Computers, Environment and Urban Systems 66 (2017) 23–37

Contents lists available at ScienceDirect

Computers, Environment and Urban Systems

journal homepage: www.elsevier.com/locate/ceus

A cloud-enabled automatic disaster analysis system of multi-sourced data streams: An example synthesizing social media, remote sensing and Wikipedia data

Qunying Huang [a,]*, Guido Cervone [b], Guiming Zhang [a]

[a] Department of Geography, University of Wisconsin-Madison, Madison, WI, 53706, United States
[b] Department of Geography and Institute for CyberScience, Pennsylvania State University, University Park, PA 16802, United States

ARTICLE INFO

Article history:
Received 3 September 2015
Received in revised form 22 February 2017
Accepted 22 June 2017
Available online xxxx

Keywords:
Disaster coordination and relief
Disaster management

ABSTRACT

Social media streams and remote sensing data have emerged as new sources for tracking disaster events, and assessing their damages. Previous studies focus on a case-by-case approach, where a specific event was first chosen and filtering criteria (e.g., keywords, spatiotemporal information) are manually designed and used to retrieve relevant data for disaster analysis. This paper presents a framework that synthesizes multi-sourced data (e.g., social media, remote sensing, Wikipedia, and Web), spatial data mining and text mining technologies to build an architecturally resilient and elastic solution to support disaster analysis of historical and future events. Within the proposed framework, Wikipedia is used as a primary source of different historical disaster events, which are extracted to build an event database. Such a database characterizes the salient spatiotemporal patterns and characteristics of each type of disaster. Additionally, it can provide basic semantics, such as event name (e.g., Hurricane Sandy) and type (e.g., flooding) and spatiotemporal scopes, which are then tuned by the proposed procedures to extract additional information (e.g., hashtags for searching tweets), to query and retrieve relevant social media and remote sensing data for a specific disaster. Besides historical event analysis and pattern mining, the cloud-based framework can also support real-time event tracking and monitoring by providing on-demand and elastic computing power and storage capabilities. A prototype is implemented and tested with data relative to the 2011 Hurricane Sandy and the 2013 Colorado flooding.

1. Introduction

Every year extreme weather and climate events, such as cyclones, floods, tornados and geological events such as volcanic eruptions, earthquakes or landslides, claim thousands of lives and cause billions of dollars of damage to property and severely impact the environment (Velev & Zlateva, 2012). Disasters and their effects have been increasing both in frequency and severity in the 21st century because of climate change, increasing population and their reliance on aging infrastructure. In fact, the first decade of the 21st century witnessed 3496 natural disasters including floods, storms, droughts and heat waves, nearly five times as many disasters as the 743 catastrophes reported during the 1970s.[1] Therefore, an urgent need exists to understand spatiotemporal patterns and the general dynamics that contribute to the occurrences of disasters. These combined studies are necessary to develop effective strategies to mitigate their destructive effects, and to respond and coordinate efficiently to protect people, properties and the environment.

Social media have been primarily used as an intelligent “geo-sensor” network to detect extreme events and disasters such as hurricanes and earthquakes, and to gain situational awareness for emergency responders and relief coordinators during crises by monitoring and tracking citizens feedbacks (Sutton, Palen, & Shklovski, 2008). Additionally, they are widely used by scientists to study public risk perception, and people's reactions during disasters (Mandel et al., 2012). On the other hand, remote sensing data are paramount during disasters and have become the de-facto standard for providing high resolution imagery for damage assessment and the coordination of disaster relief operations (Cervone et al., 2016; Cutter, 2003; Joyce, Belliss, Samsonov, McNeill, & Glassey, 2009). Using high resolution imagery from commercial and research air- and spaceborne instruments, it is possible to obtain data within hours of major events, frequently including ‘before’ and ‘after’ scenes of the affected areas (Cervone & Manca, 2011). These ‘before’ and ‘after’ images are quickly disseminated through scientific portal and news channels to assess damage and inform the public. In addition, first responders rely

* Corresponding author.
E-mail addresses: qhuang46@wisc.edu (Q. Huang), cervone@psu.edu (G. Cervone), gzhang45@wisc.edu (G. Zhang).

[1] http://www.theguardian.com/environment/blog/2014/jul/14/8-charts-climate-change-world-more-dangerous

http://dx.doi.org/10.1016/j.compenvurbsys.2017.06.004
0198-9715/© 2017 Elsevier Ltd. All rights reserved.

heavily on remotely sensed imagery for coordination of relief and response efforts as well as the prioritizing of resource allocation.

Despite the wide availability of large remote sensing datasets from numerous sensors, specific data might not be collected in the time and space most urgently required. Geo-temporal gaps result due to satellite revisit time limitations, atmospheric opacity, or other obstructions. Recently, stream data from social media and remote sensing are being fused for disaster analysis and assessment. Specifically, social media are used to fill in the gaps when remote sensing data are lacking or incomplete (Schnebele & Cervone, 2013; Schnebele, Cervone, Kumar, & Waters, 2014; Schnebele, Oxendine, Cervone, Ferreira, & Waters, 2015).

However, current studies on using social media and remote sensing data for disaster analysis are performed on a case-by-case basis. The approaches typically start with identifying a specific disaster event, and then filters (e.g., keywords, spatiotemporal information) are designed to select and retrieve relevant stream data. These efforts are time-consuming. For example, identifying the tweet hashtags associated to a specific event, may take from hours to days for manual examination of hundreds of tweets to include relevant hashtags so we can use them to filter out non-relevant tweets during a disaster. Furthermore, these efforts need to be duplicated when analyzing a different event. As stated earlier, it is necessary to complete a comprehensive database that can display the historical events with relevant metadata (e.g., event type, severe category, damages, locations, and temporal spans) to allocate resources for analysis. Additionally, from the basic metadata, it is also needed to automatically derive relevant information (e.g., hashtags), which can then be used to retrieve relevant messages from long-term accumulated social media.

With multi-sourced data streams from a multitude of channels, identifying authoritative sources and extracting critical, validated messages information can be quite challenging, especially during a crisis. The volume, velocity, and variety of accumulated stream data produce the most compelling demands for computing technologies from big data management to technology infrastructure (Huang & Xu, 2014). To address these big data challenges, various types of computational infrastructures are designed, from the traditional cluster and grid computing to the recent development of cloud computing and CPU/GPU heterogeneous computing (Schadt, Linderman, Sorenson, Lee, & Nolan, 2010). Specifically, cloud computing has been increasingly viewed as a viable solution to utilize multiple low-profile computing resources to parallelize the analysis of massive data into smaller processes (Huang & Cervone, 2016).

This paper addresses these problems by proposing a novel system to support both historic disaster event analysis and upcoming event monitoring. Wikipedia is exploited as a source to build a disaster event database, which is then applied to retrieve relevant information for a specific disaster from massive social media data accumulated daily. Cloud computing is proposed to serve as the underlying infrastructure that offers the capability of providing on-demand and flexible computing resources to meet the dynamic computing requirements of real-time disaster analysis. The following contributions are made in this research:

1. An integrated system framework is proposed for historical disaster analysis based on multi-sourced data with limit, if any human interaction. To analyze and understand the public behaviors or reactions captured by social media data, our system does not rely on human identification of filtering criteria to retrieve relevant messages. An automatic system based on text mining, and geocoding technologies are developed to derive these information.
2. An event database is built based on Wikipedia. Such a database is useful for scientists easily selecting a relevant event for analysis or selecting disasters of a specific type to identify their patterns, and linking it to other GIS data (e.g., socioeconomic data), climate data, and environment data to understand the driving factors that contribute to the occurrences of these disaster events.
3. Within the proposed system, cloud computing is used as the underlying infrastructure to provide flexible computing power to address the computing challenges posed by the massive data processing and a real-time operational system for emergencies response and disaster coordination. Such a system is suitable for online services and systems where a number of texts, and remote sensing images are dynamically streaming.
4. A prototype is implemented, and recent flooding events are used as a case study to demonstrate the feasibility of the proposed system.
5. This paper provides a general methodology that it is not event specific, and can be used both for retrospective analysis and for real time monitoring and decision making. The proposed framework sheds light on integrating various emerging data sources to support scientific applications of significant interests that go beyond disaster management.

2. Related work

2.1. Social media for disaster management

As social media applications are widely deployed in various platforms from personal computers to mobile devices, they are becoming a natural extension to human sensory system. The synthesis of social media with human intelligence has the potential to be the intelligent sensor network that can be used to detect, monitor and gain situational awareness during a hazard with unprecedented scale and capacity. By monitoring tweets, for example, an earthquake can be detected by developing a probabilistic spatiotemporal model for the target event that can find the center and the trajectory of the event location (Sakaki, Okazaki, & Matsuo, 2010).

By mining social media data, it is possible to establish situation awareness for disaster response and relief (Ashktorab, Brown, Nandi, & Culotta, 2014; Gao, Barbier, & Goolsby, 2011; Huang & Xiao, 2015; Imran, Elbassuoni, Castillo, Diaz, & Meier, 2013; Kumar, Barbier, Abbasi, & Liu, 2011). Using Hurricane Sandy as an example, Huang and Xiao (2015) coded social media messages into different themes within different disaster phases during a time-critical crisis, and a classifier based on logistic regression is trained and used for classifying the social media messages into various topic categories during various disaster phases. Imran et al. (2013) extracted information from disaster-related messages posted on Twitter into several categories including warnings, casualties and damage, donations, and information sources. The coded information can be further analyzed over space and time to inform the situational awareness of the incidences as they unfold. The Australian Government developed an Automated Web Text Mining (ESA-AWTM) system that analyzes Twitter messages to provide incidence identification, near real-time notification, and monitoring (Cameron, Power, Robinson, & Yin, 2012). A web application, “TweetTracker”, was developed by Kumar et al. (2011) to track, analyze, and monitor tweets for disaster relief. It can report separately geo-referenced and non-geo-referenced tweets, support keyword search, and generate and display trends of keywords specified by the user.

However, all published methods rely on a case-by-case analysis of historical events, or support simple real-time data searching and analysis functions (Kumar et al., 2011). There is no systematic approach proposed to support both historical event and real-time event tracking capabilities.

2.2. Synthesizing multi-sourced data for disaster management

In a time of disaster, multi-sourced data can be integrated to assess the situation. Such integration results in new approaches to support disaster management in a new way that cannot be done previously and significantly improve the analysis and capability of a single data source. For example, while social media data have been successfully used to detect and track locations of disaster events (e.g., earthquake, tornado, and wildfire; Sakaki et al., 2010; De Longueville, Smith, & Luraschi, 2009; Jain, 2015), disaster detection is not always possible using single-sourced data (e.g., tweets) alone and there is a need to integrate additional space- and time-referenced information collected from other sources (Fuchs, Andrienko, Andrienko, Bothe, & Stange, 2013). As a result, an active research is to fuse data from multiple sources to support and improve disaster management. De Albuquerque et al. (2015) used "authoritative" data (e.g., sensor data, hydrological data and digital elevation models) to enhance the identification of disaster relevant social media messages. Schnebele and Cervone (2013) integrated remote sensing data with social media to verify the presence of water in a specific area during flooding events.

2.3. Data challenge and cloud computing

Social media data are real-time in nature. These stream data, along with other official and authoritative data sources, such as remote sensing, are valuable for disaster management yet require a computing environment that is adaptable to expand and store peaks of considerable amounts of data in timely manner (Wang, 2010). Traditional computing platform lacks the scalable computing infrastructures to handle streaming social media data. Cloud computing, a new distributed computing paradigm, has been widely utilized to address Geoscience challenges of computing, data and concurrent intensities (Yang, Wu, Huang, Li, & Li, 2011). One of the most important characteristics of cloud computing infrastructure is that users can provision computing resources to run computing tasks with automated workflows for maximum efficiency and scalability in minutes. There is no need to wait in queues and compete for limited computing nodes as in the traditional computing paradigm. Therefore, cloud computing is a powerful and affordable alternative to run large-scale data processing and computation that are computationally intensive (Huang & Cervone, 2016; Huang, Yang, Benedict, et al., 2013; Huang, Yang, Liu, et al., 2013; Tang & Feng, 2017; Tang et al., 2017; Yang et al., 2011).

Many studies have been conducted to explore the feasibility of utilizing cloud computing for geospatial applications and how to best adapt to this new paradigm (Evangelinos & Hill, 2008; Huang, Yang, Benedict, et al., 2013; Li et al., 2017; Yang et al., 2011). Evangelinos and Hill (2008) concluded that cloud computing could provide a potential solution to support atmosphere-ocean climate models, and Huang, Yang, Benedict, et al. (2013) utilized cloud computing to simulate dust storms. However, these studies mostly use cloud computing to support the computing- intensive geospatial science simulations, and no effort has been made to explore how cloud computing may be leveraged in disaster management. Recently, Hadoop cloud platform is a widely used scalable distributed computing environment to process social media data (Gao, Li, Li, Janowicz, & Zhang, 2017). However, only limited geospatial applications have been developed to leverage such MapReduce based framework, which has come to define big data process.

