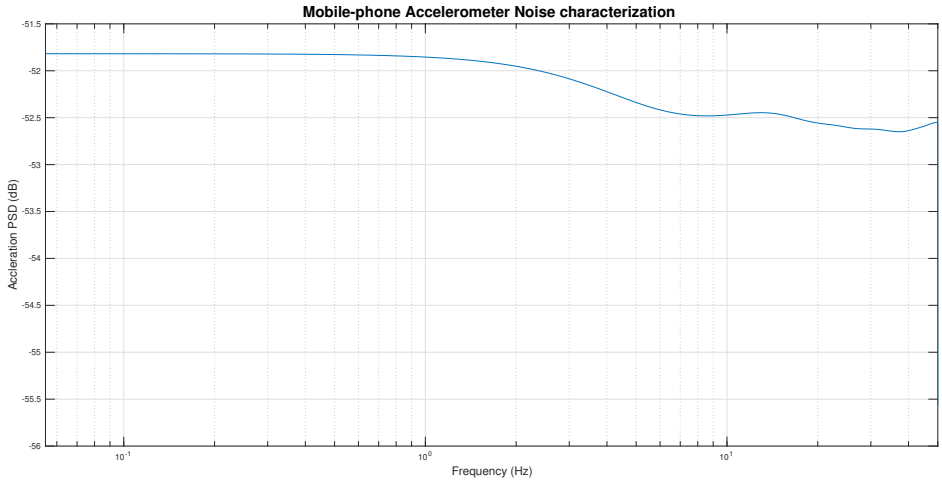


Figure 4.2: The Power Spectral Density of Mobile-phone Accelerometer Noise

The PSD of different magnitude earthquakes has been studied in [10] and presented in Appendix B. It has been used as a reference to determine which earthquake magnitude the phone's accelerometer is able to sense. It is clear that the mobile-phone's accelerometer is able to sense earthquakes that are magnitude 5 (M5) or larger. In fact, the stronger earthquakes are the ones that are focused on since they are the ones that cause sever damage to the structural health of buildings.

The noise floor of the seismic-grade accelerometer is provided in Appendix B in addition to the noise floor of a recently developed MEMS accelerometer, which can be mounted in a smart-phone in the future, when they are more economically feasible.

4.1.2 Removing the Bias

As mentioned in 3.3.1, a phone's accelerometer tends to slightly bias to a positive or a negative value. For example, in Figure 4.3 the recorded acceleration is biased by about 0.14 m/s^2 to the negative side.

The bias of mobile-phone accelerometers can severely damage displacement calculation since this bias accumulates very fast after double integrating the acceleration to calculate displacement. For instance, a 20 seconds sample of the biased accelerometer readings is presented in Figure 4.4 along with displacement that was calculated using the biased acceleration. Even though the bias was as little as 0.14 m/s^2, the error in the displacement became 2.8 m which is a completely intolerable error in seismology standards.

On the other hand, Figure 4.5 shows the fixed unbiased accelerometer readings in addition to the displacement that was calculated using the unbiased acceleration. The error in the displacement became about 0.03 m. This is a huge improvement compared the case of unfixed acceleration.

