中央研究院
資訊科學研究所
Institute of Information Science, Academia Sinica • Taipei, Taiwan, ROC

TR-IIS-13-002

# A Framework for Fusion of Human Sensor and Physical Sensor Data

P. H. Tsai, Y.-C. Lin, Y. Z. Ou, E. T.-H. Chu, and J. W. S. Liu

July 4, 2013 || Technical Report No. TR-IIS-13-002
http://www.iis.sinica.edu.tw/page/library/TechReport/tr2013/tr13.html

Institute of Information Science, Academia Sinica
Technical Report TR-IIS-13-002

# A Framework for Fusion of Human Sensor and Physical Sensor Data

P. H. Tsai, Y.-C. Lin, Y. Z. Ou, E. T.-H. Chu, and J. W. S. Liu,

**ABSTRACT**

Many disaster warning and response systems can improve their surveillance coverage of the threatened area by supplementing in-situ and remote sensor data with crowdsourced human sensor data captured and sent by people in the area. This revised version of a 2012 technical report also presents fusion methods which enable a crowdsourcing enhanced system to use human sensor data and physical sensor data synergistically to improve its sensor coverage and the quality of its decisions. The methods are built on results of classical statistical detection and estimation theory and use value fusion and decision fusion of human sensor data and physical sensor data in a coherent way. They are building blocks of a central fusion unit in a crowdsourcing support system for disaster surveillance. In addition, this version contains a brief description of CROSS, a crowdsourcing support platform that can be used to enhance existing disaster surveillance systems, as well as performance data on relative merits of the detection method proposed here.

**Keywords:** Crowdsourcing, Multiple sensor fusion, Statistical detection and estimation.



---

P. H. Tsai and Y.C. Lin are affiliated with Department of Computer Science and Information Engineering , National Cheng Kung University, Taiwan. Their email addresses are phtsai@mail.ncku.edu.tw and jim3232ya@hotmail.com

E. T. H. Chou is affiliated with Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology. His email address is edwardchu@yuntech.edu.tw.

Y. Z. Ou and J. W. S. Liu are affiliated with Institute of Information Science and Center for Information Technology Innovation, Academia Sinica, Taiwan. Their email addresses are {yzou, janeliu}@iis.sinica.edu.tw.

## A Framework for Fusion of Human Sensor and Physical Sensor Data

P. H. Tsai, Y.-C. Lin, Y. Z. Ou, E. T.-H. Chu, and J. W. S. Liu

## 1 INTRODUCTION

Experiences from recent major disasters have told us that in-situ and remote *physical sensors* deployed by disaster surveillance and early warning systems often cannot provide adequate data for situation assessment purposes. When this happens, crowdsourcing human sensor data can be an effective solution. By a *human sensor*, we mean a person armed with one or more smart mobile devices and social networking services. By *human sensor data*, we mean observation (and measurement) data captured and contributed by human sensors. Today's smart mobile devices equipped with cameras, temperature and vibration sensors, etc. and social networking services have made a wide spectrum of mobile applications and services more pervasive, location-aware and context-aware than feasible even a few short years ago [1-4]. Using them, increasingly more people are able to participate and contribute to diverse crowdsourced sensing systems and applications (e.g., [1, 5-8]) for purposes such as the generation of fine-grain maps of weather radar, noise level, air quality, snow depth, radiation level, traffic and road conditions, litters in parks, and so on.

The platform CROSS (a CROwdsouring Support system for disaster Surveillance) [9] was built to support the exploitation of human sensor data and physical sensor data synergistically for disaster surveillance and decision support purposes. When physical sensor coverage is inadequate, the system starts a *crowdsourcing data collection* (*CDC*) process by broadcasting a call for participation to a crowd of human sensors. During the process, participating human sensors make observation(s) at and around locations as requested by the system and send the data thus captured back to the system. The process completes when the system has acquired sufficient data to give it situation awareness and support its decisions and operations.