While social media is widely used to support disaster management, the variety and veracity of social media data poses grand challenge to the current streaming data processing frameworks and architecture. Zelenkauskaite and Simões (2014) implemented a mobile application based on cloud architecture to support computationally intensive operations, such as searching, data mining, and data processing at large scale. Padmanabhan et al. (2014) introduced a data-driven framework using geographic information systems (GIS) based on advanced cyberinfrastructure (CyberGIS) and massive social media data to analyze spatiotemporal events across spatial and temporal scales. Similarly, Huang, Cervone, Jing, and Chang (2015) presented a CyberGIS framework to synthesize multi-sourced data (e.g., social media, socioeconomic data) for tracking disaster events, producing maps, and performing various analyses for disaster management. The proposed framework is capable of supporting Big Data analytics from multiple sources. Additionally, multimedia streaming data (e.g., social media, remote sensing) are difficult to analyze and process in real time because these streams are diverse, complex and overwhelming in volume, velocity and in the variety of the data fields. Correspondingly, Zhang, Chen, and Yang (2015) established a Markov chain model to predict the trend of big stream data and then appropriate cloud computing nodes are allocated to process them using the predicted results.

3. A cloud-based disaster analysis system

Fig. 1 shows a general architecture for disaster analysis leveraging multiple sources. The system is designed to include six integrated components, including: 1) Data repository, responsible for archiving and retrieving datasets. An automatic system is developed to crawl and integrate unstructured, heterogonous data from various sources, such as Wikipedia, remote sensing, social media, and Web. Disaster relevant messages posted from different social medias such as Twitter, and Flickr are monitored and tracked; 2) Data server. This component provides basic data processing functions; 3) Application server, offering high-level analytical functions; 4) Web server, providing the key functions to support data search, analysis, visualization, or animation service requested from through the web portal. 5) Spatial web portal (SWP), providing information analysis for end users through geovisualization or animation with interactive tools (Roth, 2012, 2013). The SWP is a web-based or mobile-based spatial gateway for the purposes of tracking, visualizing and analyzing disaster events. With the portal, public users can search against the resource catalog to discover the historical events and track a latest disaster event; 6) Cloud clusters. Cloud clusters provide a set of core functions to implement an architecturally resilient and flexible disaster analysis system. The following sections elaborate several important components in our system design.

3.1. Data server

The data server implements a variety of functions that are performed on the raw data. The basic function that the server provides is harvesting data using different application programming interfaces (APIs) to ingest data from multiple sources. In this work, Wikipedia (https://www.wikipedia.org/) is used as a primary data source for building an event database. Wikipedia includes a wealth of information for researchers in easy to access formats including XML, SQL and HTML, which makes it an attractive repository for projects on information extraction. It is worth noting that there are many other disaster information sources could potentially be used as sources to build such an event database, such as emdat.[2] Wikipedia is ideal for the proposed task because its raw data are easily accessible using APIs and they can be imported in a local database for fast access and analysis in a high performance framework. Twitter Stream APIs can be used to harvest about 1% total tweets. Each tweet is a document entry with metadata about that tweet message, such as the user name, time stamp and location when the tweet was created, source generating the tweet, text content, and hashtags, etc. Using a data-driven approach to archive social media data (Huang & Xu, 2014), millions of tweet messages are accumulated daily.

The data server is also used to store the remote sensing data related to the events being investigated. The main challenges arise from the sheer volume of remote sensing data, where each granule (a multi-spectral scene) can be up to a terabyte in size. The data are stored with ancillary data which specify the platform, the spatial and temporal coverage, and additional metadata which include any pre-processing applied to the data from the time of acquisition to the time it is distributed.

[2] http://www.emdat.be/database; a global database on natural and technological disasters that contains essential core data on the occurrence and effects of more than 21,000 disasters in the world from 1900 to present

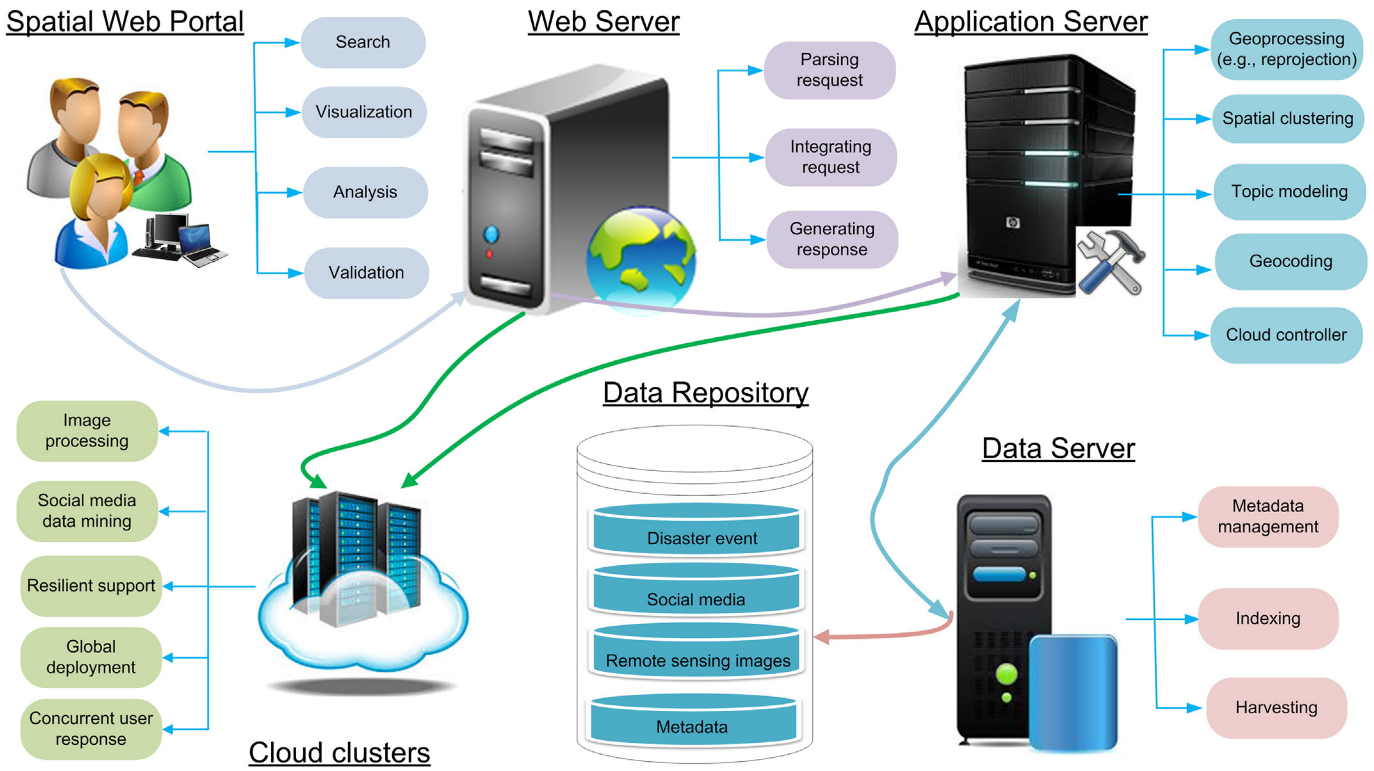


Fig. 1. Architecture of a cloud-based disaster analysis system.

Depending on the data source, the ingestion of remote sensing data into the data server can be automatic using APIs, or can occur manually. For example, NASA/USGS Landsat and NASA MODIS data can be automatically ingested using an APIs from NASA and USGS. However, commercial high resolution data must be processed manually because of licensing limitations. Most remote sensing data related to worldwide hazards can be downloaded using the USGS Hazards Data Distribution System (HDDS), which includes data both for aerial and space platforms. A Python script was created to automatically download relevant free data (e.g. USGS Landsat, Civil Air Patrol, MODIS), which can be automatically ingested into the data server.

Several data pre-processing procedures are then developed to extract relevant information from these raw data before we can import and store them to the data repository. The overall methodology is illustrated using Wikipedia data as example (Fig. 2). Wikipedia data processing requires using JWPL (Java Wikipedia Library; https://code.google.com/p/jwpl/wiki/DataMachine), a free Java-based API that allows accessing all information in Wikipedia. For each type of disaster (e.g., Category 3 Atlantic hurricanes; https://en.wikipedia.org/wiki/Category:Category_3_Atlantic_hurricanes), a manual search is performed for a category name defined in Wikipedia and use this category as the root node. Then breadth-first search (BFS) is performed from this node to identify all subcategories. The leaf nodes resulting from such transversal corresponded to Wikipedia pages or articles describing specific events.

Each article contains one or more “infobox(es)”, which include structured metadata that are added to the top right-hand corner of articles (Fig. 3). The JWPL tool allows retrieving all infobox tables (or templates) which contain important facts and statistics of a type (e.g., disaster impact information) which are common to related articles. Therefore, for each article in Wikipedia, the infobox templates are processed and used to extract metadata for a specific event. The resulting text then can be further processed with the same procedure starting

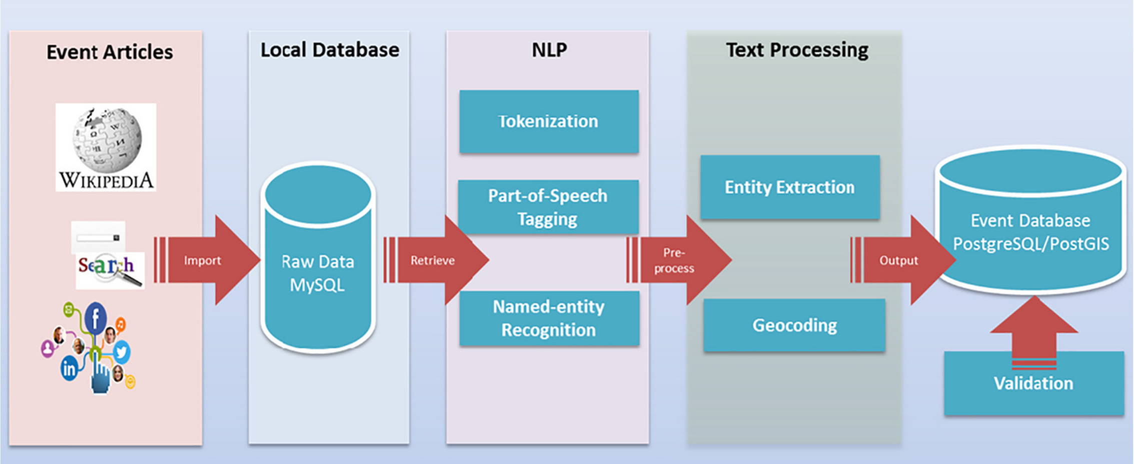
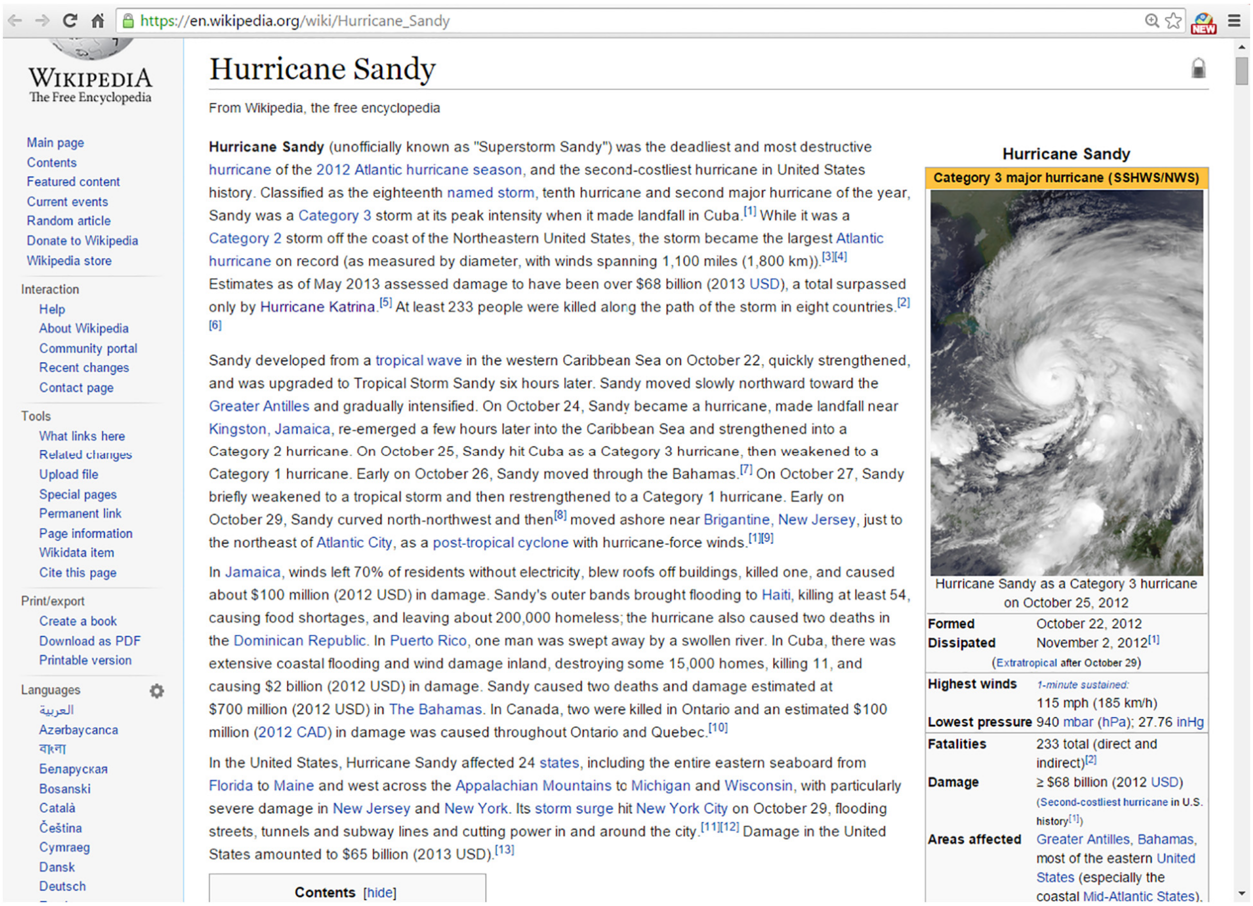


Fig. 2. Workflow to build a disaster event database.