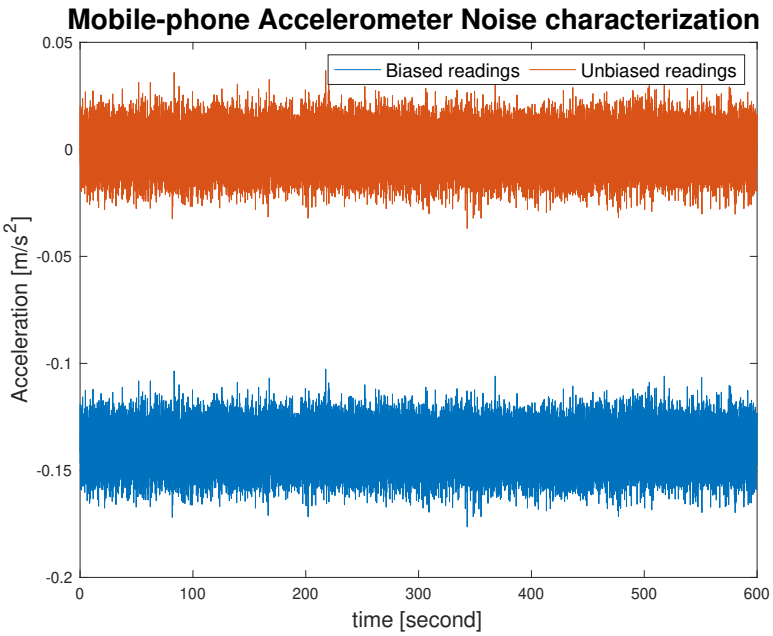


Figure 4.3: Biased vs. unbiased mobile-phone accelerometer noise

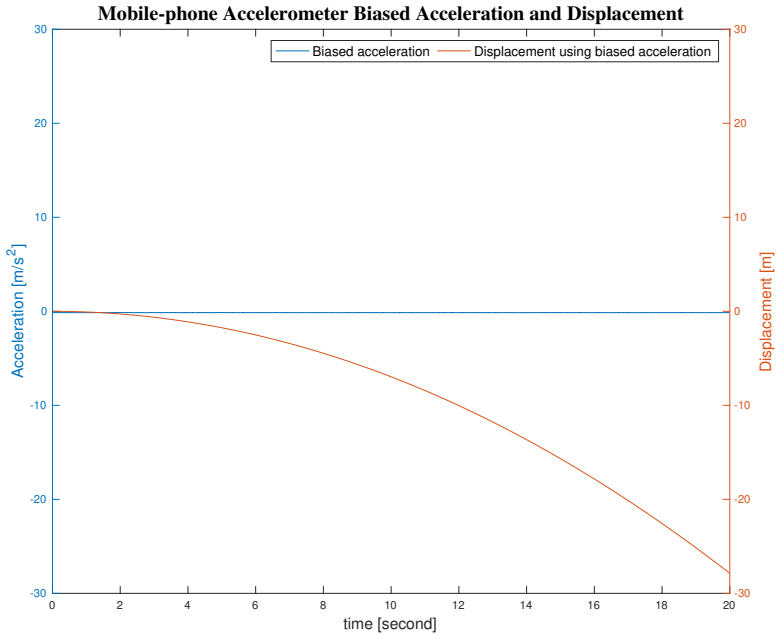


Figure 4.4: Biased mobile-phone noise acceleration vs. the displacement that was calculated using biased acceleration

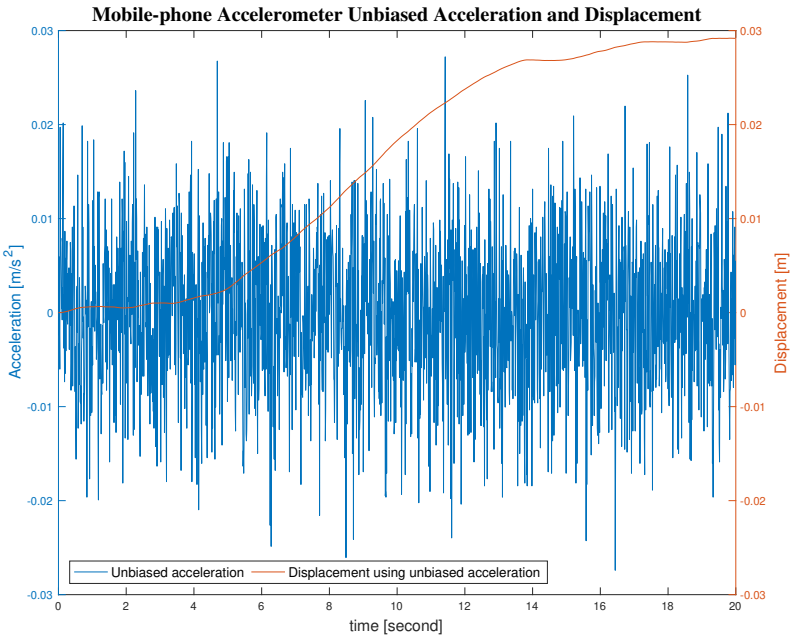


Figure 4.5: Unbiased mobile-phone noise acceleration vs. the displacement that was calculated using unbiased acceleration

4.1.3 Pre-Trigger Data

As stated in section 3.1.2, not including the pre-trigger data causes errors in displacement calculations since it does not take into account valuable pieces of the readings. An example of the displacement calculations difference between adding and not adding the pre-trigger readings is given in Figure 4.6. Not adding the pre-trigger data has added about 2 *cm* of error.

4.1.4 Low Pass Filter

As most of the features have been added to successfully improve the error, some additions did not have noticeable effect. One of those features is filtering the readings before any signal processing. The idea was removing the high frequency noise using a low pass filter.

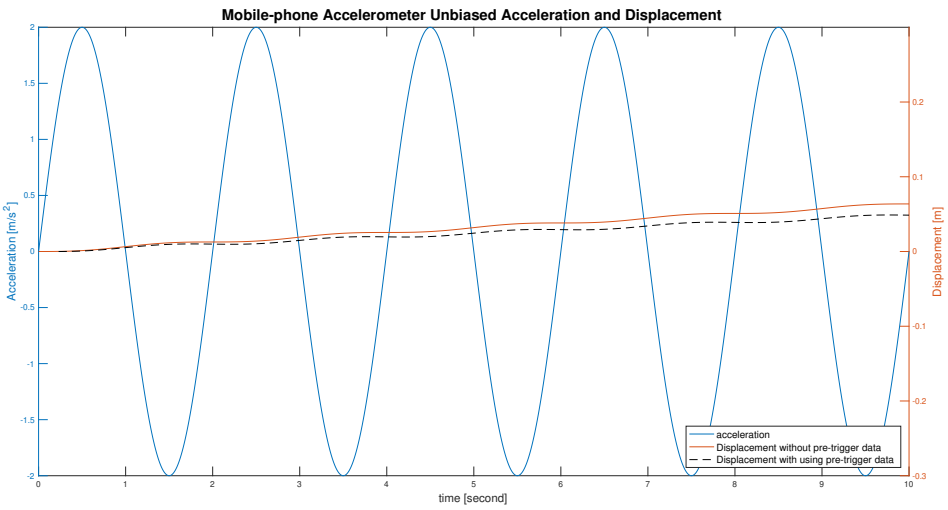


Figure 4.6: Unbiased mobile-phone noise acceleration vs. the displacement that was calculated using unbiased acceleration

However, it turned out that the effect of the high frequency noise was minimal. In addition, implementing the low pass filter added noise to the valuable low frequency earthquake acceleration.