The interactions and collaborations between the system and participants can be either crowd-driven or system-driven [10]. When the system uses a *crowd-driven strategy*, it does nothing other than collecting and processing reports from participants, relying solely on mobility and interactions of individual participants for coverage of the threatened area. Working in this way, the system is similar to many crowdsourced sensing systems and applications mentioned above.

For the purpose of collecting data to supplement physical sensor data prior to or during an emergency, the crowd-driven approach is not ideal, however: Oftentimes, the system should use as few participants as needed for each CDC process for many reasons including availability and costs of qualified participants. Without well-planned routes for participants to follow during a process, some locations may be visited by more participants than necessary while other locations are visited by too few. Consequently, the *response time* (i.e., the length of time from the start to the end of the process) of the process may be prolonged. In cases of emergencies such as wildfires and floods, the system also needs to help participants stay away from dangerous locations.

*System-driven strategies* were motivated by these considerations. In this case, the system provides each participant with an exploration path for him/her to follow during the current process and issues directives as needed to alter the path. In addition to capabilities needed to communicate and interact with the participants prior and during CDC processes, CROSS also provides the emergency manager with tools for selecting participants from human sensors who responded to its call for participation based on the cost and benefit associated with each of them and planning for each selected participant a path. Section 3 will present an overview of the CROSS prototype and its approaches to participant selection and path planning [9-11].

This paper focuses on the data fusion and processing methods which a system like CROSS can use to determine whether a CDC process needs to be launched and during a CDC process whether sufficient human sensor data has been collected and hence the process can be terminated. The underlying problem that the system must solve is how to use human sensor data and physical sensor data synergistically to improve sensor coverage and the quality of the decision based on sensor data. We call this problem the *symbiotic data fusion and processing* (*SDFP*) problem. We focus primarily on statistical detection formulations and solutions [12-17] of the problem, rather than estimations of parameters that define the state of the threatened area. A reason is that for many likely scenarios, the system aims to detect based on sensor data taken within the threatened area the occurrences of events and conditions that warrant response actions. Examples of such conditions include “tag balls on beach” and “smoke visible” on a stretch of beach and a campsite threatened by oil spill and wildfire, respectively. Appropriate actions in response to the detection of such conditions include launching cleanup operation, closing the campsite, evacuating campers, and so on.

The work described here makes two contributions: First, our work is among the first, if not the first, to characterize and treat data from both physical sensors and human sensors used for surveillance and monitoring purposes in a coherent way. Our abstract, yet realistic model of systems containing physical in-situ sensors and mobile human sensors captures all the characteristics of the sensors that are relevant to how to process the data from them. It enables us to build solutions of the problems in fusing physical and human sensor data on the rigorous foundation of classical stochastic detection and estimation theory. For many real-life scenarios, the solutions provide the system with not only quantitative assessment of its decision quality but also control over tradeoffs between conflicting quality criteria.

The second contribution is the design of a fusion unit for processing data collected from physical and human sensors and determining automatically and dynamically when to initiate and when to terminate CDC processes depending on whether the sensor coverage is sufficiently good. The design of this important component is built on the solutions presented here. We will make the fusion unit a part of CROSS and thus make CROSS a full-function crowdsourcing support platform suitable for enhancing many types of disaster surveillance systems.

Following this introduction, Section 2 discusses related work. Section 3 first describes CROSS briefly and presents three representative scenarios to provide background and rationales for our discussions here. It then presents our assumptions on the surveillance system and its physical sensors and models of disaster threatened areas, participants of the CDC process, and physical and human sensors. Section 4 presents the design and implementation of a central fusion unit for processing sensor data, making decisions, and helping the system manage CDC processes. The section also presents the rationales for the structure of the fusion unit, its reliance on the well-known *Neyman-Pearson criterion* [17] (commonly called *the N-P test* in literatures), and this choice over other binary hypothesis testing techniques. Section 5 presents statistical detection and estimation formulations of the SDFP problem and the N-P test solution of the binary hypothesis testing variant of the problem. Section 6 discusses the performance of the solution. Section 7 summarizes the paper and discusses future work.