Hurricane Sandy

From Wikipedia, the free encyclopedia

Hurricane Sandy (unofficially known as "Superstorm Sandy") was the deadliest and most destructive hurricane of the 2012 Atlantic hurricane season, and the second-costliest hurricane in United States history. Classified as the eighteenth named storm, tenth hurricane and second major hurricane of the year, Sandy was a Category 3 storm at its peak intensity when it made landfall in Cuba.[1] While it was a Category 2 storm off the coast of the Northeastern United States, the storm became the largest Atlantic hurricane on record (as measured by diameter, with winds spanning 1,100 miles (1,800 km)).[3][4] Estimates as of May 2013 assessed damage to have been over $68 billion (2013 USD), a total surpassed only by Hurricane Katrina.[5] At least 233 people were killed along the path of the storm in eight countries.[2][6]

Sandy developed from a tropical wave in the western Caribbean Sea on October 22, quickly strengthened, and was upgraded to Tropical Storm Sandy six hours later. Sandy moved slowly northward toward the Greater Antilles and gradually intensified. On October 24, Sandy became a hurricane, made landfall near Kingston, Jamaica, re-emerged a few hours later into the Caribbean Sea and strengthened into a Category 2 hurricane. On October 25, Sandy hit Cuba as a Category 3 hurricane, then weakened to a Category 1 hurricane. Early on October 26, Sandy moved through the Bahamas.[7] On October 27, Sandy briefly weakened to a tropical storm and then restrengthened to a Category 1 hurricane. Early on October 29, Sandy curved north-northwest and then[8] moved ashore near Brigantine, New Jersey, just to the northeast of Atlantic City, as a post-tropical cyclone with hurricane-force winds.[1][9]

In Jamaica, winds left 70% of residents without electricity, blew roofs off buildings, killed one, and caused about $100 million (2012 USD) in damage. Sandy's outer bands brought flooding to Haiti, killing at least 54, causing food shortages, and leaving about 200,000 homeless; the hurricane also caused two deaths in the Dominican Republic. In Puerto Rico, one man was swept away by a swollen river. In Cuba, there was extensive coastal flooding and wind damage inland, destroying some 15,000 homes, killing 11, and causing $2 billion (2012 USD) in damage. Sandy caused two deaths and damage estimated at $700 million (2012 USD) in The Bahamas. In Canada, two were killed in Ontario and an estimated $100 million (2012 CAD) in damage was caused throughout Ontario and Quebec.[10]

In the United States, Hurricane Sandy affected 24 states, including the entire eastern seaboard from Florida to Maine and west across the Appalachian Mountains to Michigan and Wisconsin, with particularly severe damage in New Jersey and New York. Its storm surge hit New York City on October 29, flooding streets, tunnels and subway lines and cutting power in and around the city.[11][12] Damage in the United States amounted to $65 billion (2013 USD).[13]

Hurricane Sandy

Category 3 major hurricane (SSHWS/NWS)

Hurricane Sandy as a Category 3 hurricane on October 25, 2012

Formed	October 22, 2012
Dissipated	November 2, 2012[1]
(Extratropical after October 29)	
Highest winds	*1-minute sustained:* 115 mph (185 km/h)
Lowest pressure	940 mbar (hPa); 27.76 inHg
Fatalities	233 total (direct and indirect)[2]
Damage	≥ $68 billion (2012 USD) (Second-costliest hurricane in U.S. history[1])
Areas affected	Greater Antilles, Bahamas, most of the eastern United States (especially the coastal Mid-Atlantic States),

Fig. 3. An example of Wikipedia article introducing Hurricane Sandy (The top right-hand corner of article shows information from an infobox).

from natural language processing (NLP; Fig. 2). NLP tasks include from relatively straightforward tokenization and part-of-speech (POS) tagging, to more complex procedure named-entity recognition (NER), which identifies and categorizes atomic elements in text (e.g., Organizations, Persons, Locations, Time, etc.). All atomic elements are mined using an ad-hoc procedure to extract the values of the fields defined in the event database. If the element is a location represented with an address, a geocoding process using the Google Geocoding API (https://developers.google.com/maps/documentation/geocoding/intro) is invoked to extract the corresponding geographic coordinates.

3.2. Data repository

After the multi-sourced data is collected, they are imported into different database types to be processed for disaster analysis. Social media data are not uniform and structured in nature, and therefore they are stored in a non-traditional database (DB) system. In the current implementation, the MongoDB (Huang & Xu, 2014) is used. This is a scalable open source NoSQL database which is designed to manage those social media datasets efficiently. The remote sensing data are stored in a PostgreSQL database using spatial extensions (PostgreSQL/PostGIS). Specifically, the spectral layers are stored as binary fields, and are associated with metadata which define the spatial extents, the temporal coverage, and other fields that describe when, where and how the data was acquired and processed. The use of a spatial database allows performing SQL queries to quickly identify associated images that match the disaster event being analyzed.

Disaster events extracted from the Wikipedia are annotated with fields that give structured information, and consequently can be efficiently managed and organized using a traditional spatial relational database management system (RDMS), such as PostgreSQL/PostGIS. For each disaster event entry, the database stores several key information categories, such as physical infrastructure damages, economic and life losses, locations impacted by the event, and temporal duration of the event. These information can be used to quantify the impact of disaster in terms of economic loss from flooding damage, and total number of injuries and deaths. Additionally, they provide a basic metadata that can be used for developing filtering criteria for stream data in the database. To efficiently retrieve social media from the NoSQL database, metadata about social media are stored in the PostgreSQL/PostGIS database. For example, the hashtags associated with each event are stored in our metadata database and can be efficiently queried.

3.3. Application server

The application server provides a series of services to address the issues of the data and service integration and interoperability across various data. The application server can perform a variety of analytical or data mining functions requested by other servers. Apache Mahout (http://mahout.apache.org/), an open-source library that implements scalable machine learning algorithms, is used as the underlying library to support big data analytics. Mahout has been widely used to perform various text mining tasks, such as grouping together similar documents by using various clustering algorithms (Huang & Xiao, 2015). It is very fast and has excellent integration with other popular open-source Apache libraries, such as Hadoop and Lucene (a high-performance text search engine library). Recently, MapReduced based systems have emerged as a new computing paradigm for massively parallel data process (Dean & Ghemawat, 2008). Hadoop, the most widely used implementation of MapReduce, has been successfully applied in large-scale Internet services to support big data analytics. In Mahout, many classic algorithms for data mining, such as naïve Bayes, Latent Dirichlet

Allocation (LDA) (Blei, Ng, & Jordan, 2003), and logistic regression are implemented as MapReduce jobs in Mahout.

In our work, two critical data mining tasks are to discover i) the emerging "hot topics" that are discussed over the social media by constantly running a topic modeling algorithm, and ii) locations of these topics by running spatial clustering algorithm (Fig. 1). These functions are critical to detect potential disasters, and are introduced in Section 4 in details. Additionally, the application server can also perform basic GIS data processing functions, such as geospatial data re-projection and format conversion. It is noted that the interaction between the data server and application server are dual. On one hand, the data preprocessing module dispatched in the data server may need to call the functions (e.g., geocoding and basic GIS data processing functions) that are deployed on the application server. On the other hand, application server needs to request the data retrieval service from the data server which can communicate with the data repository directly.

The integrated system for disaster analysis is built to integrate computing resources from both private cloud computing platform based on the open source Eucalyptus cloud solution (Huang, Yang, Benedict, et al., 2013; Huang, Yang, Liu, et al., 2013), and public cloud Amazon EC2 (https://aws.amazon.com/ec2/). To manage cloud computing resources (e.g., virtual machines and virtual storages) from different cloud platforms, a cloud control module is developed and deployed on the application server to interact with different clouds through APIs. This module provides an abstraction interface and handles the implementation details for different underlying cloud platforms. Whenever there is a data processing task (e.g., hot topic detection) coming in, this module will invoke different types of cloud cluster to execute it.

3.4. Cloud clusters

Specifically, the cloud clusters can support the underlying computing requirements posed by an operational system based on the proposed framework by providing the following functions (Fig. 1).

- Remote sensing image processing: Remote images are used to derive damage extent and assessment maps. With the rapid improvement of data acquisition technologies, remote sensing images of before and after a disaster are available at high spatial and temporal resolution. The damage assessment for the 2013 Colorado flooding was performed using 12 satellite and over 6000 aerial images. Parallel computing and distributed systems (Guan, Zeng, Gong, & Yun, 2014; Shook, Wang, & Tang, 2013) are needed to process these large amount of high-resolution imagery in order to derive customized products in real-time. Specially, multiple cloud nodes can be launched with each node processing a portion of aerial images instead of using a single machine to handling all images in serial.
- Social media data mining: Text and spatiotemporal mining of massive social media datasets are time-consuming. The computing cost of the DBSCAN algorithm (Ester, Kriegel, Sander, & Xu, 1996), for example, used for detecting the locations of potential disaster events (Section 3.2) is the n^4 where n represents the number of clustered points; it is computing intensive, especially when the results are expected to produce in a real-time fashion. Therefore, parallel computing and the latest computing models such as cloud computing (Huang, Yang, Benedict, et al., 2013) should be applied into the proposed framework. Detecting the "hot topics" from streaming social media data using topic modeling algorithm is also computational intensive. Our current system implementation uses MapReduced based framework to support such real-time stream data process. However, more efficient big data computing paradigms, such as Apache Storm (Ranjan, 2014), Apache Kafka (Kreps, Narkhede, & Rao, 2011), and Amazon Kinesis,[3] can be leveraged and integrated into our proposed system in future.
- High performance: Cloud computing provides scientists with a complete new computing paradigm for accessing and utilizing computing infrastructure. Cloud computing services, especially Infrastructure as a Service (IaaS), a category of popular cloud services, can be easily adopted to offer the prevalent high-end computing technologies to provide more powerful computing capabilities. Many cloud providers offer a range of diverse computing resources for users' computing needs, such as Many Integrated Cores (MICs), Graphics Processing Units (GPUs; Tang & Jia, 2014), and Field Programmable Gate Arrays (FPGAs). For example, Amazon EC2 Cluster, with 17,024 CPU cores in total, a clock speed of 2.93 GHz per core, and 10G Ethernet network connection, was ranked as 102th on the TOP 500 supercomputer lists in the November 2012. The HPC capability of cloud computing can be easily leveraged to support critical scientific computing demands (Huang, Yang, Benedict, et al., 2013).
- Resilient support: Architectural resilience can be achieved in many ways including 1) having back-up redundant systems that automatically deploy when primary systems fail, or 2) employing multiple solutions to ensure that a minimum level of system functionality is available during massive system failures (Pu & Kitsuregawa, 2013). Cloud services provide an ideal platform to implement this resilient mechanism. Cloud computing providers offer computing and storage services that are globally distributed. For example, Amazon EC2 has multiple data centers around the world with the service. An image containing the configured system could be built in cloud services, and then a new replicated application can be easily launched on failover systems in a different cloud zone in a few minutes (Huang, Yang, Benedict, et al., 2013) after a failure.
- Concurrent user response: Hazard events often have annual or seasonal variability and are of short duration. Most events typically last a relatively short period from several hours (e.g. tornados) to several days (e.g. hurricanes). As a result, a real-time response system for such events would experience different computing and access requirements during different times of the year and even different hours within the day. During a disaster, the computing platform supporting an emergency response system should be able to automatically scale up enough computing resources to produce and deliver relevant and useful information for the end users. After the emergency response, the access to information can be reduced and the system can switch back to "normal mode" for reduced costs. Computing resources would be released for other science, application, and education purposes. Applications, running on the cloud, can increase computing resources to handle spike workloads and accelerate geocomputation in a few seconds to minutes. Additional computing resources can be released in seconds once the workloads decrease.

Within the proposed framework, a cloud cluster generally comprises a virtual head node and multiple virtual computing nodes. Both head and computing nodes are created as virtual machines that can run independently and communicate through a virtual network. Middleware is installed and configured on all nodes to monitor and support communication between the head node and computing nodes. The head node is responsible for (1) scheduling and dispatching tasks to computing nodes, (2) activating the computing tasks by configuring the middleware, and (3) collecting results from the computing nodes. Different types of cloud cluster are configured by deploying different open-source middleware solutions, such as Apache Hadoop (https://hadoop.apache.org/), Condor (http://research.cs.wisc.edu/htcondor/) and MPICH (https://www.mpich.org/), and used to create images, which in turn are readily launched to run different types computing tasks. For instance, a Hadoop cluster can be configured to process and analyze the large amount of social media data because of its key characteristics of being reliable, flexible, economical, and a scalable solution.

