

Collaborative Filtering of Spatial-Temporal Information for Crisis Informatics

Jimmy Secretan
ad summos, inc.
215 Celebration Place
Suite 150
Celebration, FL 34747
Email: jimmy@adsummos.com

Abstract—In a disaster, accurate information is a resource that is often in short supply. The combination of rapidly unfolding events and the frequent loss of communication infrastructure including mobile phones, landlines, Internet and television broadcast make it difficult to gain situational awareness. And while there is obvious value in aggregating information for centralized emergency authorities, disasters like Hurricane Katrina have shown that authoritative organizations are not always prepared to mobilize quickly enough. There is significant potential for systems that support self-organization and information sharing among the victims of a disaster. This paper discusses an application designed to share disaster information over an ad-hoc network built from common mobile devices (e.g. smart phones and PDAs). A Collaborative Filtering (CF) algorithm aggregates the observations of numerous distributed users and disseminates maps of disaster related events back to them. This paper describes a framework for analysis of disaster events, including both the availability of necessary resources (e.g. water and fuel) as well as disruptive events (e.g. downed power lines, leaks, etc). In addition, a spatial-temporal clustering algorithm combines the observations of multiple users into coherent reports. The goals of the framework are to quickly spread and accurately update vital information while resisting rumors and errors. The results demonstrate that sharing information in this way can increase the group's situational awareness.

I. INTRODUCTION

When a disaster strikes an area, whether it is man-made (e.g. terrorism, toxic spills) or natural (e.g. hurricanes and floods), it often leaves the residents and organizations within the area in disarray. Disasters like Hurricane Katrina and the recent earthquake in Haiti demonstrate the difficulty for victims to find much needed information and resources. Traditional communications infrastructure such as cellular towers and landlines can be crippled by disasters. In these circumstances, ad-hoc Disaster Area Networks (DANs), composed of cell phones and other mobile devices, can provide robust emergency communications infrastructure.

While the communication technologies to assemble large area wireless networks are available, it is unclear whether the information shared can be leveraged by those who need it. Disasters frequently involve thousands or millions of people, each of whom is capable of contributing data to the system. Even if users of the system can gain access to the information from the DAN, the amount of data can be overwhelming. A user is much more likely to benefit from a few high level pieces

of important information than from thousands of observations. The data must be compressed and filtered to be manageable.

This paper introduces an algorithm for distributed aggregation of observations from a disaster area. The algorithm is designed to operate over an ad-hoc network of portable devices. Emphasis is placed on minimizing communication and data storage costs, which are primary concerns for low power devices and networks. The algorithm aims to ensure freshness of the aggregated observations by filtering out data that is out of date.

The remainder of the paper is laid out as follows. Section II describes two technologies upon which this research relies: ad-hoc networks for sharing information among mobile devices and systems which aggregate observations in emergency scenarios. Section III describes a model of both disaster scenarios and DAN communication. Section IV delineates the algorithm used for aggregation of observations based on disaster-oriented design goals. Section V details the simulation and associated parameters that demonstrate the operation of the algorithm. Results are presented in section VI and their implications are discussed in section VII. Finally, conclusions and future work are described in section VIII.

II. BACKGROUND

In disaster areas, the advantage of a mobile ad-hoc network is clear: Those affected by the disaster can continue to exchange information even though typical telecommunications infrastructure is unavailable. In several different applications, mobile ad-hoc networks (MANETs) have been proposed to address the needs of first responders [1]. Because of the availability of advanced mobile computing platforms (e.g. the iPhone and Android) as well as mature mobile ad-hoc protocols, it is now possible to connect mobile devices in a disaster area into an effective mesh network. The Serval Project [2] focuses on the development of software to assemble mesh networks of cell phones based on Wi-Fi and other unlicensed spectra. While the developers of the project recognize that the network could function as a vital channel for broadcasting emergency information, it is not clear how this information will be shared and how users will make sense of it.

The proliferation of participatory Internet media (e.g. Youtube, Twitter, etc.) suggests that the paradigm of one-way

information flow from authorities to citizens is incomplete [3]. Ushahidi is a crowdsourcing and collaborative filtering platform for aggregating tweets, blog posts and other user generated content for emergency reporting [4]. It was initially developed to map reports of election-related violence in Kenya. The Ushahidi platform now supports several different crowdsourcing applications to aggregate and visualize real time information submitted by users over the Internet. The related SwiftRiver platform [5], which came out of Ushahidi, focuses on employing natural language processing and data mining to aggregate tweets, e-mails and RSS feeds. While these platforms offer enormous potential for democratizing emergency management, they are dependent upon standard network infrastructure. To leverage some of the same techniques for coordination in a distributed ad-hoc environment is the focus of this paper.