## 2 RELATED WORK

In recent years, platforms such as Sahana and Ushahidi [18, 19] have been used worldwide to support crowdsourcing the collection and dissemination of crisis management information during and after major disasters. In contrast, modern disaster surveillance (and early warning) systems typically do not incorporate crowdsourcing social reports as an integral part of their standard operating procedures. As stated earlier, a disaster surveillance systems needs to make critically important decisions: A decision such as to start oil cleanup operations, close a popular campsite, evacuate residents, and so on, carries the risk of crying wolf, hopefully with an acceptably small probability, but a failure to call for action when action is warranted may cause costly damages and loss of lives. Except for disasters that take days and months to develop, the system must be able to acquire situation awareness and make a decision within hours, even minutes. If the system is crowdsourcing enhanced, it must be able to process social reports in real-time automatically using relatively efficient rules and extract from the reports decision support information of sufficiently good and quantifiable quality. The solutions presented in later sections aim to meet these needs.

Our work was motivated by the fact that existing techniques and tools for processing social reports cannot meet these needs. Problems in processing social reports have been a focal point of intense efforts by many research communities. Prior work in this area targeted primarily crowdsourcing applications such as the ones mentioned above [1, 5-8, 18, 19]. A challenge shared by these diverse applications arises from their use of social reports contributed by mass, and often unknown, crowds. In addition to the problems in discovering, extracting, and refining information from a vast number of reports, these applications must also deal with unbounded uncertainty in the information extracted from the reports, hence, the problem of verifying veracity and assessing accuracy of the information. Numerous techniques and tools (e.g., [20-33]) based on a wide range of technologies, including machine learning, fuzzy systems, data mining, information retrieval and natural language processing, are now available. In addition, some heuristic combinations of technologies and semiautomatic processing tools (e.g., [28, 35]) were developed by open source software community for processing eyewitness reports and request messages from victims, responders and general public during the crises in days after 2010 Haiti and Chile earthquakes. By and large, common shortcomings of existing solutions include that they take manual efforts and processing time too high to be acceptable for our application and that they cannot provide the system with quantitative quality measures of the extracted information.

Unlike previous efforts, our work makes two restrictive but realistic assumptions: First and foremost is that disaster surveillance and warning systems use only known participants, selected from registered volunteers whom have been recruited as a part of community-, region- and country-wide disaster preparedness efforts. In addition to having mobile devices and applications capable of communicating with the crowdsourcing support system, the volunteers have promised to never lie, make observations independently, and report observed data as requested. Second, as a part of the volunteer registration process, the system acquires the statistical characteristics of noises (and hence errors) in human sensor data, at least bounds to uncertainties in the data contributed by them. This is analogues to the fact that the system must have statistical distributions of noises in physical sensor data in order to process the data. The abstract model of symbiotic sensors in surveillance systems presented in Section 3 is based on these assumptions. As we stated earlier, the model characterizes data from human sensors and in-situ physical sensors in a consistent way and thus enables us to build a rigorous and efficient solutions needed by the fusion unit using techniques in classical stochastic detection and estimation in general and distributed multiple sensor fusion in particular [12-17].

In terms of objective, our problem resembles the problems of improving the coverage of physical sensors such as the one studied by Xing, *et al*. [12]: Like them, our system also wants to improve its coverage, except that our system uses human sensor data to reduce the limitations of physical sensor coverage. In subsequent discussions, we will use the term $(\alpha, \beta)$-coverage as in [12] to state the desired quality of sensor coverage precisely: In a process of trying to detect a specified condition of a disaster threatened area based on sensor data taken in the area, the system is said to have made a *false alarm* when it declares that the condition is true when the condition is in fact not true. It is said to have detected the condition when it declares the condition to be true when the condition is indeed true. We use $F$ and $D$ to denote the *false alarm probability* (i.e., the probability that the system makes a false alarm) and *detection probability* (i.e., the probability that the system succeeds in detecting the condition), respectively. For given *threshold false alarm probability* $\alpha$ ($0 \leq \alpha < 0.5$) and *threshold detection probability* $\beta$ ($0.5 < \beta \leq 1$), the system is said to have achieved $(\alpha, \beta)$-*coverage* when its false alarm probability $F$ is at most equal to $\alpha$ and its detection probability $D$ is at least equal to $\beta$. Sensor coverage is said to be *sufficiently good* when the system can achieve $(\alpha, \beta)$-coverage of the threatened area by processing available data in some way(s).