[3] https://aws.amazon.com/kinesis.

4. Real-time event detection and tracking

4.1. Hashtag detection: Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA; Blei et al., 2003) is an example of a topic model for analyzing a large number of unlabeled data. LDA can be used to cluster words into "topics" and documents into mixtures of "topics" by uncovering the hidden thematic structure (or "topics") in a large collection of documents. In LDA, each document is represented as a probability distribution of various topics, which are in turn distributions over words. Each word could belong to one or more topics.

As each tweet is limited to 140 characters, it is highly unstructured, including a large number of abbreviations, and hashtags, unspaced phrases prefixed with the sign "#". Any user who wants to create a concept category and to discuss and share specific information about a subject can create a hashtag. A hashtag, an identifier unique to Twitter, is often used to search for tweets that have a common topic. Therefore, in our study, each tweet is a document and only hashtags are extracted as words to represent the document while modeling the tweeting topics with LDA.

The Apache Mahout machine learning library is integrated to run the LDA over the recently collected tweets to discover topics that are currently discussed over the social media, and detect potential disaster events. While running the LDA algorithm, we need to enter a critical parameter - the number of topics (k). A low *k* value could produce in broader topics, while a large *k* value results in focused topics. A large value also results in more computation to estimate the word distribution for all topics. In order to choose an appropriate value, a program is developed to calculate the frequency of hashtags in the document, and count the number of hashtags (n) that have the frequency larger than a predefined threshold (e.g., 100). This number (n) is then used to set up the value of *k*.

An advantage using Mahout is that the LDA algorithm is implemented as a MapReduce job, which can be run in a large Hadoop clusters. Therefore, we can leverage cloud clusters (Fig. 1) to speed up the process of topic detection by adding more virtual machines into the computation. After running LDA model, an output of the computed topics with each topic being represented as a set of frequently occurring words (hashtags) with certain probability is produced.

4.2. Event detection: a density-based method

After discovering "hot topics" through LDA model, the words (hashtags) in each topic are used to match a predefined list of keywords related to a natural hazard, such as "hurricane", "storm", "quake" etc. If a match is detected, an alert would be posted by the system to provide early warnings about a potential disaster. At the same time, the geo-tagged tweets with these hashtags are retrieved from the database to detect the locations where the disaster might occur. We can apply spatial clustering to learn the regions that may have potential natural disasters causing unusual tweeting behaviors over the social media. K-means (Ashbrook & Starner, 2003) is a well-known clustering algorithm commonly used to identify Point of Interests (POIs). However, K-means algorithm needs us to specify the number of clusters (K) in advance. This is quite challenging since we do not know appropriate values for the number of places that have potential disaster events.

The density-based spatial clustering algorithm (Ester et al., 1996) is designed to discover clusters of arbitrary shape with noises. It does not require specifying the number of clusters, but requires two inputs: the minimum number of points (minpts) forming a cluster, and the radius of the ε-neighborhood of a point (eps). These two parameters are less likely to be changed within a particular application. We therefore chose DBSCAN to cluster these geo-tagged tweet points into regions of potential interests. After performing the spatial clustering, a set of clusters is discovered with each cluster indicating a region of potential natural disaster area. A boundary (convex hull or concave hull) of each cluster including all the points in the cluster is used to better represent the shape of the region. In fact, simpler features, such circumscribed circles and minimum rectangles may also be used to represent the potential regions.

5. Demonstration

A system prototype is implemented based on JAVA, Java Server Page (JSP) and Python to automatically harvest and analyze various types of data. Several open-sources are used for the prototype development. For instance, Apache Mahout package is used for performing data and text mining tasks, Apache Lucene for text processing and indexing tasks, and Google Maps and Geocoding APIs for mapping and geocoding the tweets. Various geovisual tools are developed and accessible through the developed spatial web portal that allows users to customize analysis and view multi-sourced data in different types of maps and plots for disaster management.

5.1. Event database construction and analysis

The system was tested using data relative to Atlantic hurricanes of category 1 to 5 that have occurred since 18th century. First, event data are harvested and imported to our event database using Wikipedia. An Atlantic hurricane or tropical storm is a tropical cyclone that forms in the Atlantic Ocean, Caribbean Sea and Gulf of Mexico, usually in the Summer or Fall. 365 events in total are extracted. The earliest record is '1812 Louisiana hurricane', which was a major hurricane that struck New Orleans, Louisiana, during the War of 1812. The most recent one is 'Hurricane Bertha (2014)',[4] an unusual tropical cyclone in early August 2014. As a tropical cyclone, Bertha's impact was relatively minor. Widespread power outages occurred along its path but no major damage or loss of life took place.

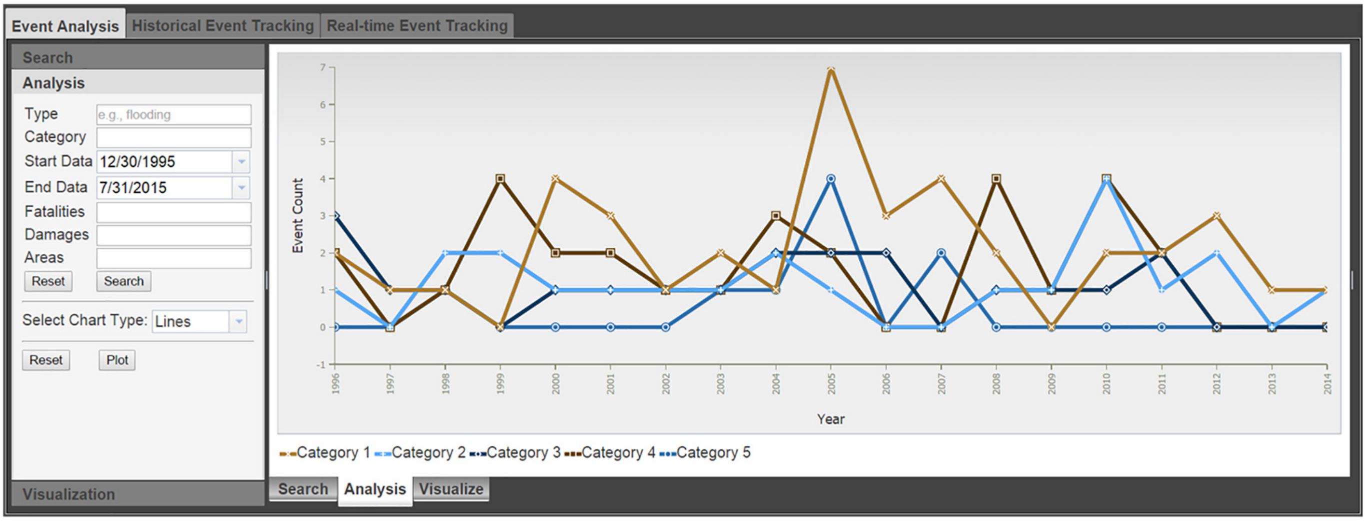
The developed web portal has several functions to support the analysis of the events temporally (Fig. 4) and spatially (Fig. 5). The left panel allows users to analyze a specific event by selecting the available event types and categories stored in the database, and configuring spatial and temporal scopes. The system is able to return the statistical results as statistical plots to the client. It can be observed that we have witnessed a large number of flooding events triggered by a tropical cyclone in 2005 (Fig. 4). Within this year, most of flood events are categorized as 1 (seven events), where damage is mostly to trees and shrubbery with no real building damage, followed by the flooding of category 5 (four events), where extreme damages would be made to both the physical infrastructure and human. One of the five deadliest hurricanes in the history of the U.S, Katrina, occurred exactly in this year. Among all the states along the east coast line, the most vulnerable state is Florida, which was struck by hurricanes for 70 times in the past 200 years, followed by Texas (39 times) and Louisiana (37 times).

Authorized users also allow checking details, editing, approving or disapproving a specific event (Fig. 6). Since the event metadata are extracted automatically through the Wikipedia, and its accuracy are highly reliant on the text mining techniques used in the research. Therefore, authorized users can check the information about an event, such as location, and time spans. If there is any error discovered, they can submit an edit request, which is approved or rejected by users with higher level of access permission of the system.

5.2. Hashtag detection and real-time event tracking

The system is able to support the automatic detection of potential events by integrating topic modeling and spatial clustering algorithms. Fig. 7 (left panel) shows that the end users can set up relevant

[4] https://en.wikipedia.org/wiki/Hurricane_Bertha_(2014).



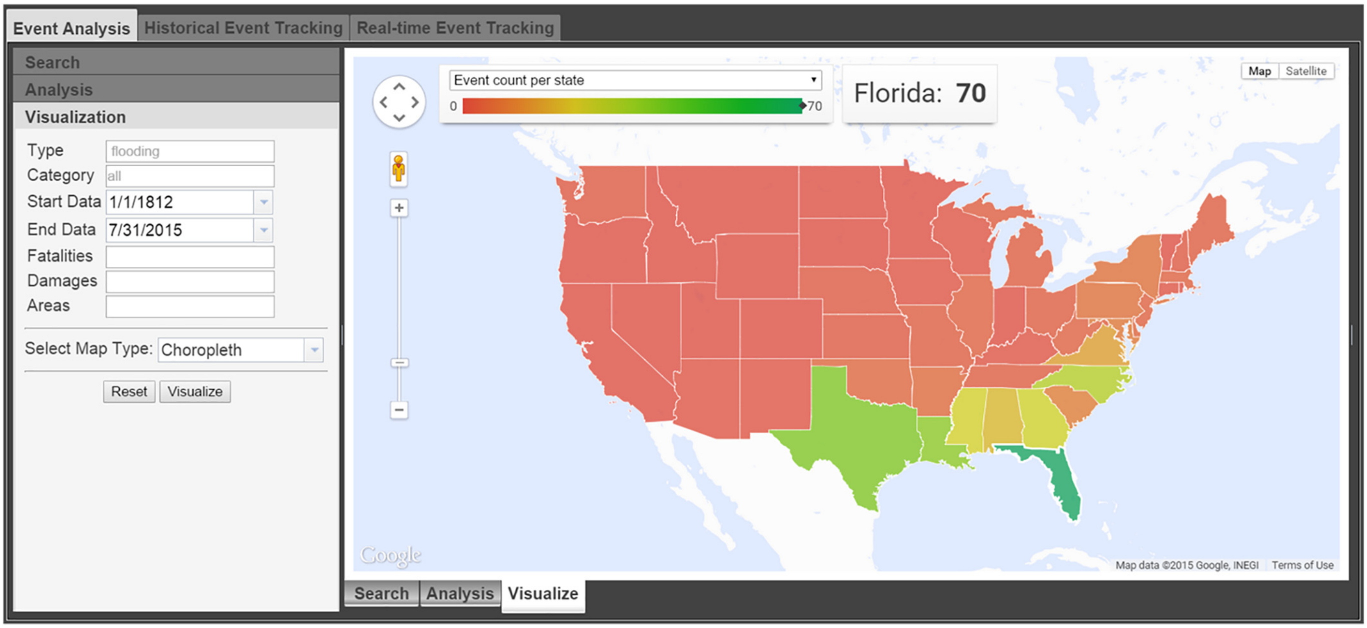
Type: e.g., flooding
Category:
Start Data: 12/30/1995
End Data: 7/31/2015
Fatalities:
Damages:
Areas:
Select Chart Type: Lines

Fig. 4. Different categories (severity levels) of Atlantic hurricanes in the past two decades.

parameters to detect potential hashtags associated with a specific event. A critical parameter is "track interval", indicating the time span for which tweets are monitored and used for detection. For example, if the interval is 1 h, then tweets posted within the past 1 h will be used as the input for the LDA modeling (Section 4.1).

The application server can consistently calculate the frequency of each hashtag with our developed program, and triggers a cloud cluster with Hadoop environment to run the LDA algorithm as a MapReduce job, which can detect topics discussed over the Twitter at the predefined "track interval". For each hashtag (e.g., Sandy), the system will find the topic that it belongs, and associated words (e.g., HurricaneSandy) of the topic, which are hashags posted along with the hashtag. This information is then automatically stored in the database, accessible and displayed on the web interface (Fig. 7 middle panel). For this demonstration, the tweets between "2012-10-27 22:33:27", when the first geo-tagged tweet with the hashtag "#sandy" in the text was posted on Twitter, and "2012-10-27 23:59:59" were used. Within this period, it was detected that there are nine hashtags that have the frequency more than the predefined tweeting threshold (100 in this case), which is an adjustable variable defined in our system to control the number of topics. A high threshold produces less topics, therefore may miss the detection of an emerging disaster event. On the other hand, if the threshold is too small, a large number of topics may be produced. While we tentatively set it as 100, trial-and-error may be needed to determine its value eventually. To run LDA, the number of topics (k) and the number of top words (w) within each topic should be predefined. In our work, we rely on the tweeting threshold to determine the number of frequently posted hashtags, and use this number as the k value. Therefore the topic number is set to nine while running the LDA algorithm. It can be observed that several topics, such as SAT test and studies (using the hashtags #gottaacemysatexam, and #satstudytime), Halloween parties (using hashtag #halloween) emerged in the Twitter.