4.2 Full System Experiments

A full system experiment was conducted where two phones are placed on two adjacent floors and manually triggered and kept not moving for the entire period. Furthermore, another full system test were conducted to calculate the relative displacement and the IDR of two moving devices. The applied shaking was identical , which means the relative displacement should be pure zero since the actual distance between the devices is constant.

Furthermore, several experiments were conducted to examine the effect of adding different features into the system. The two features that were studied in this section are one, the addition of the pre-trigger readings and two, relying on the NTP time for reading synchronization

instead of using the phone's internal clock. The reason a full system test was not conducted without fixing the mobile-phone's accelerometer bias is because the error is extremely large for estimating the displacement of one device (see section 4.1.2). This error will increase in the case of relative displacement calculations as explained in section 3.3.4.

4.2.1 Case I: At Rest

The first experiment conducted determines the effect that pure phone accelerometer noise has on the calculation of the relative displacement between two non-moving floors containing one device each. The two devices are triggered manually and left at rest for the whole experiment period. Ideally, the resultant relative displacement should be zero. However, the noise from phone accelerometer accumulated rapidly due to double integration. This error increased sharply as the the duration of the registered event increased. The error was relatively small (around 0.5%) for durations below 10 seconds. Therefore, it is not likely to cause misclassification of a building's health status. According to [9], about 66% of earthquakes that hit California have strong motion duration below 10 seconds.

4.2.2 Case II: Identical Motion Before Adding Pre-Trigger Data

Adding the pre-trigger data is a crucial step to ensure that we are not losing any shaking information. In fact, not using pre-trigger data worsens the IDR calculation by a factor of 5 making it intolerable even for shorter durations. Figure 4.8 shows the average of full system tests repeated 30 times for each duration. The error in the IDR calculation is found to be slightly more than 3% for 10 seconds stored acceleration which is unbearable in structural health classification.

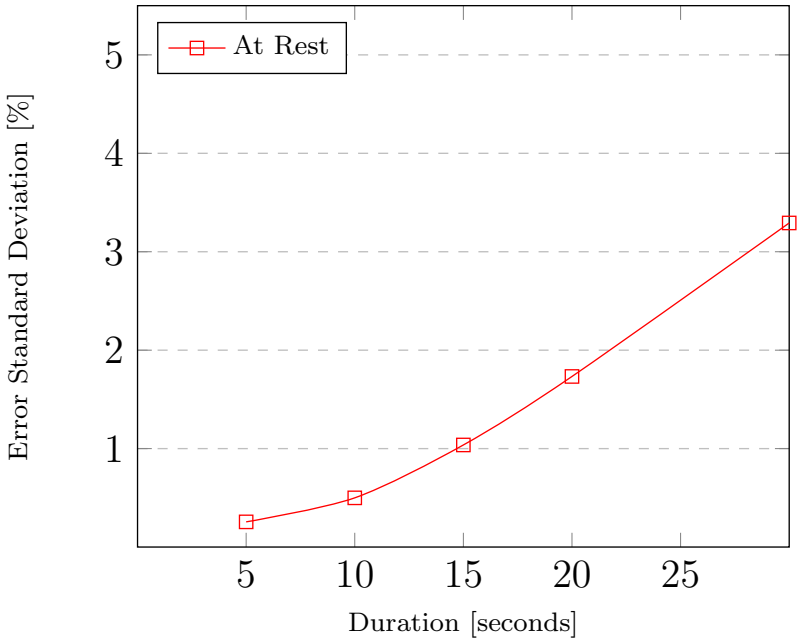


Figure 4.7: Error in IDR Using Smart-phone Accelerometer at Rest

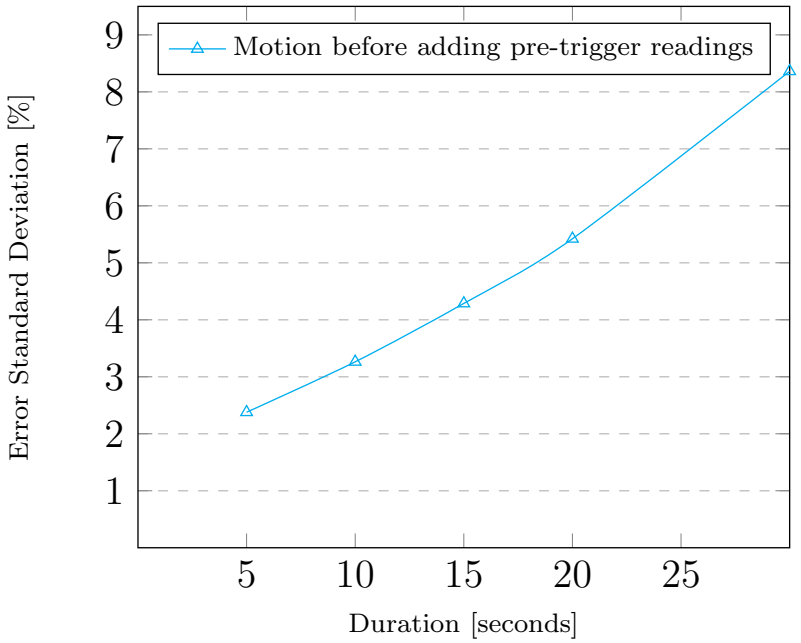


Figure 4.8: Error in IDR Using Smart-phone Accelerometer using Identical Motion Before Adding Pre-Trigger Data

4.2.3 Case III: Using Mobile-Phone's Internal Clock for Synchronization

Relying on the phone's internal clock slightly enlarges the error in the IDR calculation. As explained in section 3.3.3 the best available option for indoor synchronization is using NTP epoch time as a common reference for all devices in the system. Figure 4.9 displays the average of the repeated full system tests relying on the mobile-phone's internal clocks for data alignment. The IDR calculation error in the system in the case of using the internal clock was slightly larger than the case of using NTP time. In fact, using the phone's internal clock is still tolerable for durations less than 10 seconds.