III. MODELING A DISASTER AND ITS DENIZENS

To create a distributed algorithm for aggregated observations from disaster areas, they must first be modeled. In this paper, the disaster area is modeled as being populated with n agents who move around the area through Levy walks [6]. In many scenarios, human mobility is well described by these walks, consisting of mostly short movements with less frequent long movements. An agent who moves through these walks can help to spread information to distant groups. It is unknown whether or not these patterns hold in times of disaster: more research into disaster mobility patterns is needed.

Every disaster is a complex collection of emergencies, aid missions and agents trying to meet their basic needs and adapt to a changing environment. For instance, a hurricane can embody smaller disasters at many different scales. The aftermath can be as dangerous as the weather patterns themselves. A downed power line, a damaged building, a levy breach, or a blocked road can pose dangers that are difficult for authorities to track and warn others about. The arrival of aid is another unpredictable element of a disaster scenario. There is often a delay in mobilizing the necessary resources for the affected areas. Governments and NGOs that deploy that aid do not always know where it should be concentrated. Drawing again from the example of the hurricane, resources are of primary importance in the aftermath. Victims need to know when gas stations have supply and when they are out, when organizations are distributing food, ice and water, and when shelters, both official and impromptu are available. From the standpoint of the distributed algorithm, all of these occurrences are modeled as *events*.

An event E_i begins at a specific time t_i^S and ends at a specific time t_i^F . Each event is considered to be of a specific type C_j . This classification differentiates between events such as the availability of fuel at a particular station or a road way that is blocked off. Each event type has an associated average lifetime t_{C_j} . Ice may only be available for a couple of hours, while roadways can often be blocked for days.

An event can span a significant amount of time, and can be observed by many unique users in the system. When an agent

witnesses an event (i.e. passes within a distance d_O of the event) E_i , he records it on his mobile device with probability P_r . This reflects the fact that the system users will not always care to record their observations.

Working from this model of a disaster and the affected victims, it is possible to design a distributed algorithm that will suit the rapidly changing environment of a disaster. We develop the algorithm in the next section.

IV. SYSTEM DESIGN

In this section, a distributed system for emergency situational awareness is developed, based on the model from the previous section. First, the timing and location of an event will influence a user's decision to take some sort of action (e.g. the user would like to know if a road way is likely still blocked before trying to travel on it, or if resources are still available at a distribution point before going to it.) It is assumed that the users' mobile devices have both GPS receivers and clocks. Observations are tagged with spatial coordinates and a timestamp. The users' observations may be added through a graphical application, in free form text, or in a special markup language [7]. Time synchronization is non-trivial for mobile devices, especially if the cellular tower is down. Therefore, the system does not try to establish a shared absolute time frame; instead, times for events are recorded within the time of the observer's device. When the device transmits the observation, it transforms the observation's timestamp into a time difference from the transmission time. The receiving device then re-stamps the time of the observation to match its own time.

The devices also need a shared framework for understanding each other's observations. Each mobile device is identified by a unique ID, such as network MAC address. When a user makes an observation with the mobile device, the observation is assigned a universally unique ID (*UUID*) based on both the unique ID of the device and time of the observation. Uniquely identifying the observations allows them to be shared and aggregated among devices without double-counting them in aggregations.

Because a large number of users may enter pertinent observations into their devices, it may be impractical for users to comb through each individual report. Therefore, observations must be clustered into meaningful events. The first way to cluster observations is by geographic location. For instance, if there are several different reports of a downed power line, and the reports all come from locations within one or two blocks of each other, all of these events can be condensed into a single downed power line event, described somewhere within a one-block area. Another way to condense the collected observations is temporally. Disaster observations may only have a short time before they become stale. A gas station may only have gas available for several hours. If a user's device were to report information that is significantly out of date, the mistake would not only undermine the system, but could be costly for the user who followed incorrect information. Even worse, information that is not up to date could lead users into danger. To keep information relevant and to avoid the power

costs of transmitting unnecessary information, observations are removed from the user’s device once they reach a certain age.

Another important component for organizing observations and keeping them up to date is a set of event type codes and associated metadata. The event codes are a pre-determined set of descriptions for the observed events. Each code is associated with four pieces of information: average and standard deviation for both the size and duration of events like this one, based on previous study. This information helps the system to automatically filter out observations that are likely out of date.

The algorithm in figure 1 is executed by each individual mobile device. Starting with line 1, the algorithm iterates through \mathbf{O} , which is the set of all observations currently stored in the device. In lines 2–5, it discards observations whose timestamp (t_i) is less than the current time (t) by 3 standard deviations ($t_{C_j}^\sigma$) from the average ($t_{C_j}^-$) for comparable events. Expiring old observations from the cache helps to ensure data freshness and to minimize communication costs.

In lines 6–8, the device transmits the unique identifiers of its set of observations (\mathbf{UUID}) to another device within range d_T and the other device sends back its set. The device then finds which observations the other device has stored that it has not (line 9). In lines 10–12, the devices exchange full observation data, including the observation id, the event code, the latitude / longitude location, and the time of the observation.

