CROWDSAFE: Crowd Sourcing of Crime Incidents and Safe Routing on Mobile Devices (Demo Paper)

Sumit Shah, Fenye Bao, Chang-Tien Lu, Ing-Ray Chen
Department of Computer Science, Virginia Tech, USA
{sshah,baofenye,ctlu,irchen}@vt.edu

ABSTRACT
Crowd sourcing is based on a simple but powerful concept: Virtually anyone has the potential to plug in valuable information. The concept revolves around large groups of people or community handling tasks that have traditionally been associated with a specialist or small group of experts. With the advent of the smart devices, many mobile applications are already tapping into crowd sourcing to report community issues and traffic problems, but more can be done. While most of these applications work well for the average user, it neglects the information needs of particular user communities. We present CROWDSAFE, a novel convergence of Internet crowd sourcing and portable smart devices to enable real time, location based crime incident searching and reporting. It is targeted to users who are interested in crime information. The system leverages crowd sourced data to provide novel features such as a Safety Router and value added crime analytics. We demonstrate the system by using crime data in the metropolitan Washington DC area to show the effectiveness of our approach. Also highlighted is its ability to facilitate greater collaboration between citizens and civic authorities. Such collaboration shall foster greater innovation to turn crime data analysis into smarter and safe decisions for the public.

Categories and Subject Descriptors
H.3 [**Information Storage and Retrieval**]: Information Storage and Retrieval

Keywords
mobile interface, spatio-temporal search engine, crime analysis

1. INTRODUCTION
"If you see something, say something", is an ad slogan by the New York City Metropolitan Transportation Agency (MTA). The city cannot dispatch its police force on every train, bus, and subway so it ran the campaign to ask commuters to be the eyes and ears of public safety [3]. With significant increase in global terrorism and organized crime incidents, it is even more critical to be able to search, report, and analyze location specific crime data. In general people may also be concerned to know about various crime incidents happening in their vicinity increasing awareness and resilience. Hence, retrieving crime incidents associated with a particular location will prove to be of as significant use as reporting of new crime incidents. However, with the aid of ubiquitous mobile applications, citizen reporting of crime incidents can be made easier and richer [1][2].

This motivates us to implement an easy to use mobile application, CROWDSAFE, that leverages crowd intelligence by allowing users to report and search location specific crime incidents. CROWDSAFE integrates and refines crime incidents reported by users and those published by the government. Mining of this information allowed us to go beyond basic crowd sourcing. Hence, we developed a safety router within CROWDSAFE that leverages crime data reported by the users and the government to provide a safe driving route between two locations within a city. Additional value added analytics are provided in the form of crime hotspots and clusters within a city and historical crime statistics in a dashboard like mobile interface. We currently leverage crime data for the Washington DC metro area [4]. Google's Android platform is used to deploy the mobile interface.

Similar applications for the crime domain are in existence. One such application is Citizens Connect for iPhone. It allows reporting of service requests such as removing graffiti, filling potholes, and fixing traffic lights. The service provided by CrimeReports.com allows users to search for crime information based on location, crime type and date range. As far as we know, there are very few existing applications using the crime data to provide value added services, like safety routing, crime hotspots and dashboard etc. on mobile devices.

The major contributions of CROWDSAFE are as follows:

- **Crowd sourcing of crime incidents:** Utilization of a hybrid index to store textual, spatial and temporal crime data and apply it to a specialized search engine along with a very novel mobile user interface that displays textual, spatial and temporal results.
- **Safety router:** Extension of crowd sourcing to provide a novel feature, a Safety Router, allows users to chart out a safe route between two locations within a city thus avoiding high crime areas.
- **Crime analytics:** Crime hotspots, clusters and historical statistics in a dashboard interface alleviate much of the time consuming and manual tasks of knowledge discovery and aid decision making.
- **Mobile platform:** Provision of crowd sourcing, safety router and analytics via the ubiquitous mobile platform promotes effective and efficient use of crime data by citizens and the government.