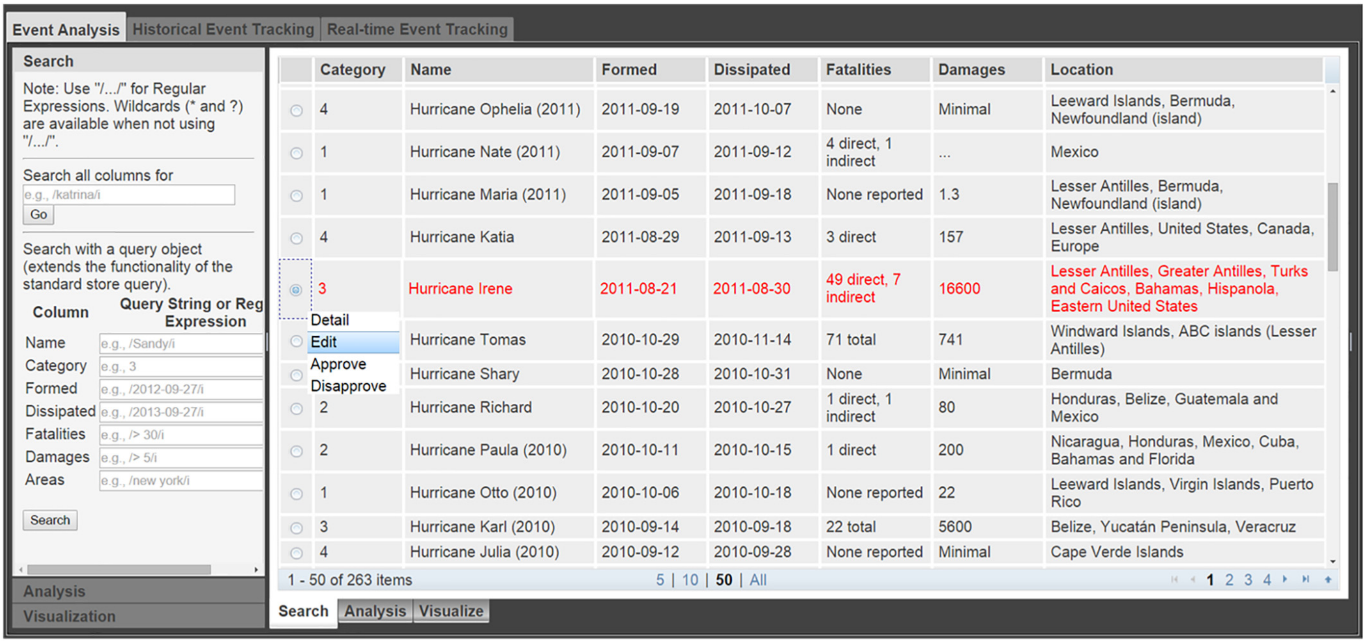
For each topic, the top 10 associated hashtags produced by LDA modeling are displayed in the third column of the table (Fig. 7 middle



Type: flooding
Category: all
Start Data: 1/1/1812
End Data: 7/31/2015
Fatalities:
Damages:
Areas:
Select Map Type: Choropleth

Event count per state
Florida: 70

Fig. 5. Spatial distribution of Atlantic hurricanes.



Note: Use "/.../" for Regular Expressions. Wildcards (* and ?) are available when not using "/.../".

Search all columns for: e.g., /katrina/i

Search with a query object (extends the functionality of the standard store query).

Column	Query String or Reg Expression
Name	e.g., /Sandy/i
Category	e.g., 3
Formed	e.g., /2012-09-27/i
Dissipated	e.g., /2013-09-27/i
Fatalities	e.g., /> 30/i
Damages	e.g., /> 5/i
Areas	e.g., /new york/i

	Category	Name	Formed	Dissipated	Fatalities	Damages	Location
○	4	Hurricane Ophelia (2011)	2011-09-19	2011-10-07	None	Minimal	Leeward Islands, Bermuda, Newfoundland (island)
○	1	Hurricane Nate (2011)	2011-09-07	2011-09-12	4 direct, 1 indirect	...	Mexico
○	1	Hurricane Maria (2011)	2011-09-05	2011-09-18	None reported	1.3	Lesser Antilles, Bermuda, Newfoundland (island)
○	4	Hurricane Katia	2011-08-29	2011-09-13	3 direct	157	Lesser Antilles, United States, Canada, Europe
●	3	Hurricane Irene	2011-08-21	2011-08-30	49 direct, 7 indirect	16600	Lesser Antilles, Greater Antilles, Turks and Caicos, Bahamas, Hispanola, Eastern United States
○		Hurricane Tomas	2010-10-29	2010-11-14	71 total	741	Windward Islands, ABC islands (Lesser Antilles)
○		Hurricane Shary	2010-10-28	2010-10-31	None	Minimal	Bermuda
○	2	Hurricane Richard	2010-10-20	2010-10-27	1 direct, 1 indirect	80	Honduras, Belize, Guatemala and Mexico
○	2	Hurricane Paula (2010)	2010-10-11	2010-10-15	1 direct	200	Nicaragua, Honduras, Mexico, Cuba, Bahamas and Florida
○	1	Hurricane Otto (2010)	2010-10-06	2010-10-18	None reported	22	Leeward Islands, Virgin Islands, Puerto Rico
○	3	Hurricane Karl (2010)	2010-09-14	2010-09-18	22 total	5600	Belize, Yucatán Peninsula, Veracruz
○	4	Hurricane Julia (2010)	2010-09-12	2010-09-28	None reported	Minimal	Cape Verde Islands

1 - 50 of 263 items 5 | 10 | 50 | All 1 2 3 4

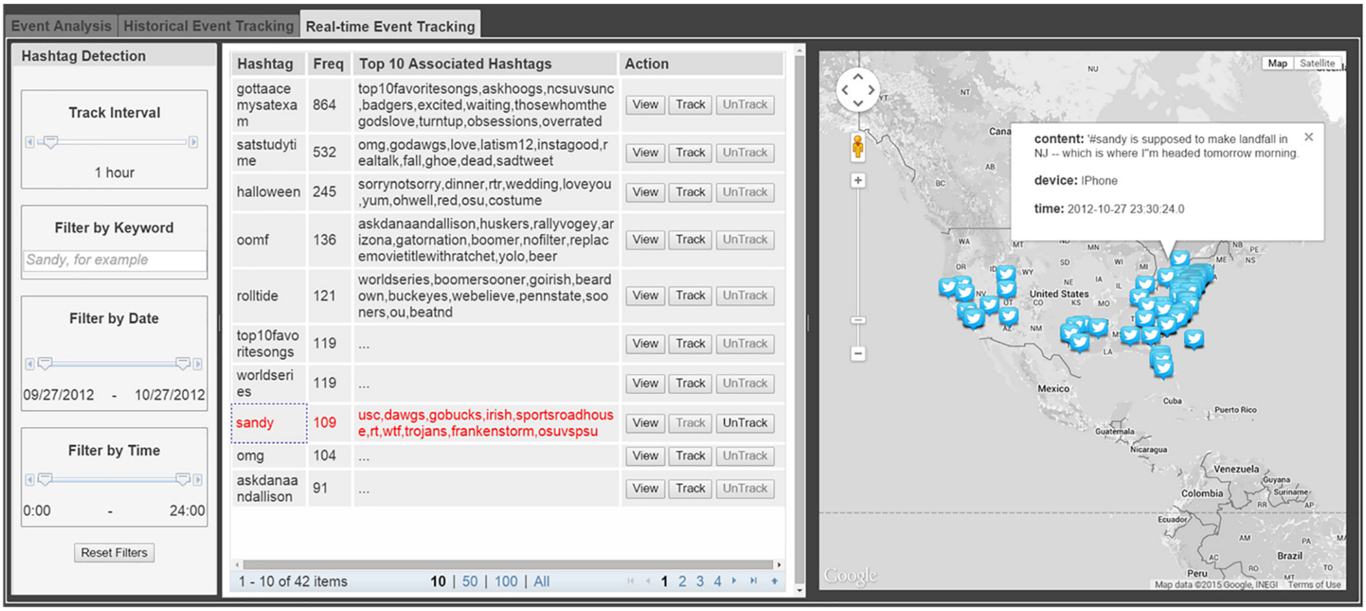
Fig. 6. Check, edit, approve or disapprove the information of an event.

panel). The model output is intuitive, and it can be seen that SAT test hashtag (#gottaacemysatexam) is mostly associated with hashtags, such as #excited (about the results), or #waiting (for the results) etc. We also noticed that topic related to Hurricane Sandy lead by the hashtag "#Sandy" also emerged. Its associated hashtags include "#frankenstorm" with the word storm matched with our predefined sensitive keyword list related to natural hazards. Therefore, the system will post a warning alert to relevant personnel through cell phone text messages or email addresses and start to monitor and track all tweets relevant to Sandy.

Through the user interface, users can also view the spatial distribution of the tweets with a specific hashtag included (Fig. 7 middle panel). If the hashtag is relevant to a disaster, then users can click "Track" button, the system will then continually monitor the streaming tweets, and the application server will retrieve all disaster relevant tweets from the Mongo database, and process and store them in the Postgresql/PostGIS database so they are ready to be accessible. If it is a false alarm, users can click "UnTrack" button, and a revoke process will trigger the application server to remove the monitoring task.

5.3. Identification of real-time event spatial region

In addition to show the spatial distribution of the tweets for an event (Fig. 7), the system can also identity the spatial regions of the event (Fig. 8 and Fig. 9). On Oct 28th, President Barack Obama announced states of emergency including Connecticut, District of Columbia, Maryland, Massachusetts, New Jersey, and New York. Fig. 8a and b shows the intensity maps (or heatmaps) of geo-tagged tweets mentioning about Hurricane Sandy within the same time period as Fig. 7 (between 22:33:27 and



Hashtag	Freq	Top 10 Associated Hashtags	Action
gottaacemysatexam	864	top10favoritesongs,askhoogs,ncsuvsunc,badgers,excited,waiting,thosewhomthegodslove,turntup,obsessions,overrated	View Track UnTrack
satstudytime	532	omg,godawgs,love,latism12,instagood,realtalk,fall,ghoe,dead,sadtweet	View Track UnTrack
halloween	245	sorrynotsorry,dinner,rtr,wedding,loveyou,yum,ohwell,red,osu,costume	View Track UnTrack
oomf	136	askdanaandallison,huskers,rallyvogey,arizona,gatornation,boomer,nofilter,replacemovietitlewithratchet,yolo,beer	View Track UnTrack
rolltide	121	worldseries,boomersooner,goirish,beardown,buckeyes,webelieve,pennstate,sooners,ou,beatnd	View Track UnTrack
top10favoritesongs	119	...	View Track UnTrack
worldseries	119	...	View Track UnTrack
sandy	109	usc,dawgs,gobucks,irish,sportsroadhouse,rt,wtf,trojans,frankenstorm,osuvspsu	View Track UnTrack
omg	104	...	View Track UnTrack
askdanaandallison	91	...	View Track UnTrack

1 - 10 of 42 items 10 | 50 | 100 | All 1 2 3 4

content: '#sandy is supposed to make landfall in NJ -- which is where I"m headed tomorrow morning.

device: IPhone

time: 2012-10-27 23:30:24.0

Fig. 7. Hashtag detection and real-time event Tracking (For better display purpose, the hashtag sign "#" is removed).

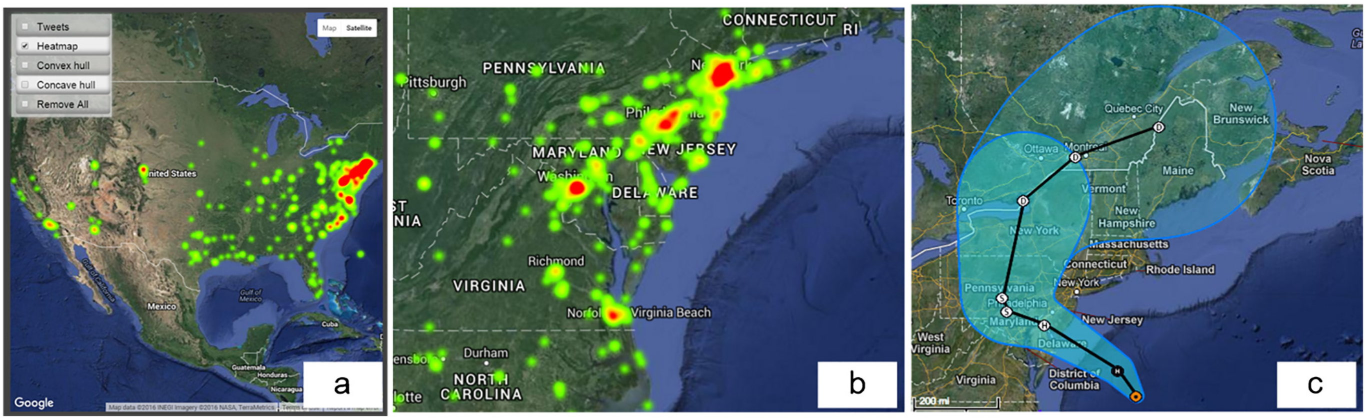


Fig. 8. Intensity map of geo-tagged tweets related to Hurricane Sandy at different scale (a and b), and Hurricane Sandy path map (Courtesy of National Hurricane Center).