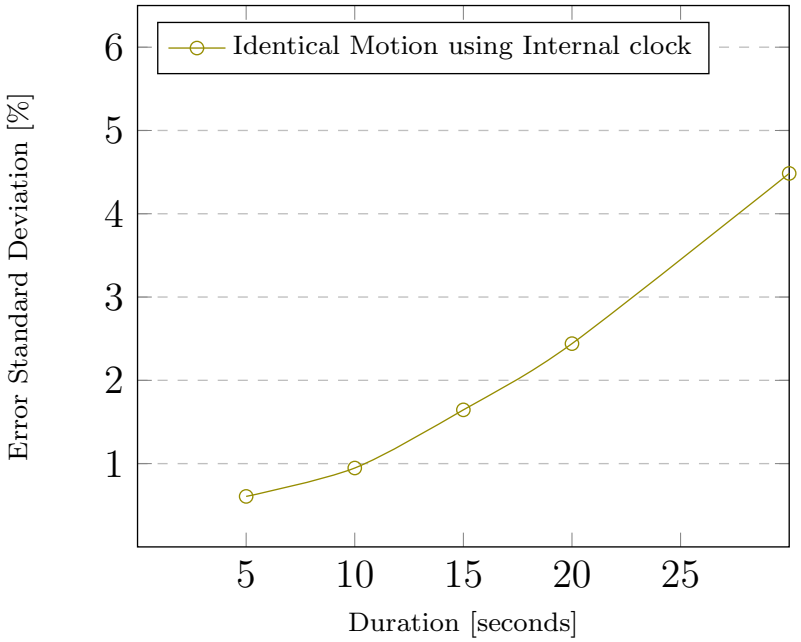


Figure 4.9: Error in IDR Using Smart-phone Accelerometer for Identical Motion Using Mobile-Phone's Internal Clock for Synchronization

4.2.4 Case IV: Identical Motion with Optimal Settings

The last test applies an identical shaking motion to the two phones. The two devices are both bonded to a horizontal shaking slider. Ideally, they should get triggered at the same time

and record the same acceleration. Additionally, since both phones are attached together, their relative displacement should be perfect zero. Consequently, any measured relative displacement is due to several other factors such as: time synchronization, trigger delay, and amplitude-dependent accelerometer noise in addition to the accelerometer noise. As shown in Fig. 4.11, the error is still tolerable for short durations (roughly 0.75% for periods shorter than 10 seconds).

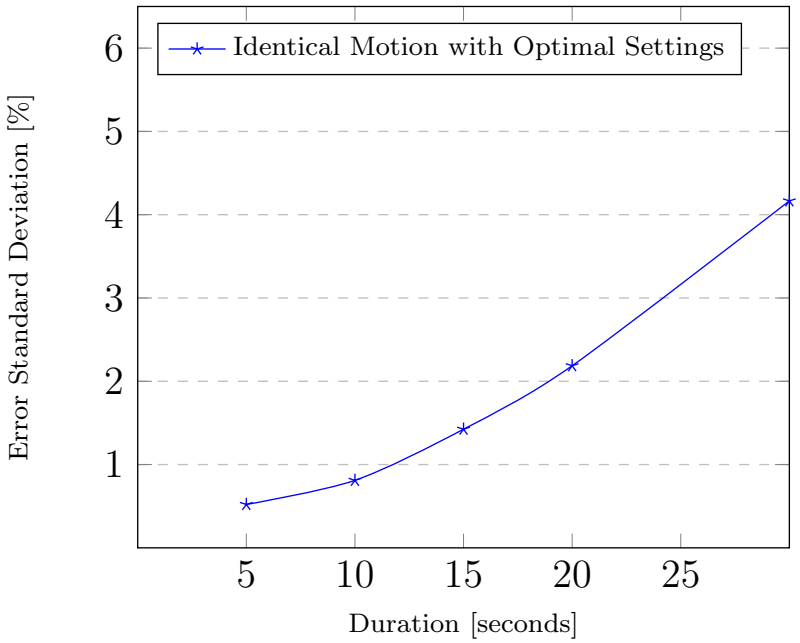


Figure 4.10: Error in IDR Using Smart-phone Accelerometer using Identical Motion with Optimal Settings

4.3 Readings Aggregation

Aggregating same-floor readings reduces the error in the relative displacement. As mentioned in section 3.1.2, the pre-trigger readings can be used for noise characterization. Noisy readings can be given less weight, resulting in a more-efficient aggregation. A more complex aggregation mechanism could be applied for even better results. Determining the effect of using different aggregation methods is out of the scope of this thesis.

4.4 Summary

To validate the feasibility of the system for structural health assessment, full system tests were conducted using different settings. The first experiment was determining effect of the noise of the mobile-phone's accelerometer in addition to the every day life sources, such as walking around the device which cause vibration that can be read by the accelerometer.