2. SYSTEM ARCHITECTURE
In this section we describe the system architecture of CROWDSAFE as pictured in Figure 1. At the high level, there are

ACM SIGSPATIAL GIS'11, November 1-4, 2011. Chicago, IL, USA
Copyright © 2011 ACM ISBN 978-1-4503-1031-4/11/11…$10.00

three main components of CROWDSAFE: data processing, application server and the mobile interface.

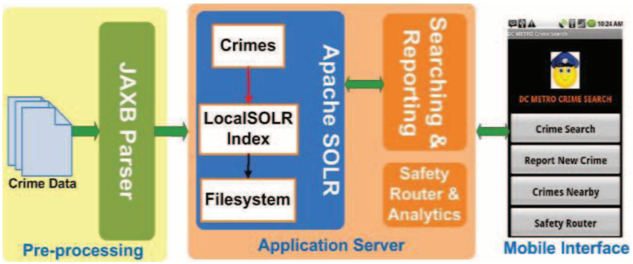


Figure 1: System Architecture

Data Processing: This component pre-processes crime data sets and transforms them into search records to be indexed by the search engine. This component also weeds out bad data such as invalid dates, locations, etc. JAVA API for XML Binding (JAXB) is used to parse and transform raw XML based crime data.

Application Server: This is the core server component of the application. It provides indexing, searching and reporting functionalities. It also provides access to the safety router and analytics component which are separately implemented. Since real-time crime retrieval and safety routing along with analytics are functionally separate features, we split the underlying architecture accordingly while making it seamless to the end user.

The retrieval and reporting features are built using Apache Solr [5] along with LocalSolr (for spatial indexing) as the underlying search engine. Apache Solr is an open source enterprise search server based on Apache Lucene, with XML/HTTP and JSON APIs, hit highlighting, faceted search, caching, replication, a web administration interface and many more features. Solr is highly scalable, providing distributed search and index replication.

The safety router and the analytics features utilize the PostGreSQL with PostGIS database as the underlying information retrieval platform. Apache Tomcat is used as the application server. Figure 2 depicts the architecture of the safety router and the analytics component.

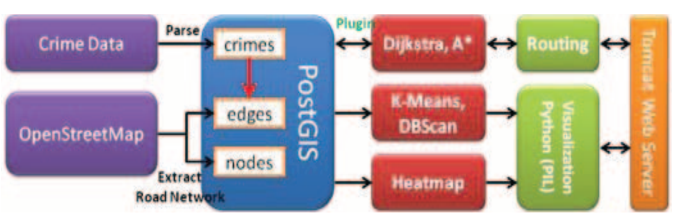


Figure 2: Safety Router and Analytics

The crime data is parsed and stored into a PostGIS database. We assign equal weight to all crime types in the current implementation. Work is in progress to design a weighting scheme for different types of crime. The road network for the Safety Router is obtained by extracting OpenStreetMap downloaded from CloudMade. We use density based method to generate heatmap for crime incidents in each month. Python Image Library (PIL) is used for visualization.

Mobile Interface: This is the primary user interface of the application. It acts as a client to the core search, reporting, safety routing, and analytical features of the application. The user interface is built as a native application for Google's Android platform. The client server interaction is achieved by exposing Representational State Transfer (RESTful) Web services on the server that can be remotely accessed from the mobile device.

3. FEATURES

CROWDSAFE allows mobile users to search for crimes near a given location and report new crime incidents. We develop a novel feature with crime data, namely, a safety router. When given two locations, our approach ranks the routes between these two locations according to route length and safety level in the neighborhood area of the route. The application also provides users with value added analytics such as crime hot spots, crime clusters within a city, and crime statistics for a city.

3.1 CROWD SOURCING OF CRIME DATA

Upon launching the search feature of the application, the user can enter the search criteria (keyword, location and date range) and choose the desired ranking for the search results. Figure 3 shows different views to display the search results and reporting a new crime incident.