23:59:59, Oct 27) in different map scales. Highest densities are found in the U.S states along the east coast, such as New Jersey and New York. In fact, these areas (Fig. 8b) relatively matched with the states declared as the emergency ones and the path of Sandy (Fig. 8c). Therefore, using social media data can reasonably identify and predict the potential affected in advance.

While manually interpreting the intensity maps can help us identify the general regions that may be impacted by an upcoming event, the specific regions could be outlined using the proposed approach which relies on DBSCAN to cluster the geo-tagged tweet points to detect potential regions of events, and then delineates the boundary of each cluster given the points in the cluster (Section 3.2). Ester et al. (1996) show that using a *minpts* value smaller than four may misclassify random points as clusters and a *minpts* value ≥ four unlikely produces clusters of varying results. Additionally, a large number of geo-tagged tweets (≥four) must be generated at an area before it can be considered as a potential event zone. Therefore, it is reasonable to use a *minpts* value of four while running DBSCAN algorithm. Correspondingly, the size of radius (*eps*) is the most influential in affecting the results in DBSCAN. As shown in Fig. 9 displaying the potential region identified in the New York, a large *eps* value (e.g., 10,000 m) tends to produce a large boundary area, and a small value (e.g., 2500 m) may underestimate the regions that are impacted. We also noticed that a value of 7500 m and 10, 000 m produced similar results, which means that when *eps* reaches a certain value, the clustering results become stable. In this case, a value of 5,000 m (Fig. 9d and Fig. 9g) produced the most realistic regions by delineating the Manhattan area as the boundary.

Additionally, different boundary reconstruction algorithms (e.g., convex or non-convex) would produce different region shapes (Fig. 9). Given a set of points, while it exists only one convex hull, there can be many different non-convex (concave) shapes generated using different algorithms and their parameterizations. In this paper, we use a popular non-convex algorithm, known as *chi* algorithm (Duckham et al., 2008) to characterize the boundary of clustered geo-tagged tweets. The *Chi* algorithm needs to supply a length parameter (l), which is the maximum length of border edges for the concave hull. The boundaries produced by the convex hull algorithm may include a large area that overestimates the disaster region (Fig. 9a, c, d,

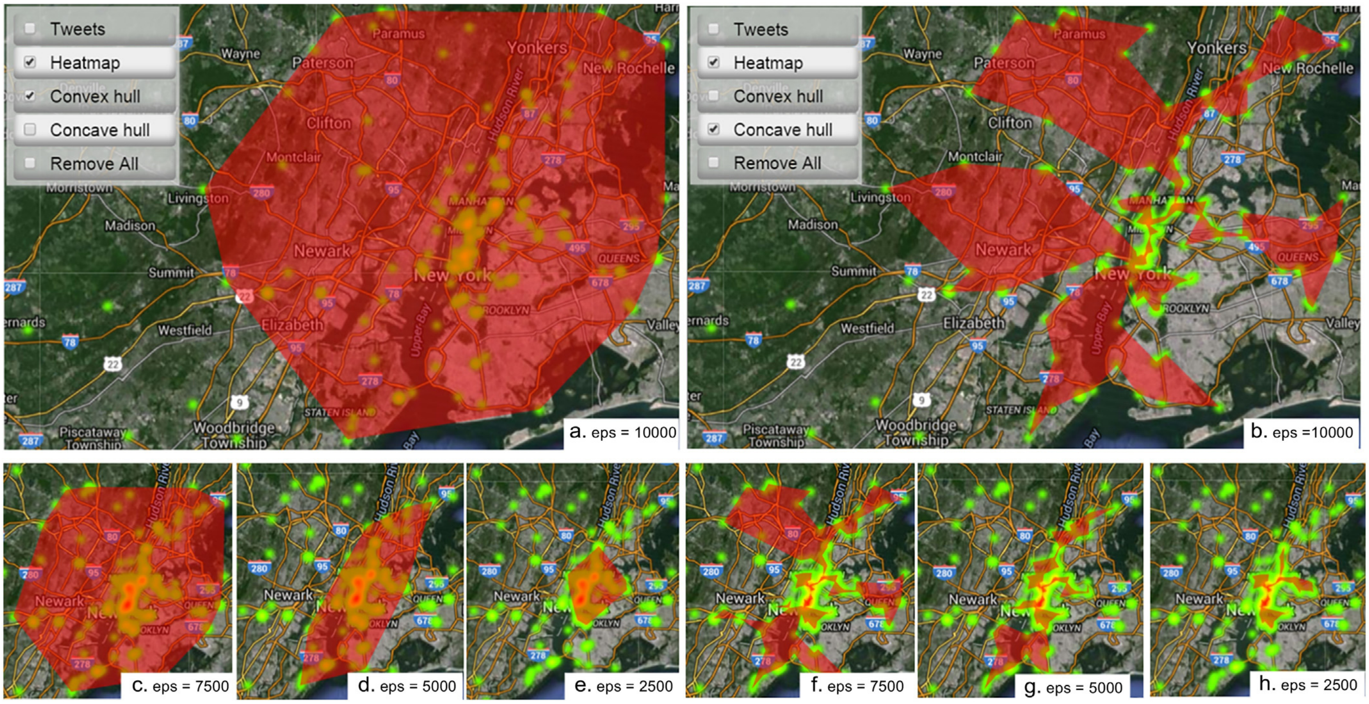
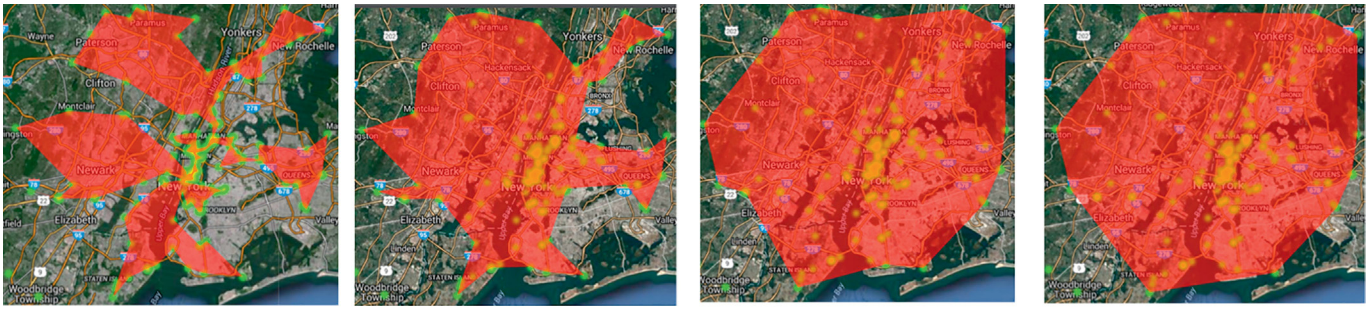


Fig. 9. Identification of spatial region of Hurricane Sandy event in NY using DBSCAN with different eps values (10,000, 7500, 5000 and 2500 m) and using different spatial region construction algorithms (convex hull algorithm: a, c, d, and e; concave hull algorithm [Duckham, Kulik, Worboys, & Galton, 2008] with $l = 0$: b, f, g and h).



a. $l = 0$ b. $l = 10000$ c. $l = 20000$ d. $l = 30000$

Fig. 10. Use varying l values while constructing concave hull regions for the DBSCAN derived cluster with eps value as 10,000.

and e). On the other hand, the boundaries estimated by the concave hull are parameter-dependent, and varying the l values would produce different regions (Fig. 10). A small l value provides a better characterization of the disaster regions associated with numerous tweets (Fig. 9b, f, g and h with $l = 0$). Additionally, a small l value better matches the shape of the hotspot regions (Fig. 10 a). However, it could underestimate the area with a low number of tweets that are characteristics of certain groups (i.e. low income, low education, and elderly) who may lack the tools, skills and motivations to access social media. However, increasing the l value would eventually result in a convex hull (Fig. 10 d). Therefore, it is more flexible to use a concave algorithm to generate the disaster boundaries because it is possible to control the level of generalization and thus produce results that meet specific requirements.

While we discussed some general principles to choose the values of DBSCAN and chi-algorithm parameters, how to select the most appropriate value or algorithm is still less than straightforward. Therefore, we set them as adjustable variables that can be tuned through the web interface and the end users can flexibly make changes depending on different events during the implementation.

Ideally the influence of the background population needs to be filtered out when mapping events 'hotspots' if such population information is available. For instance, the underlying population-at-risk can be accounted for when mapping disease clusters (Shi, 2010). For this study, the underlying background population is the twitter users. However, an accurate estimation of the spatial distribution of the twitter users is lacking. Thus, the intensity of tweets mentioning disaster events (e.g., intensity maps on Figs. 8, 9 and 10) is computed based on the raw numbers of tweets. If data on the spatial distribution of tweet users are available, such information can be used to account for the inhomogeneous background population. For example, the raw numbers of tweets can be normalized using the number of tweet users by computing the ratio of number of disaster-related tweets per thousand tweet users. Among others, the kernel density estimation (KDE) method can be adopted to generate intensity maps and account for population density in the estimation process (Shi, 2010). The Dual KDE method estimates the ratio of the intensity of events and the intensity of the background population (Wang, Ye, & Tsou, 2016). However, the intensity maps are only useful for examining regions that may be impacted by an upcoming disaster through visual analysis and human interpretation. Note that spatial clustering using DBSCAN is a separate process which does not depend on the intensity maps.

One limitation of DBSCAN is that it fails to detect clusters properly when the clusters are of different point densities. In this study, the density of tweets mentioning the disaster events of interests might be different across regions due to the inhomogeneous density of the underlying tweets. When interpreting the clustering results, it should be kept in mind that DBSCAN might fail to detect clusters over regions with low tweet density (e.g., sparsely-populated areas). To address this limitation, other more sophisticated spatial clustering algorithms that can accommodate varying densities, VDBSCAN (Liu et al., 2007) and DECODE (Pei, Jasra, Hand, Zhu, & Zhou, 2009) for example, could be leveraged. VDBSCAN relies on the k-nearest neighbor distance plot to interactively determine the number of clusters of varying densities and the eps value for each cluster. DECODE first computes the probability distribution of the m*th* nearest distances of the points, it then uses reversible jump Markov Chain Monte Carlo to determine the number of clusters and the eps value for each cluster based on the probability distribution. We adopted DBSCAN in this study mainly for its simplicity of implementation. Nevertheless, it is the densely-populated areas that are most subjected to disaster damages and are of most interest for disaster response, and clusters in the densely-populated areas can be identified properly by DBSCAN. The proposed approach is still a useful exploratory tool for disaster response. Other approaches on disaster event detection could also be examined and compared to develop a more sophisticated disaster event detection solution in future. For example, Cervone et al. (2016) divided the study area into grids (e.g., 10 × 10 km^2), and checked if there are a certain number of geo-tagged tweets with specific keywords (e.g., floods, tornadoes) generated within each grid (Cervone et al., 2016).

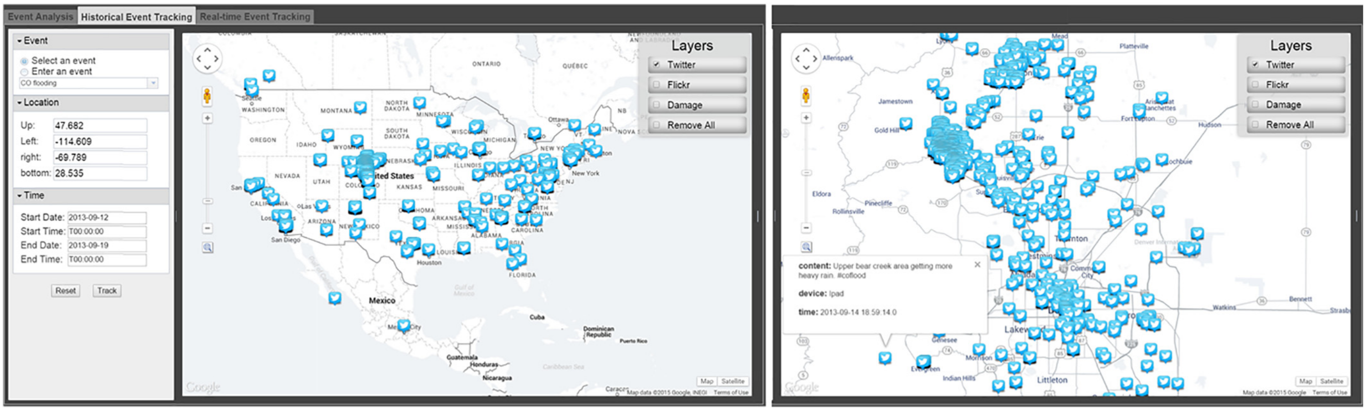
5.4. Analysis of historical events

In addition to tracking a real-time event, the system also supports the analysis of a specific historic event and the 2013 Colorado flooding event is chosen as test case. It was a natural disaster occurring in the U.S. state of Colorado, primarily the Front Range, El Paso County and 10 County, as well as portions of metro Denver. The event starts on Sep 9, 2013, but the intensified situation occurred between Sep 12 and

Table 1
The query criteria for collecting tweets and Flickr photos and collected data for the 2013 Colorado flooding event.