After that, few experiments were conducted to test possible advances through introducing different improvements into the system. One of the improvements added is including the pre-trigger data to avoid losing any valuable piece of readings. Relying on NTP time as a reference for synchronization instead of the phone's internal clock is another improvement that was studied.

Furthermore, a full system experiment was conducted where two phones were placed on two adjacent floors and manually triggered and kept stationary for the entire period.

Finally, full system tests were conducted to calculate the relative displacement and the IDR of two moving devices. The applied shaking was identical , which means the relative displacement should be pure zero since the actual distance between the devices is constant. Figure 4.11 shows the results of tests conducted on the system for every setting used.

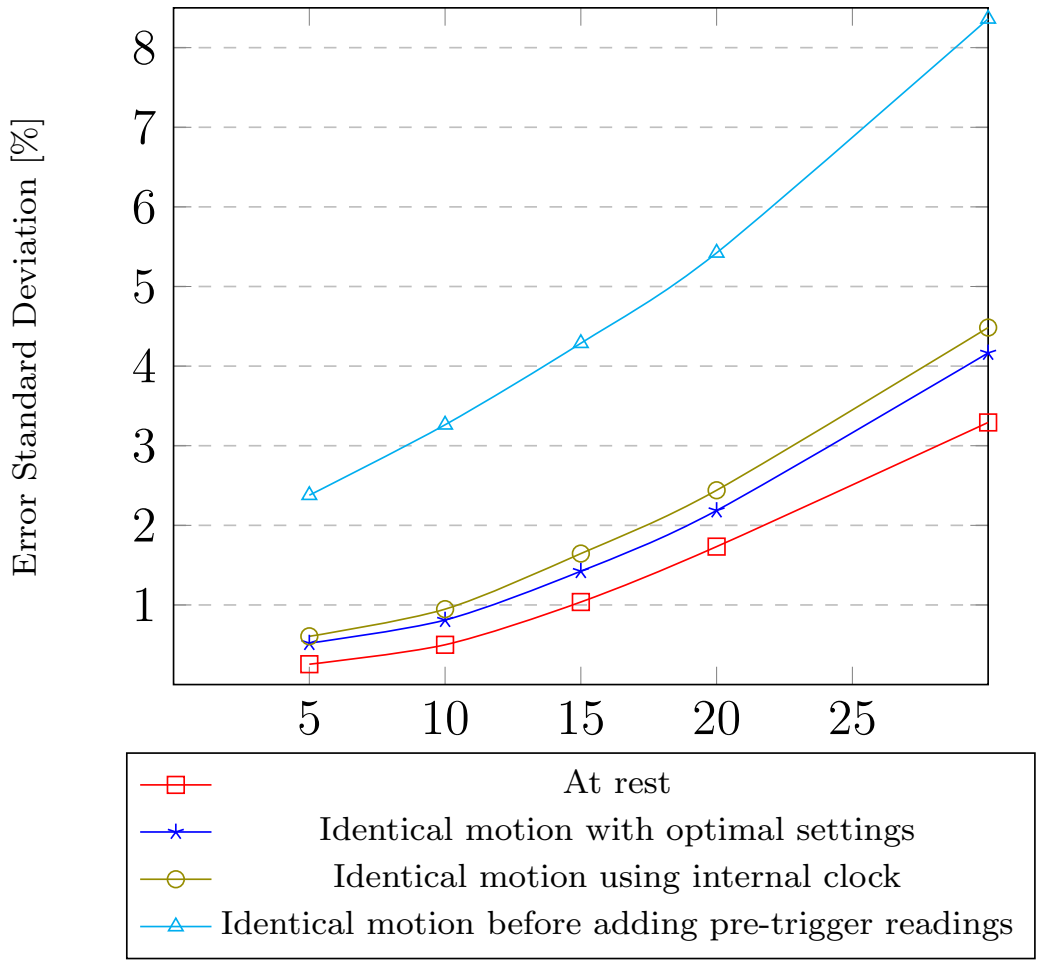


Figure 4.11: Error in IDR Using Smart-phone Accelerometer. Different settings and features were tested.

Conclusion

A post-disaster structural health assessment system has been developed using client-server architecture. A mobile-phone application is designed to detect and record earthquake acceleration. The stored data is then sent to a centralized cloud server which calculates the maximum IDR for every structure to be classified as either safe, needs further evaluation or unsafe according to FEMA standards. The resultant classification is then presented to emergency personnel as well as the to public in the form of a disaster map with buildings tagged by their most likely health state. Promising results were achieved for short duration events which showed error in IDR less that 0.75% for an event period shorter than 10 seconds. Future directions of the project include applying machine learning (ML) earthquake detection algorithms, implementing noise cancellation algorithms, and using the mobile-phone's barometer to determine the user's floor.

Appendix A

Earthquake Strong-Motion

Most of the damage caused by earthquakes is happening during the strong shaking of the earthquake. Addressing the duration of earthquake's strong shaking provides a great reference for the periods we should focus on. The strong motion duration calculated based on acceleration magnitude or cumulative energy obtained by integrating squared acceleration.

The strong motion duration of 140 California earthquake records was calculated in [9]. The cumulative density function (CDF) of strong motion duration is presented in Figure A.1.