(a) List (b) Map

(c) Crime Details (d) Crime Report

Figure 3: Search & Report

The results screen provides a list view (Figure 3(a)) and a map view (Figure 3(b)) (native Google Maps) of the search results. Clicking on any item in the list displays additional details about the crime (Figure 3(c)). The user is presented with the search results organized as a list or plotted on the map. The user can select a crime from the results and view additional details about the crime. A user can also choose to report a new crime incident as shown in Figure 3(d).

A ranker is used to improve the quality of the query results. In this system, we provide a tunable combination of three independent ranking methods to rank query results. (1) The vector model's inner product is used instead of the cosine similarity as the measure method for the relevance between textual query and each search result. (2) The spatial ranking is based on Euclidean distance. (3) We provide both *increasing* and *decreasing* ranking based on the temporal information of crime record. The *increasing* ranking gives higher score to old record and the *decreasing* ranking gives higher score to new record.

We provide a weighted linear combination of them. The weight for score from each ranking method is tunable by user. Let's denote the weight for score from each ranking method as follows:

- w_1: weight for the score from textual ranking method;
- w_2: weight for the score from spatial distance ranking method;
- w_3: weight for the score from temporal ranking method (*increasing* or *decreasing*);

w_1, w_2, and w_3 have the same range [0, 100]. We use the normalized ranking scores of independent ranking methods, which have the same range [0, 1] and the same physical meaning: the record is ranked higher when it has a lower normalized ranking score of independent ranking method. Finally, the joint ranking method employs following formula:

$$R_{joint} = w_1 \cdot R_{text} + w_2 \cdot R_{spatial} + w_3 \cdot R_{temporal} \quad (1)$$

The joint ranking score has a range [0,300]. The record with smaller value of joint ranking score is ranked higher.

Considering slow wireless transmission, for crime searching, at most 10 records are returned each time. The system loads more results when the user scrolls beyond the provided results.

3.2 SAFETY ROUTER

The Safety Router allows the user to enter the locations of a source and a destination and provides the user with the safest route between the two locations. As far as we know, there is little existing work for safety route discovery. Most existing work considers shortest route or fastest route. In our system, we consider both physical distance and safety level in route discovery. Users can customize the weights (safety level) to trade distance off for safety. We develop two routing algorithms, based on Dijkstra and A* shortest path algorithms. Both are implemented as PostGIS plugins. We examine the performance of these two algorithms. A* algorithm employs heuristic searching to find the shortest path. In general, it should perform better than Dijkstra algorithm. However, A* involves extra computation to calculate the distance to the target node. It may not always outperform Dijkstra algorithm, especially when the network is relatively small.

Routing is essentially an optimization problem to minimize an objective function, which indicates the cost of the route. We categorize the routing methods according to three different edge-cost-weighting schemes: shortest path, safest path, and best path.

A. Shortest Path Scheme

In this scheme, each edge is weighted only by the length. Equation 2 is the objective function for this scheme.

$$\underset{s \xrightarrow{E} t}{MIN} \left\{ \sum_{e \in E} length(e) \right\} \quad (2)$$

where s and t are source and target nodes respectively. The algorithm tries to find a set of edges, E, connecting the source and target nodes and in which the total length of edges is minimal.

B. Safest Path Scheme

In this scheme, each edge is weighted only by its number of crime incidents within its distance-*d* buffer area. Equation 3 gives the objective function for this scheme.

$$\underset{s \xrightarrow{E} t}{MIN} \left\{ \sum_{e \in E} crime(e) \right\} \quad (3)$$

where $crime(e)$ represents the total number of crime incidents within distance-*d* buffer area of the road segment e.

C. Best Path Scheme

This scheme combines the above two edge weighting schemes by specifying a *safety level* β. Equation 4 gives the objective function for this scheme.