Source	Total number		Geo-referenced		Query criteria*
	Message	User	Message	User	
Twitter	99,515	31,725	1507	681	Hashtags: #coflood || #boulderflood; Content: flood && (Colorado || boulder)
Flickr	1231	34	1150	11	User tags: flood && colorado || boulder Machine tags: flood && colorado || boulder text: flood && (colorado || boulder)

*|| means logic or; && means logic and. As an example, using query criterion "#coflood || #boulderflood" would return all tweets that contain either hashtag "#coflood" or "#boulderflood".



▾ Event
○ Select an event
○ Enter an event
CO flooding

▾ Location
Up: 47.682
Left: -114.609
right: -69.789
bottom: 28.535

▾ Time
Start Date: 2013-09-12
Start Time: T00:00:00
End Date: 2013-09-19
End Time: T00:00:00

☑ Twitter
□ Flickr
□ Damage
□ Remove All

content: Upper bear creek area getting more heavy rain. #coflood

device: Ipad

time: 2013-09-14 18:59:14.0

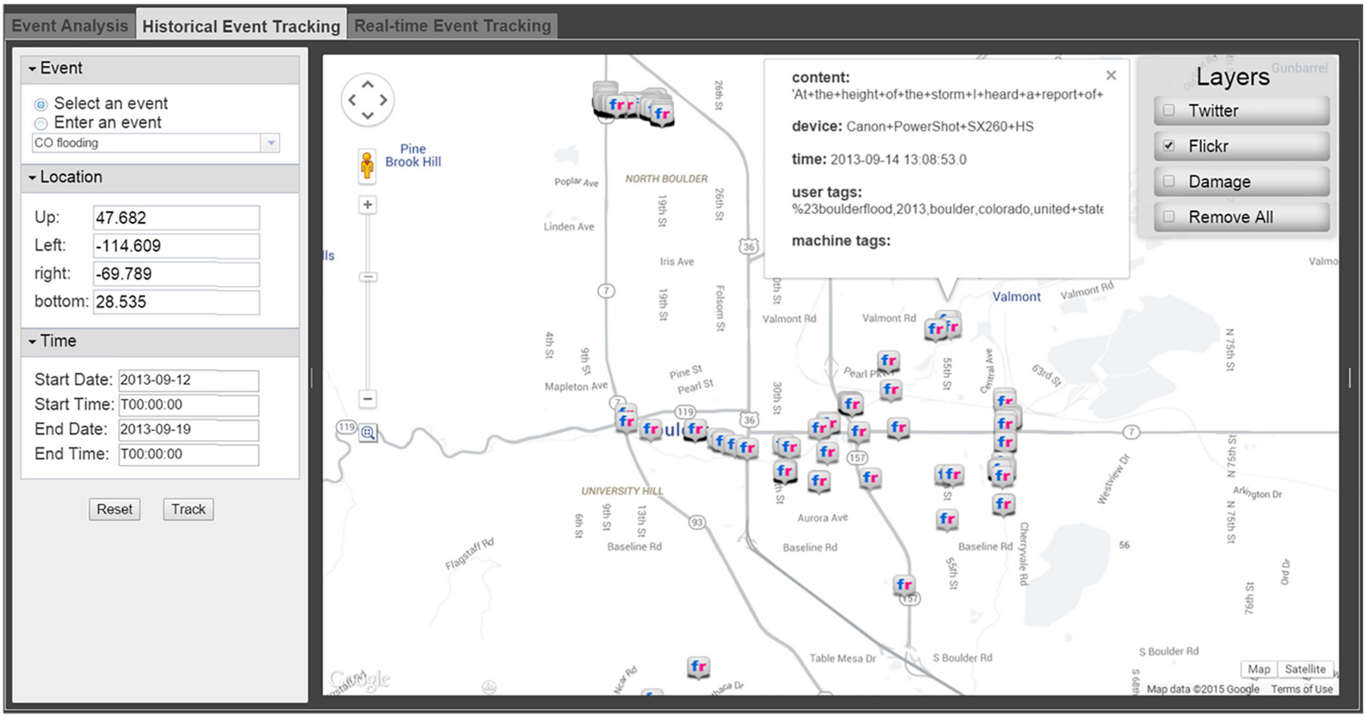
Fig. 11. The spatial distribution of geo-tagged tweets (Left: tweets in U.S; Right: tweets in the Boulder and Denver); The tweets are most distributed in the state of Colorado, and a relatively large number of tweets are also from the locations in the East and West coasts with high population density. Within the state of Colorado, tweets are mostly from the cities of Denver, Boulder and Longmont.

Sep 18, 2013. Social media data collected from Twitter and Flickr during this time span are used. Table 1 shows the query criteria for retrieving tweet and Flickr photos from our data repository and collected data posted. The query criteria are derived using similar approaches for detecting "hot topics" for a real-time event. Since we have the metadata information (e.g., event name) about a historic event, the detected hashtags and keywords using LDA algorithm, and frequency counting program can be matched by these metadata. For this reason, more accurate query criteria can be derived and used to retrieve data for a historic event. From the Table 1, we can see while the tweets are posted by a relatively large number of users, Flickr photos are mostly contributed by a small number of users.

Users can explore the social media in various themes by configuring the input parameters of the query such as temporal information (timestamps when messages were posted), area of interest (AOI, also known as spatial domain information), and analytical methods (visualization or charting), etc. After obtaining query results back, users are able to visualize the results to get an overall view of the spatial and temporal patterns of the tweets (Fig. 11) or photos (Fig. 12) retrieved from the database. The tweets are mostly distributed in the Colorado (CO) state, and a relatively large number of tweets are also from east coast and west coast with high population density. Within the CO, tweets are mostly from the three big cities, including Denver, Boulder and Longmont (Fig. 11). However, most of Flickr photos are posted from the cities of Denver, Boulder and Loveland (Fig. 12).

5.5. Fusing social media and remote sensing data

The system also allows for the fusion of remote sensing data and social media. The data fusion problem consists of merging together heterogeneous data with different temporal and spatial resolutions. The data fusion problem is complex because remote sensing observations have a high spatial but low temporal resolution, and social media have a high temporal but low spatial resolution. In this work, the data fusion problem is accomplished by generating multiple layers of information through kernel smoothing, and then vertically assimilating each layer



▾ Event
● Select an event
○ Enter an event
CO flooding

▾ Location
Up: 47.682
Left: -114.609
right: -69.789
bottom: 28.535

▾ Time
Start Date: 2013-09-12
Start Time: T00:00:00
End Date: 2013-09-19
End Time: T00:00:00

□ Twitter
☑ Flickr
□ Damage
□ Remove All

content: 'At+the+height+of+the+storm+I+heard+a+report+of+

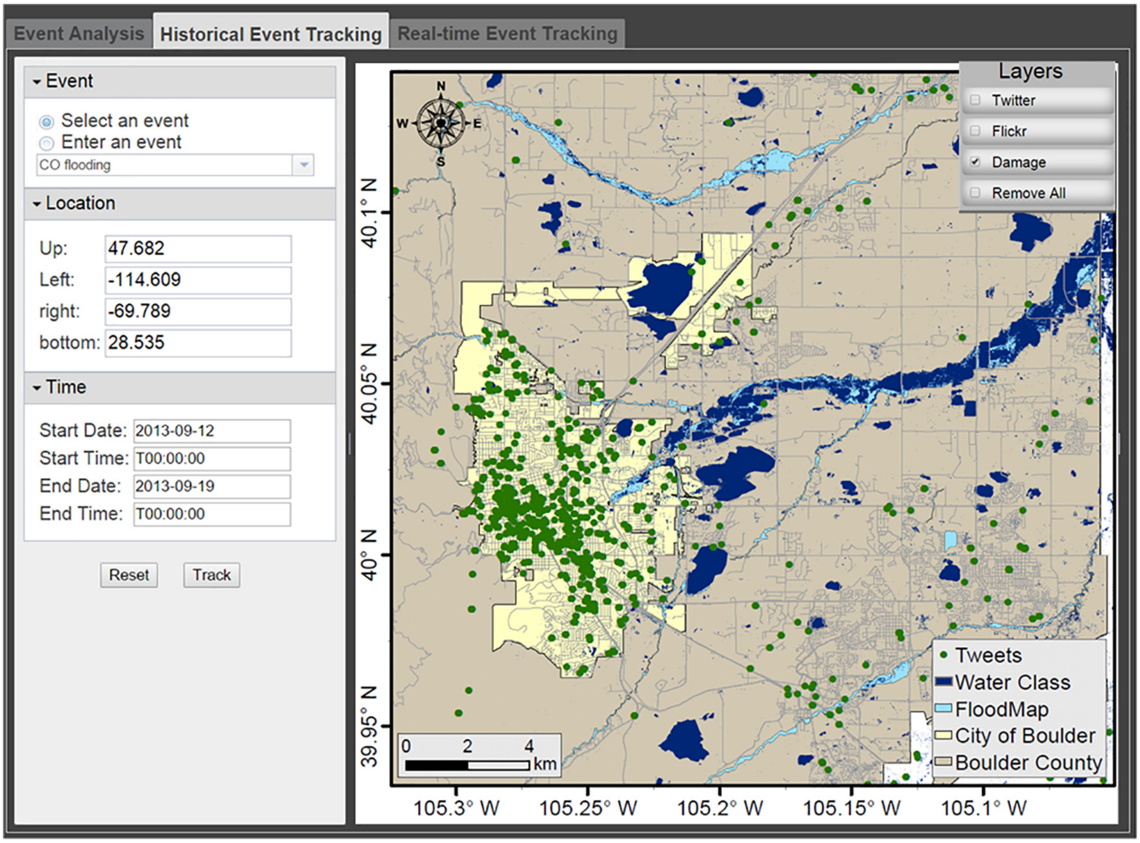
device: Canon+PowerShot+SX260+HS

time: 2013-09-14 13:08:53.0

user tags: %23boulderflood,2013,boulder,colorado,united+state

machine tags:

Fig. 12. The spatial distribution of Flickr photos; Most of photos are posted from the cities of Denver, Boulder and Loveland.



Event

● Select an event
○ Enter an event
CO flooding

Location

Up: 47.682
Left: -114.609
right: -69.789
bottom: 28.535

Time

Start Date: 2013-09-12
Start Time: T00:00:00
End Date: 2013-09-19
End Time: T00:00:00

Layers

□ Twitter
□ Flickr
☑ Damage
□ Remove All

Fig. 13. The overlay of water classification from Satellite data, and tweets in the city of Boulder CO for damage assessment. The FEMA flood map is also shown. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

through weighted averaging. Each layer corresponds to a different data source, specifically from remote sensing and from social media. Because confidence in data may vary with source characteristics, the kernel bandwidth selection can be adjusted for each data type. The basic idea is that the more certain the information given by a kind of data, the higher the chosen kernel bandwidth. For example, aerial or ground images can be weighted according to the amount of water pixels identified by the machine learning classification. Some layers might be weighted more highly because they originated from official sources, or can be quantified and verified. On the other hand, tweets are very subjective and often reflect users' feelings.

We tested the system using data relative to the 2013 Boulder floods, specifically fusing tweets with remote sensing data from the Landsat and Worldview-2 satellite. First, the Normalized Difference Water Index (NDWI) classification was performed for both Landsat and WorldView-3 data. NDWI is a ratio of reflected electro-magnetic radiation sensed using the green and the mid-infrared parts of the energy spectrum. Pixels with high green and low mid-infrared values indicate water presence. The generation of NDWI is fully automatic, and it does not require human intervention.

Geolocated Twitter data were acquired for the region of Boudler CO, accounting to nearly 100,000 tweets over a period of 10 days. A Kernel smoothing was performed over the Twitter data to generate a surface indicating likely damage caused by the flood. Finally, the two layers, the NDWI from satellite data, and the Twitter smoothed surface were averaged to generate a new layer indicating the likeliness of flood damage.

As an example, several scenes from Landsat and WorldView-2 were downloaded during the 2013 Colorado flooding event, and their multispectral channels used to classify water pixels. Fig. 13 shows a map for the city of Boulder and a portion of Boulder County illustrating the water classification from satellite imagery (dark blue), the official FEMA flood map that was released after the event, and the tweets downloaded. The figure shows a good agreement between the two flood maps, however, the satellite estimation underestimates the flood extent due to the presence of thick clouds in the top of the image. When clouds are not present, the classification agrees with FEMA map.

However, there is an absence of areas classified as water in the downtown Boulder area, which is in contrast to the large amount of contributed data available throughout the city of Boulder that indicate different levels of flooding. This result is due to the fact that the flooding downtown Boulder was localized and not fully discernable at the resolution of the satellite data. Furthermore, Twitter data are very subjective and more representative of the users' feelings and perceptions. Therefore, even limited flooding in downtown of the city of Boulder, massive tweets were reported extensively.

The results obtained are important because they show that damage assessment can be effectively performed by fusing remote sensing and social media. Furthermore, the contributed data capture the subjective perception of the people who experience a flood event. Fusing remote sensing data and Twitter and other VGI presents the additional challenge of fusing quantitative data acquired through precise measurements of EM radiation, with qualitative data that is very subjective and only based on perceptions (Schnebele & Cervone, 2013).