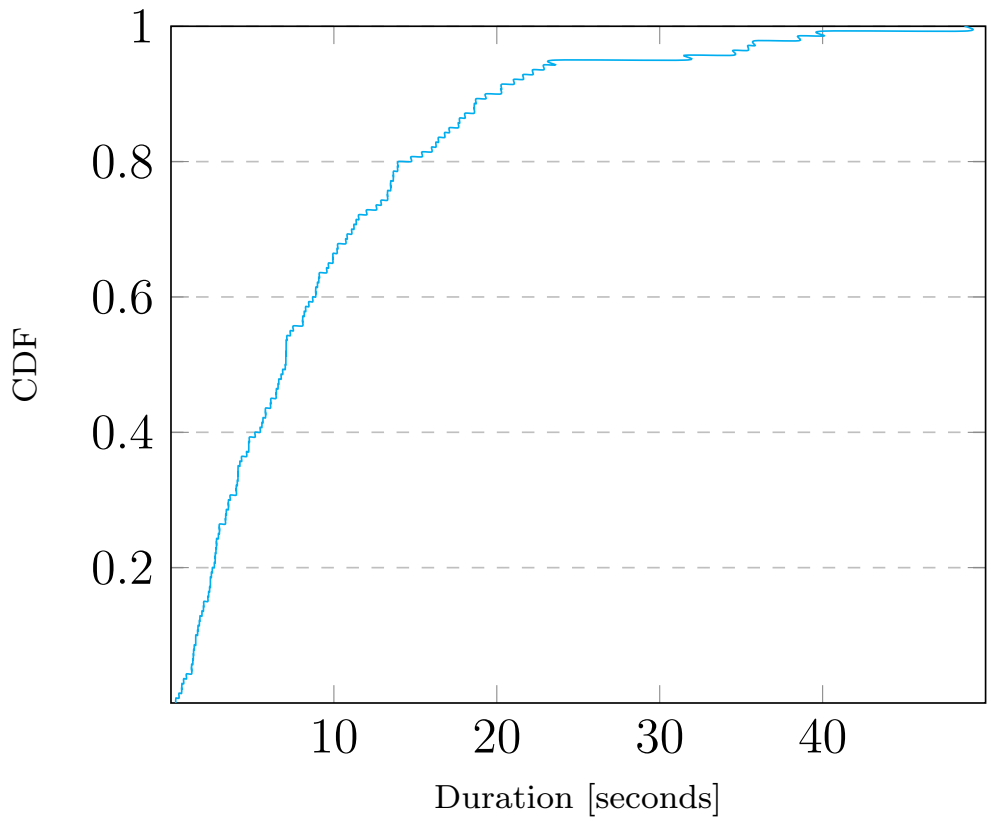


Figure A.1: CDF of strong motion duration of 140 horizontal components of earthquake ground motion recorded in California reported in [9].

Appendix B

Power Spectrum Density of Different Magnitudes Earthquakes and the Noise Floor of Seismic Grade Accelerometers

B.1 PSD of Different Magnitudes Earthquakes

To better now the ability of mobile-phone accelerometers in seismic field, the noise floor of accelerometers have to be compared with the amplitudes of earthquakes of different magnitudes. This comparison has been previously done in MyShake [10] and presented in Figure B.1. Phone accelerometers are able to sense the shaking for magnitude 5 (M5) or larger earthquakes at distance of 10 km away from the phone in the frequency range of 1 to 10 Hz.

B.2 Noise Floor for Seismic Grade Accelerometers

An example of the seismic grade accelerometers used today in seismic monitoring is presented in Figure B.1. It can clearly be seen that those accelerometers are able to capture much weaker earthquakes for almost all earthquake frequency range. The noise floor of the seismic grade accelerometer is around $-120\ dB$ compared to $-40 \sim -60\ dB$ for mobile-phone accelerometers.