When new data is received from another device, all observations including both existing ones and new ones are reclustered to generate events that are meaningful to the user (lines 14–24). The clustering algorithm employed is an inexpensive one inspired by the Probabilistic Neural Network (PNN) [8]. The PNN employs Parzen window estimation [9] for generating non-parametric probability density functions based on observations.

The clustering algorithm iterates through all observations remaining on the device and generates clusters of observations called *interpreted events* \mathbf{E}_i . A match value M_i is calculated between each cluster \mathbf{E}_i and an observation O_i by summing the negative squared distance between O_i and each observation O_j that is part of \mathbf{E}_i . The match value is then normalized by the size of the cluster. The distance function dis is defined as the 3-D Euclidean distance between the normalized spatial and temporal coordinates of both observations. Normalization parameters must be chosen to bring all components of the distance to ≤ 1 .

Finally, the observation O_i is assigned to the cluster \mathbf{E}_i that has the lowest match value M_i , as long as the match value is less than a minimum cutoff value (O_c). If not, a new cluster (\mathbf{E}_{i+1}) is created and the observation O_i is assigned to it.

V. SIMULATIONS

To evaluate the performance of the ad-hoc event clustering algorithm, a simulation was developed based on MASON [10], a popular tool for simulating multi-agent interaction. The simulation abstracted away the MAC layer and specific low-level network protocols; instead, it assumed that the ad-hoc network transfers data losslessly between users when they are

```

1 foreach  $O_i \in \mathbf{O}$  do
2   if  $t_i < (t - t_{C_j}^- - 3t_{C_j}^\sigma)$  then
3      $\mathbf{O} \leftarrow \mathbf{O} / \{O_i\}$ 
4   end
5 end
6 foreach neighbor  $N_i$  in transmission distance  $d_T$  do
7   send ( $N_i, \mathbf{UUID}$ );
8    $\mathbf{UUID}_r \leftarrow$  receive ( $N_i$ );
9    $\mathbf{O}_s \leftarrow O_i \forall i \in (\mathbf{UUID} \setminus \mathbf{UUID}_r)$ ;
10  send ( $N_i, \mathbf{O}_s$ );
11   $\mathbf{O}_r \leftarrow$  receive ( $N_i$ );
12   $\mathbf{O} \leftarrow \mathbf{O} \cup \mathbf{O}_r$ ;
13 end
14 foreach  $O_i \in \mathbf{O}$  do
15   foreach  $\mathbf{E}_i$  do
16      $M_i = (\sum_{O_j \in \mathbf{E}_i} \exp(-dis(O_i, O_j)^2)) / |\mathbf{E}_i|$ ;
17   end
18   if  $\min M_i < O_c$  then
19      $s \leftarrow \text{argmin}_i M_i$ ;
20      $\mathbf{E}_s \leftarrow \mathbf{E}_s \cup O_i$ ;
21   else
22      $\mathbf{E}_{i+1} \leftarrow \{O_i\}$ ;
23   end
24 end

```

Fig. 1. Algorithm for observation sharing and clustering.

Parameter	Value
<i>Max transmit distance</i>	20m
<i>Observation distance</i>	20m
<i>P of reporting event</i>	0.5
<i>Simulation area</i>	400mx400m
<i>Number of agents</i>	150
<i>Range of average event duration</i>	2–24hrs
<i>Range of average event size</i>	10–100m

Fig. 2. Simulation parameters.

within range d_T . When they are outside of that range, there is no transmission. The simulation keeps track of each agent’s *awareness*, defined in this case to be the number of currently occurring events of which the agent is aware divided by the total number of currently occurring events.

The parameters used in the simulation are given in figure 2. Five different simulations were run for 120 timesteps each after the occurrence of the first event.

VI. RESULTS

The table in figure 3 shows, averaged over 5 simulation runs, the average agent awareness, the average data sent and received and the average number of observations stored by the end of the run. The data transmitted and received, as well as the size of the observations stored remained reasonable throughout the operation of the system. These modest requirements support the suitability of this algorithm for use on mo-

Metric	Value
Average event awareness ratio	0.416
Average data sent and received (bytes)	3393
Average observations stored	8.48

Fig. 3. Average event awareness, false positives, data transfer (sent and received) and number of observations stored averaged over 5 runs.

mobile devices. The system also provides situational awareness to users who may not directly observe events occurring, with an average awareness ratio of 0.416.

Figure 4, taken from a simulation execution, demonstrates how awareness gradually spreads throughout the network as new events occur.