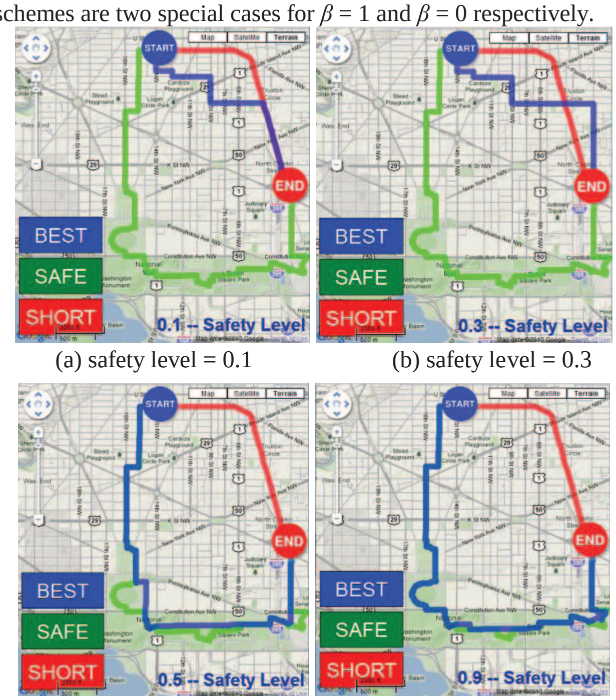
$$\underset{s \xrightarrow{E} t}{MIN} \left\{ \sum_{e \in E} \{(1-\beta) \cdot nlenght(e)\} + \{\beta \cdot ncrime(e)\} \right\} \quad (4)$$

where $nlength(e)$ is the normalized edge length and $ncrime(e)$ is the normalized crime number. That is to say,

$nlength(e) = length(e)/\max_{e' \in E}\{length(e')\}$ and

$ncrime(e) = crime(e)/\max_{e' \in E}\{length(e')\}$

The *safety level* β is a real number in range [0, 1]. The above two schemes are two special cases for $\beta = 1$ and $\beta = 0$ respectively.



(a) safety level = 0.1 (b) safety level = 0.3

(c) safety level = 0.5 (d) safety level = 0.9

Figure 4: Safety Router

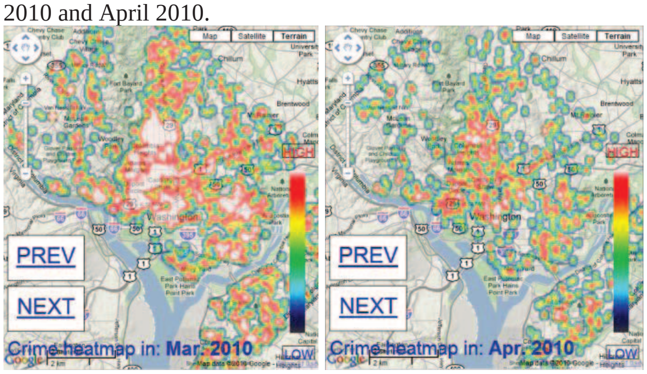
In Figure 4, the red and blue circles represent the start and end locations respectively. The blue, green and red lines are best path, safest path and the shortest path, respectively. The value safety level β is displayed on the bottom-right of the screen. User can hide or show each path by clicking the corresponding button on the bottom-left screen.

As we can see, the red path is the shortest one but not necessarily safe. The green path goes around reaching the end point. It has smallest number of crime incidents along the path but the total length of this path is very long. The blue path, called best path, is a result of the tradeoff between the shortest path and safety path. By changing safety level of the best path, it approaches the shortest path when safety level is low (0.1 in Figure 4 (a) and 0.3 in Figure 4 (b)) and approaches the safety path when safety level is high (0.5 in Figure 4 (c) and 0.9 in Figure 4 (d)).

In general, A* algorithm is better than Dijkstra algorithm. However, A* has extra overhead for distance computation. We implement both of these two algorithms and compare the performance of them. We adopt Dijkstra as the default since it performs better in our current system which has a relative small road network.