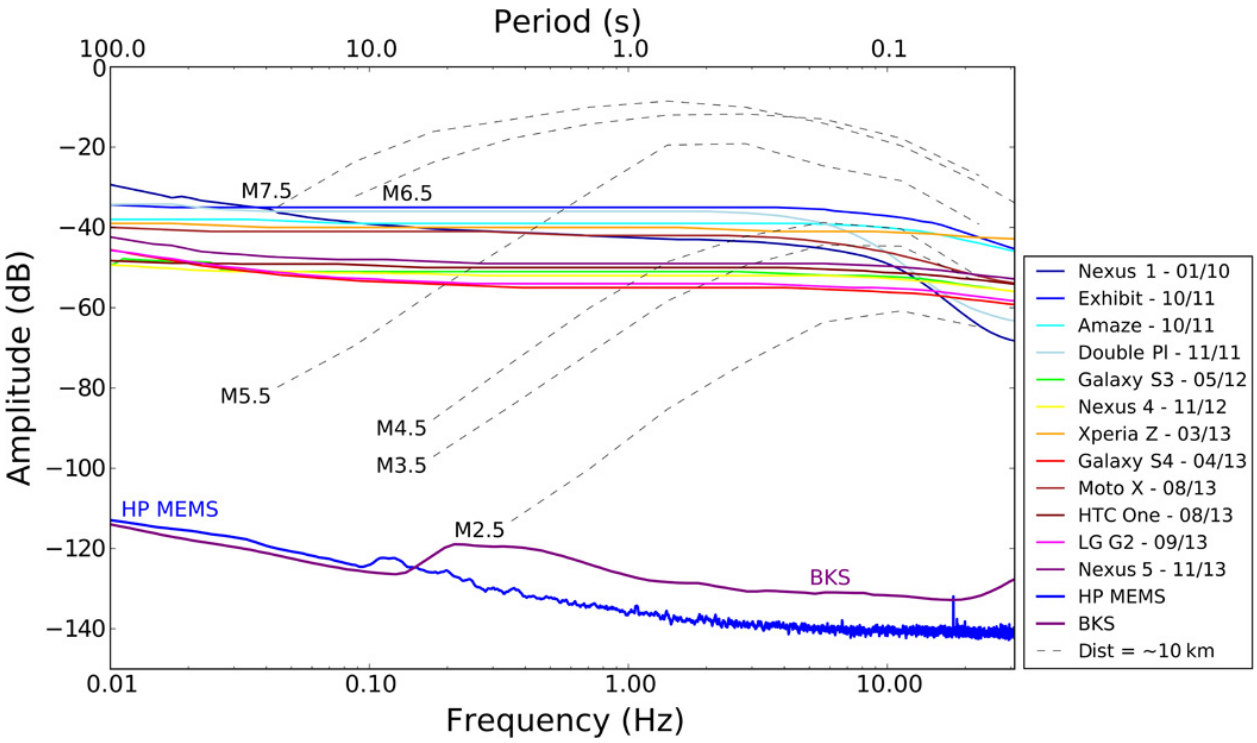


Figure B.1: PSD of Different Magnitudes Earthquakes and seismic grade accelerometers, as reported in [10].

B.3 Noise Floor of MEMS Accelerometers

The future of accelerometers that can be mounted in hand-held devices is very bright. A recently developed MEMS accelerometer has nose floor that is comparable to seismic-grade accelerometers. However, this type of sensors is not attractive for mobile-phone manufacturers due to the high cost. The plot of the HP MEMS [11] accelerometer noise floor is presented in Figure B.2.

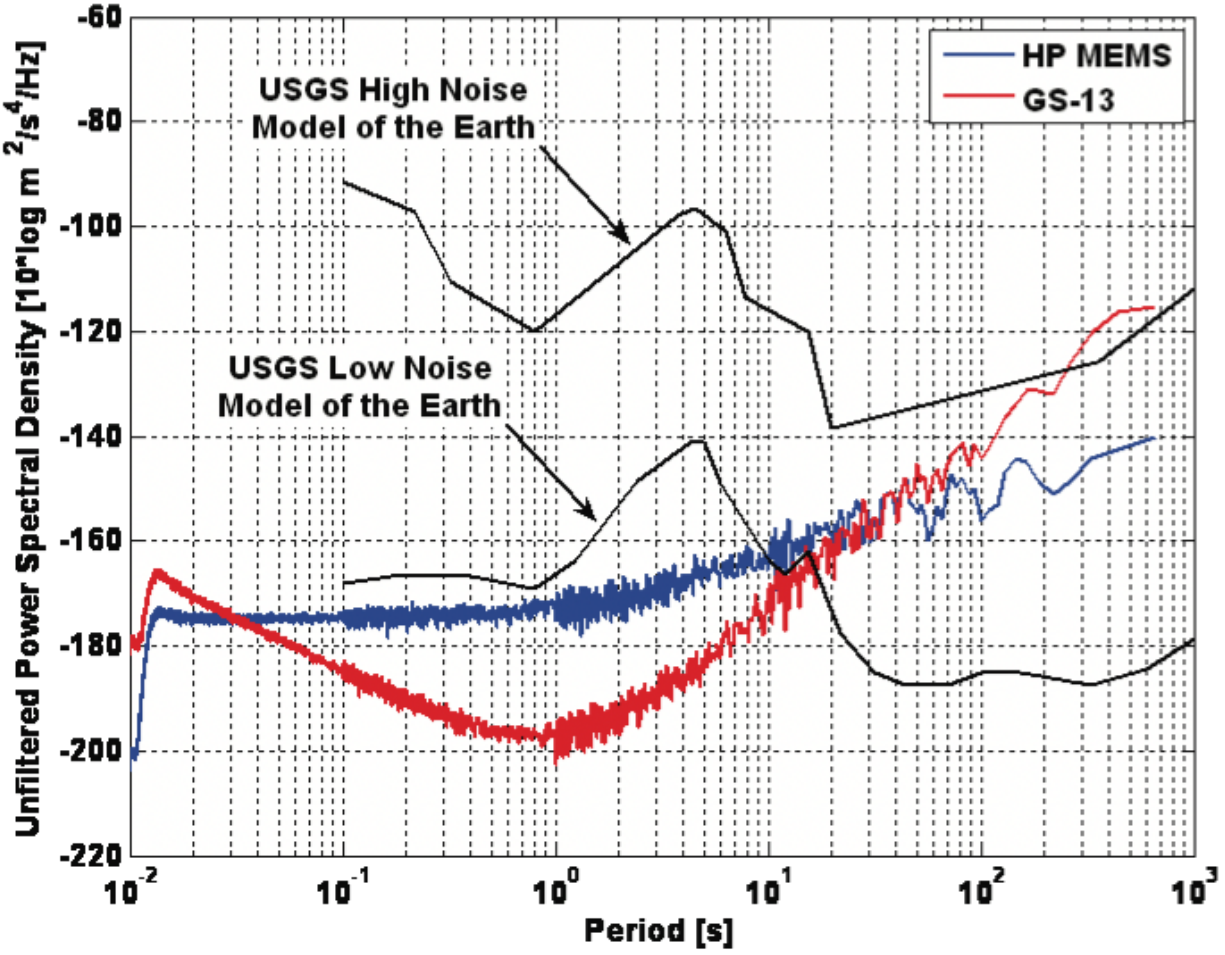


Figure B.2: Noise Floor for HP MEMS Accelerometers, as reported in [11].