6. Conclusion

This paper presents a novel framework to support the analysis of historical disaster events, and the real-time event detection and tracking of new events. Massive spatiotemporal data from social media streams and remote sensing are generated continuously and dynamically, posing new challenges and opportunities to study disasters. To meet the dynamic computing requirements of disaster analysis for real-time events, cloud computing is proposed as the infrastructure to provide on-demand and flexible computing resources.

A prototype is implemented that queries Wikipedia to gather data about past events, or even detects new unfolding events. Once an event is selected, data from multiple sources, including remote sensing and social media streams, are downloaded and analyzed in real time using a cloud computing platform. Initial research results for two case studies show a great potential to support disaster analysis and management that rely on heterogeneous data from multiple sources and require a resilient and scalable computing platform. The present work provides a general solution that it is not event specific, and can be used both for retrospective analysis and for real time monitoring and decision making.

In this paper, Wikipedia is primarily used as a data source to obtain metadata (e.g., event type, location, and time) for historical natural, the system could also use Wikipedia to harvest disaster information in real time in future. The collection and analysis of crawling data can occur automatically as a Wikipedia page is created for the hazard. It is of interest to note that Wikipedia pages about medium to large disasters, in most cases, are created and updated in real time. Remote sensing data become integral part of the disaster assessment process (Schnebele & Cervone, 2013) when they are available. However, orbital and atmospheric limitations often hinder our ability to collect real time data using satellites. We envision that while satellite remote sensing data can provide only snapshots at discrete temporal points, aerial remote sensing from airplanes and/or Unmanned Aerial Vehicles (UAVs) can help provide a more continuous data stream that can be integrated in real time. Therefore, while the Boulder test case (Section 5.5) was performed using only 4 satellite images collected over a period of one week, the proposed data fusing methodology can utilize remote sensing data in real time, which are likely to become widely available in the future.

To detect emerging topics over the social media, we propose a tracking interval and a tweeting threshold to detect significant hashtags. However, this approach could produce many hot topics that are not associated with any developing disasters. To reduce the noise in the system, it is possible to detect relevant topics by further filtering tweets according to their time-stamps, and by selecting only tweets that cluster in time and space when available.

Acknowledgement

Work performed under this project has been supported by grants from the Wisconsin Alumni Research Foundation, University of Wisconsin-Madison (Project No.: #PRJ93X), Department of Energy (Project No.:DE-AR0000717), and Office of Naval Research (Project No.: #N00014-16-1-2543).

References

Ashbrook, D., & Starner, T. (2003). Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, *7*(5), 275–286.

Ashktorab, Z., Brown, C., Nandi, M., & Culotta, A. (2014). *Tweedr: Mining twitter to inform disaster response. Proc. of ISCRAM*.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research*, *3*, 993–1022.

Cameron, M. A., Power, R., Robinson, B., & Yin, J. (2012). *Emergency situation awareness from twitter for crisis management. Paper presented at the Proceedings of the 21st International Conference Companion on World Wide Web*.

Cervone, G., & Manca, G. (2011). Damage assessment of the 2011 Japanese tsunami using high-resolution satellite data. *Cartographica: The International Journal for Geographic Information and Geovisualization*, *46*(3), 200–203.

Cervone, G., Sava, E., Huang, Q., Schnebele, E., Harrison, J., & Waters, N. (2016). Using Twitter for tasking remote-sensing data collection and damage assessment: 2013 boulder flood case study. *International Journal of Remote Sensing*, *37*, 100–124.

Cutter, S. L. (2003). GI science, disasters, and emergency management. *Transactions in GIS*, *7*(4), 439–446.

De Albuquerque, J. P., Herfort, B., Brenning, A., & Zipf, A. (2015). A geographic approach for combining social media and authoritative data towards identifying useful information for disaster management. *International Journal of Geographical Information Science*, *29*, 667–689.

De Longueville, B., Smith, R. S., & Luraschi, G. (2009). *Omg, from here, i can see the flames!: A use case of mining location based social networks to acquire spatio-temporal data on forest fires. Proceedings of the 2009 international workshop on location based social networks. ACM*. (pp. 73–80), 73–80.

Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. *Communications of the ACM*, *51*(1), 107–113.

Duckham, M., Kulik, L., Worboys, M., & Galton, A. (2008). Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. *Pattern Recognition*, *41*(10), 3224–3236.

Ester, M., Kriegel, H. -P., Sander, J., & Xu, X. (1996). *A density-based algorithm for discovering clusters in large spatial databases with noise. Paper presented at the Kdd*.

Evangelinos, C., & Hill, C. (2008). Cloud computing for parallel scientific HPC applications: Feasibility of running coupled atmosphere-ocean climate models on Amazon's EC2. *Ratio*, *2*(2.40), 2–34.

Fuchs, G., Andrienko, N., Andrienko, G., Bothe, S., & Stange, H. (2013). *Tracing the German centennial flood in the stream of tweets: first lessons learned. Proceedings of the second ACM SIGSPATIAL International Workshop on crowdsourced and volunteered geographic information. ACM*. (pp. 31–38), 31–38.

Gao, H., Barbier, G., & Goolsby, R. (2011). Harnessing the crowdsourcing power of social media for disaster relief. *IEEE Intelligent Systems*, *3*, 10–14.

Gao, S., Li, L., Li, W., Janowicz, K., & Zhang, Y. (2017). Constructing gazetteers from volunteered big geo-data based on Hadoop, 2014. *Computers; Environment and Urban Systems*, *61*, 172–186.

Guan, Q., Zeng, W., Gong, J., & Yun, S. (2014). pRPL 2.0: Improving the parallel raster processing library. *Transactions in GIS*, *18*(S1), 25–52.

Huang, Q., & Cervone, G. (2016). Usage of social media and cloud computing during natural hazards. In T. C. Vance, N. Merati, C. Yang, & M. Yuan (Eds.), *Cloud Computing in Ocean and Atmospheric Sciences* (pp. 297–324). Academic Press.

Huang, Q., Cervone, G., Jing, D., & Chang, C. (2015). *DisasterMapper: A CyberGIS framework for disaster management using social media data*Proceedings of ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data. Seattle, WA, USA: ACM.

Huang, Q., & Xiao, Y. (2015). Geographic situational awareness: Mining tweets for disaster preparedness, emergency response, impact, and recovery. *International Journal of Geo-Information*, *4*(3), 19. http://dx.doi.org/10.3390/ijgi4031549.

Huang, Q., & Xu, C. (2014). A data-driven framework for archiving and exploring social media data. *Annals of GIS*, *20*(4), 265–277.

Huang, Q., Yang, C., Benedict, K., Chen, S., Rezgui, A., & Xie, J. (2013). Utilize cloud computing to support dust storm forecasting. *International Journal of Digital Earth*, *6*(4), 338–355.

Huang, Q., Yang, C., Liu, K., Xia, J., Xu, C., Li, J., ... Li, Z. (2013). Evaluating open-source cloud computing solutions for geosciences. *Computers & Geosciences*, *59*, 41–52.

Imran, M., Elbassuoni, S., Castillo, C., Diaz, F., & Meier, P. (2013). *Practical extraction of disaster-relevant information from social media. Paper presented at the Proceedings of the 22nd International Conference on World Wide Web Companion*.

Jain, S. (2015). *Real-time social network data mining for predicting the path for a disaster*.

Joyce, K. E., Belliss, S. E., Samsonov, S. V., McNeill, S. J., & Glassey, P. J. (2009). A review of the status of satellite remote sensing and image processing techniques for mapping natural hazards and disasters. *Progress in Physical Geography*, *33*(2), 183–207.

Kreps, J., Narkhede, N., & Rao, J. (2011). *Kafka: A distributed messaging system for log processing. Proceedings of the NetDB*. (pp. 1–7), 1–7.

Kumar, S., Barbier, G., Abbasi, M. A., & Liu, H. (2011). *TweetTracker: An analysis tool for humanitarian and disaster relief. Proceedings of the Fifth International Conference on Weblogs and Social Media, ICWSM, Barcelona, Spain, 17–21 July 2011*.

Li, Z., Yang, C., Huang, Q., Liu, K., Sun, M., & Xia, J. (2017). Building model as a service to support geosciences. *Computers, Environment and Urban Systems*, *61*, 141–152.

Liu, P., Zhou, D., & Wu, N. (2007). VDBSCAN: Varied Density Based Spatial Clustering of Applications with Noise. *Proceedings of 2007 International Conference on Service Systems and Service Management* (pp. 1–4) (IEEE 2014, Jun 09-11).

Mandel, B., Culotta, A., Boulahanis, J., Stark, D., Lewis, B., & Rodrigue, J. (2012). *A demographic analysis of online sentiment during hurricane Irene. Proceedings of the Second Workshop on Language in Social Media*. (pp. 27–36)Association for Computational Linguistics, 27–36.

Padmanabhan, A., Wang, S., Cao, G., Hwang, M., Zhang, Z., Gao, Y., ... Liu, Y. (2014). FluMapper: A cyberGIS application for interactive analysis of massive location-based social media. *Concurrency and Computation: Practice and Experience*, *26*, 2253–2265.

Pei, T., Jasra, A., Hand, D. J., Zhu, A. X., & Zhou, C. (2009). DECODE: A new method for discovering clusters of different densities in spatial data. *Data Mining and Knowledge Discovery*, *18*, 337–369.

Pu, C., & Kitsuregawa, M. (2013). *Big data and disaster management a report from the JST/NSF Joint Workshop*. Georgia Institute of Technology.

Ranjan, R. (2014). Streaming big data processing in datacenter clouds. *IEEE Cloud Computing*, *1*, 78–83.

Roth, R. E. (2012). Cartographic interaction primitives: Framework and synthesis. *The Cartographic Journal*, *49*(4), 376–395.

Roth, R. E. (2013). Interactive maps: What we know and what we need to know. *Journal of Spatial Information Science*, *2013*(6), 59–115.

Sakaki, T., Okazaki, M., & Matsuo, Y. (2010). *Earthquake shakes Twitter users: Real-time event detection by social sensors. Paper presented at the Proceedings of the 19th international conference on World wide web*.

Schadt, E. E., Linderman, M. D., Sorenson, J., Lee, L., & Nolan, G. P. (2010). Computational solutions to large-scale data management and analysis. *Nature Reviews Genetics*, *11*(9), 647–657.

Schnebele, E., & Cervone, G. (2013). *Improving remote sensing flood assessment using volunteered geographical data*.

Schnebele, E., Cervone, G., Kumar, S., & Waters, N. (2014). Real time estimation of the Calgary floods using limited remote sensing data. *Water*, *6*(2), 381–398.

Schnebele, E., Oxendine, C., Cervone, G., Ferreira, C. M., & Waters, N. (2015). Using non-authoritative sources during emergencies in Urban areas. *Computational Approaches for Urban Environments* (pp. 337–361). Springer.

Shi, X. (2010). Selection of bandwidth type and adjustment side in kernel density estimation over inhomogeneous backgrounds. *International Journal of Geographical Information Science*, *24*, 643–660.

Shook, E., Wang, S., & Tang, W. (2013). A communication-aware framework for parallel spatially explicit agent-based models. *International Journal of Geographical Information Science*, *27*(11), 2160–2181.

Sutton, J., Palen, L., & Shklovski, I. (2008). *Backchannels on the front lines: Emergent uses of social media in the 2007 southern California wildfires. Proceedings of the 5th International ISCRAM Conference.*

Tang, W., & Feng, W. (2017). Parallel map projection of vector-based big spatial data: Coupling cloud computing with graphics processing units. *Computers, Environment and Urban Systems*, *61*, 187–197.

Tang, W., Feng, W., Jia, M., Shi, J., Zuo, H., Stringer, C. E., & Trettin, C. C. (2017). A cyber-enabled spatial decision support system to inventory Mangroves in Mozambique: coupling scientific workflows and cloud computing. *International Journal of Geographical Information Science*, *31*(5), 907–938.

Tang, W., & Jia, M. (2014). Global sensitivity analysis of a large agent-based model of spatial opinion exchange: A heterogeneous multi-GPU acceleration approach. *Annals of the Association of American Geographers*, *104*(3), 485–509.

Velev, D., & Zlateva, P. (2012). Use of social media in natural disaster management. *Internatonal Proceeedings of Economic Development and Research*, *39*, 41–45.

Wang, S. (2010). A CyberGIS framework for the synthesis of cyberinfrastructure, GIS, and spatial analysis. *Annals of the Association of American Geographers*, *100*(3), 535–557.

Wang, Z., Ye, X., & Tsou, M. H. (2016). Spatial, temporal, and content analysis of Twitter for wildfire hazards. *Natural Hazards*, *83*(1), 523–540.

Yang, C., Wu, H., Huang, Q., Li, Z., & Li, J. (2011). Using spatial principles to optimize distributed computing for enabling the physical science discoveries. *Proceedings of the National Academy of Sciences*, *108*(14), 5498–5503.

Zelenkauskaite, A., & Simões, B. (2014). *Big data through cross-platform interest-based interactivity. Proceedings of the 2014 International Conference on Big Data and Smart Computing (BIGCOMP), pp. 191–196, Jan 15th–17th, 2014. IEEE.*

Zhang, Q., Chen, Z., & Yang, L. T. (2015). A nodes scheduling model based on Markov chain prediction for big streaming data analysis. *International Journal of Communication Systems*, *28*, 1610–1619.