3.3 CRIME ANALYTICS

As part of the value added analytics, we provide high crime density areas in Washington DC as **crime hotspots**. We use density based method to generate heatmap for crime incidents in each month. Python Image Library (PIL) is used for visualization. Multiple resolutions for the same heatmap are generated, so that we can load low resolution heatmap when zoom level is low to reduce response time and load high resolution heatmap when zoom level is high. Figure 5 shows the crime hot spots in March 2010 and April 2010.



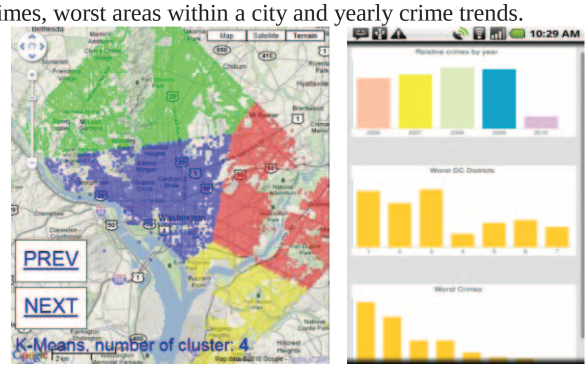
(a) March 2010 (b) April 2010

Figure 5: Hotspots

There are several practical uses of crime hotspots. Hotspots help law enforcement agencies identify high-crime areas, types of crime being committed, and the best way to respond. Visitors to can city can utilize the crime hotspots and chalk out a safe travel itinerary. A quick look at the heatmap in Figure 5(a) tells users that south-central and south-east DC has high crime density and they should use caution while travelling through those areas.

In addition to the crime hotspots, we provide **crime clusters** within the city. We use K-Means and DBScan as the clustering algorithms. For each clustering algorithm, we vary the parameters to compare the results. Both algorithms consider the spatial attributes (longitude and latitude) of the dataset and use Euclidean distance to measure the distance between two crime locations. Figure 6(a) shows 4 crime clusters generated by K-Means. Crime clusters help law enforcement agencies to decide patrol boundaries and plan patrol routes. CROWDSAFE allows police to further split the clusters to align with their patrolling needs.

We also provide **historical crime statistics** to the user in a **dashboard** interface (Figure 6(b)). A quick glance at the crime dashboard provides users with statistical information such as top crimes, worst areas within a city and yearly crime trends.



(a) Crime Clusters (b) Dashboard

Figure 6: Clusters & Dashboard

The dashboard is especially useful for members of the government to monitor the effectiveness of policy amendments, law enforcement agencies and officials. It is also useful to citizens who can get the crime statistics at their finger tips to make wise decisions. For example, a user looking at the dashboard in Figure 6(b) can quickly deduce that the 4th district in Washington DC is the safest and a potentially good place to have a residence.

These analytics alleviate much of the time consuming and manual tasks of knowledge discovery, thus promoting the effective and efficient use of crime data.

4. SUMMARY

CROWDSAFE is developed to meet the needs of the end user who wishes to participate in crowd sourcing of crime incidents. CROWDSAFE allows the user to search and report new crime information. The safety router feature helps the user schedule a safe and convenient travel plan. Through CROWDSAFE, we advocate the use of civic mobile applications to enable real time, location specific, and actionable information sharing. We envision CROWDSAFE to go beyond crowd sourced crime data and evolve into a collaboration platform between citizens and the government to improve the overall quality of life. We also foresee increased use of the Safety Router and crime analytics to aid public decision making.

A video demonstration can be viewed at this YouTube URL: http://www.youtube.com/view_play_list?p=53CFF52CCA8A6F6C

5. REFERENCES

[1] Jeff Howe, *Crowdsourcing: why the power of the crowd is driving the future of business*. Crown Business, 2008.

[2] Samuel Greengard, *Following the Crowd*. Communications of the ACM, February 2011: p. 20.

[3] Bertrand De Longueville et al, *"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.* In Proceedings of the 2009 International Workshop on Location based Social Networks. ACM, NY, USA, 2009

[4] *Washington DC Crime Data*. 2010; Available from: http://data.octo.dc.gov/

[5] Apache Solr. http://lucene.apache.org/solr/.