Bibliography

[1] Kenneth W Hudnut and Jeffrey A Behr. Continuous gps monitoring of structural deformation at pacoima dam, california. *Seismological Research Letters*, 69(4):299–308, 1998.

[2] Hirotada Hasegawa, Fumio Yamazaki, Masashi Matsuoka, and Izumi Sekimoto. Determination of building damage due to earthquakes using aerial television images. In *Proceedings of the 12th world conference on earthquake engineering*, 2000.

[3] AS Kiremidjian, TW Kenny, KH Law, and T Lee. A wireless modular health monitoring system for civil structures. *Proposal to the National Science Foundation, NSF*, 121842, 2001.

[4] Masashi Matsuoka and Fumio Yamazaki. Use of satellite sar intensity imagery for detecting building areas damaged due to earthquakes. *Earthquake Spectra*, 20(3):975–994, 2004.

[5] SF Masri, LH Sheng, JP Caffrey, RL Nigbor, M Wahbeh, and AM Abdel-Ghaffar. Application of a web-enabled real-time structural health monitoring system for civil infrastructure systems. *Smart Materials and Structures*, 13(6):1269, 2004.

[6] Jong-Han Lee, Hoai-Nam Ho, Masanobu Shinozuka, and Jong-Jae Lee. An advanced vision-based system for real-time displacement measurement of high-rise buildings. *Smart Materials and Structures*, 21(12):125019, 2012.

[7] Mardiyono Mardiyono, Reni Suryanita, and Azlan Adnan. Intelligent monitoring system on prediction of building damage index using neural-network. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 10(1):155–164, 2012.

[8] Robert W Clayton, Thomas Heaton, Mani Chandy, Andreas Krause, Monica Kohler, Julian Bunn, Richard Guy, Michael Olson, Mathew Faulkner, MingHei Cheng, et al. Community seismic network. *Annals of Geophysics*, 54(6), 2012.

[9] Erik H Vanmarcke and Shih-Sheng P Lai. Strong-motion duration and rms amplitude of earthquake records. *Bulletin of the seismological Society of America*, 70(4):1293–1307, 1980.

[10] Qingkai Kong, Richard M Allen, Louis Schreier, and Young-Woo Kwon. Myshake: A smartphone seismic network for earthquake early warning and beyond. *Science advances*, 2(2):e1501055, 2016.

[11] Brian D Homeijer, Donald J Milligan, and Charles R Hutt. A brief test of the hewlett-packard mems seismic accelerometer. Technical report, US Geological Survey, 2014.

[12] Building Seismic Safety Council. Nehrp recommended seismic design criteria for new steel moment-frame buildings. *FEMA-273, Federal Emergency Management Agency, Washington, DC*, 2000.

[13] Earthquake statistics - usgs earthquake hazards program.

[14] Thomas L Holzer and James C Savage. Global earthquake fatalities and population. *Earthquake Spectra*, 29(1):155–175, 2013.

[15] Richard M Allen and Hiroo Kanamori. The potential for earthquake early warning in southern california. *Science*, 300(5620):786–789, 2003.

[16] Thomas H Heaton. A model for a seismic computerized alert network. *Science*, 228(4702):987–990, 1985.

[17] JP Lynch, Y Wang, KC Lu, TC Hou, and CH Loh. Post-seismic damage assessment of steel structures instrumented with self-interrogating wireless sensors. In *Proceedings of the 8th National Conference on Earthquake Engineering*, pages 18–22, 2006.

[18] Matthew Faulkner, Robert Clayton, Thomas Heaton, K Mani Chandy, Monica Kohler, Julian Bunn, Richard Guy, Annie Liu, Michael Olson, MingHei Cheng, et al. Community sense and response systems: Your phone as quake detector. *Communications of the ACM*, 57(7):66–75, 2014.

[19] ASCE/SEI Seismic Rehabilitation Standards Committee et al. Seismic rehabilitation of existing buildings (asce/sei 41-06). *American Society of Civil Engineers, Reston, VA*, 2007.

[20] Matthew Faulkner, Michael Olson, Rishi Chandy, Jonathan Krause, K Mani Chandy, and Andreas Krause. The next big one: Detecting earthquakes and other rare events from community-based sensors. In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pages 13–24. IEEE, 2011.

[21] A Massari, M Kohler, R Clayton, R Guy, T Heaton, J Bunn, KM Chandy, and D Demetri. Dense building instrumentation application for city-wide structural health monitoring. 2017.

[22] Jack Reilly, Shideh Dashti, Mari Ervasti, Jonathan D Bray, Steven D Glaser, and Alexandre M Bayen. Mobile phones as seismologic sensors: Automating data extraction for the ishake system. *IEEE Transactions on Automation Science and Engineering*, 10(2):242–251, 2013.

[23] Getting raw accelerometer events — apple developer documentation.

[24] Network time protocol website.

[25] David L Mills. Internet time synchronization: the network time protocol. *IEEE Transactions on communications*, 39(10):1482–1493, 1991.

[26] Simon Ostermann, Alexandria Iosup, Nezih Yigitbasi, Radu Prodan, Thomas Fahringer, and Dick Epema. A performance analysis of ec2 cloud computing services for scientific computing. In *International Conference on Cloud Computing*, pages 115–131. Springer, 2009.