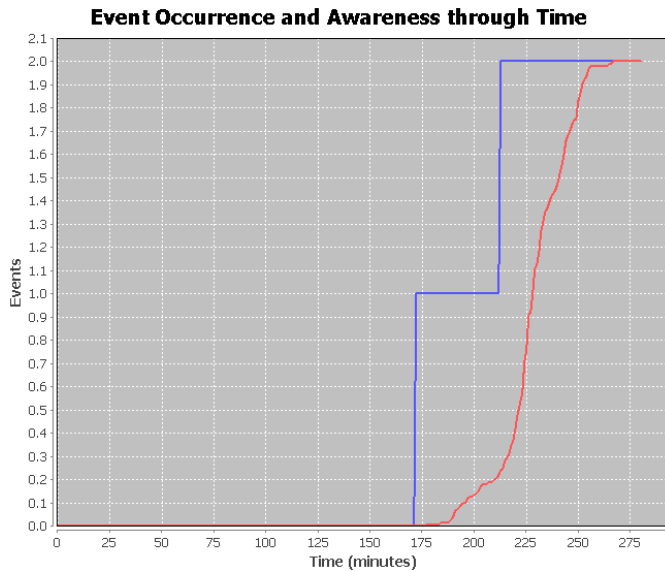


Fig. 4. How awareness follows event occurrences. The top line shows events as they begin, and the bottom line shows the average number of events of which the agents are aware.

VII. DISCUSSION

While the simulations presented in this paper demonstrate the potential effectiveness of such a system, there are several caveats. First, there is little detailed data describing the exact times and locations of events during an actual disaster. The same confusion and lack of coordination that this system aims to prevent is precisely what makes it difficult to faithfully record events during a disaster. It is unclear whether the assumptions employed in the design of this system truly hold in a disaster scenario. Furthermore, it will not be clear without significant additional study how users would interact with the system. It is possible that users of the system will be less likely to exercise caution if they feel that the system will alert them of any dangers, and it could cause users to be surprised by something the system has missed. It is likely that several events, including the availability of resources, will depend upon the number of users aware of those resources. Cognizant

of that fact, it is unclear if users will even be willing to share information. However, the potential benefits of such a system make finding answers to these questions worth while.

VIII. CONCLUSION AND FUTURE WORK

This paper described an algorithm for exchanging and clustering observations about disaster-related events through an ad-hoc wireless network of mobile devices.

Crisis informatics is still a nascent field. Little is understood about how information can be most effectively shared among disaster victims. To begin with, there is little research on movement models of victims of disaster. Until recently, information like this was difficult to obtain; now the ubiquity of cell phones can transform a tower still standing in a disaster area into a valuable source of mobility data. Sources of data like these will contribute significantly to the field.

In addition to conducting more extensive simulations across a number of parameters, there are numerous potential enhancements that can be made to this work. First, the quality of information shared in these systems can be improved through a trust mechanism: Observations coming from friends on your contact list, for instance, should carry more weight than those from strangers. Furthermore, there is significant potential for employing natural language technologies and text mining to make data entry easier. Although a great deal of work remains to make these technologies suitable for deployment in highly stressful and unpredictable disaster scenarios, the potential savings of lives and property make solutions to these challenges worth pursuing.

REFERENCES

- [1] J. B. Kopena, E. A. Sultanik, R. N. Lass, D. N. Nguyen, C. J. Dugan, P. J. Modi, and W. C. Regli, "Distributed coordination of first responders," *IEEE Internet Computing*, vol. 12, pp. 45–47, 2008.
- [2] "The Serval Project," <http://www.servalproject.org>, 2010.
- [3] L. Palen, K. M. Anderson, G. Mark, J. Martin, D. Sicker, M. Palmer, and D. Grunwald, "A vision for technology-mediated support for public participation & assistance in mass emergencies & disasters," in *ACM-BCS '10: Proceedings of the 2010 ACM-BCS Visions of Computer Science Conference*. Swinton, UK, UK: British Computer Society, 2010, pp. 1–12.
- [4] "Ushahidi :: Open source crowdsourcing tools (FOSS)," <http://www.ushahidi.com/>, 2010.
- [5] "SwiftRiver — verifying and filtering news (FOSS)," <http://swift.ushahidi.com/>, 2010.
- [6] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong, "On the levy-walk nature of human mobility," in *INFOCOM*, Arizona, USA, 2008.
- [7] "Project EPIC: Helping Haiti Tweak the Tweet," <http://epic.cs.colorado.edu/tweak-the-tweet/>, 2010.
- [8] D. F. Specht, "Probabilistic neural networks," *Neural Networks*, vol. 3, pp. 109–118, 1990.
- [9] E. Parzen, "On estimation of probability density function and mode," *Annals of Mathematical Statistics*, vol. 33, pp. 1065–1073, 1962.
- [10] S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan, "MASON: A Multi-Agent Simulation Environment," *Simulation: Transactions of the society for Modeling and Simulation International*, vol. 82, no. 7, pp. 517–527, 2005.