The thresholds $\alpha$ and $\beta$ are application specific. Without loss of generality, we assume here that they are chosen by the emergency manager based on the costs and benefits of taking or not taking the response action associated with the detected condition: The manager wants the chance of taking the action unnecessarily to be at most equal to $\alpha$ and the chance of failing to take the action indicated by the condition to be at most equal to $(1 - \beta)$. For our application, $\alpha$ and $(1 - \beta)$ are typically in the range from 0.1 to 0.01, while for typically sensor data processing and statistical communication applications such as the ones described in [12-16], they are typically in orders of $10^{-6}$, $10^{-7}$ and even smaller.

Our approach resembles the one taken by Wang, *et al*. [34, 35] who are among the first to apply statistical estimation and hypothesis testing techniques to processing social sensor data in order to discover and assess the truth carried by the data. Unlike our model, their models do not capture the symbiotic nature of sensors used by crowdsourcing enhanced disaster surveillance systems: Wang, *et al.* demonstrated via a case study that the EM (Expectation Maximization) algorithm [36] can out-perform the Bayesian interpretation scheme and Truth Finder [33, 35] for fusing binary-valued observations. For our application, a shortcoming of these schemes is that they do not give the fusion unit control over tradeoffs between quality measures, such as false alarm and detection probabilities, that cannot be optimized at the same time. This is why we treat the SDFP problem as a detection problem and use the N-P test whenever appropriate. The test is not only optimal when *a priori* probabilities are unknown but also simpler to implement than the EM algorithm. Moreover, as performance data presented in Section 6 will show, the performance of EM algorithm is at best comparable to that of the N-P-test.

Finally, we note that recent studies (e.g., [3, 7-9]) on issues in integrating social sensing with pervasive and ubiquitous computing also consider fusion of data from mobile and ubiquitous physical sensors with data from social sensors. CROSS can be thought of as an application-specific infrastructure for integration of physical and human surveillance sensors. A difference between the SDFP problem and their fusion problems is that data provided to CROSS by physical and human sensors are of the same types, whereas they provide context-aware pervasive services and applications with data of complementary types.

## 3 BACKGROUND, ASSUMPTIONS AND MODELS

This section first presents an overview of CROSS and describes three representative scenarios in order to provide background and support the assumptions, rationales, models and problem formulations presented in later sections. It then presents the models of physical and human sensors and introduces the terms and notations used in later sections.

### 3.1 System-Driven Crowdsourcing

Again, CROSS is a system of tools designed to support system-driven crowdsourcing human sensor data collection by disaster surveillance and early warning systems. According to this strategy, the system directs participants of the current CDC process to locations where human sensor data are needed along exploration paths planned by the system for them. It is convenient to think that there is a virtual sensor at each of these locations: During a CDC process, each *virtual sensor* provides the system with human sensor data taken and sent by human sensors in a neighborhood of a specified size around the virtual sensor. Figure 2(a) shows the key components of CROSS needed to support this strategy. Currently, the central fusion unit, shown as a dashed rectangle, is not yet available. Without it, CDC processes are started and terminated manually, and data from human sensors are displayed at the command center, where CROSS executes, for the consumption of the emergency manager(s).

The system requires each registered volunteer to be equipped with a smart phone that uses GPS or other means to locate itself. It has an “I am here” application for sending location information and geotagging short messages in conformance to a standard (e.g., Open GeoSMS standard [37]) and a web application built on Google maps API for displaying the exploration path planned by the system for him/her if and when the volunteer is selected to participate. As the last step of registration, the volunteer is asked to join a volunteer group in Facebook and is given by CROSS an account on the system.

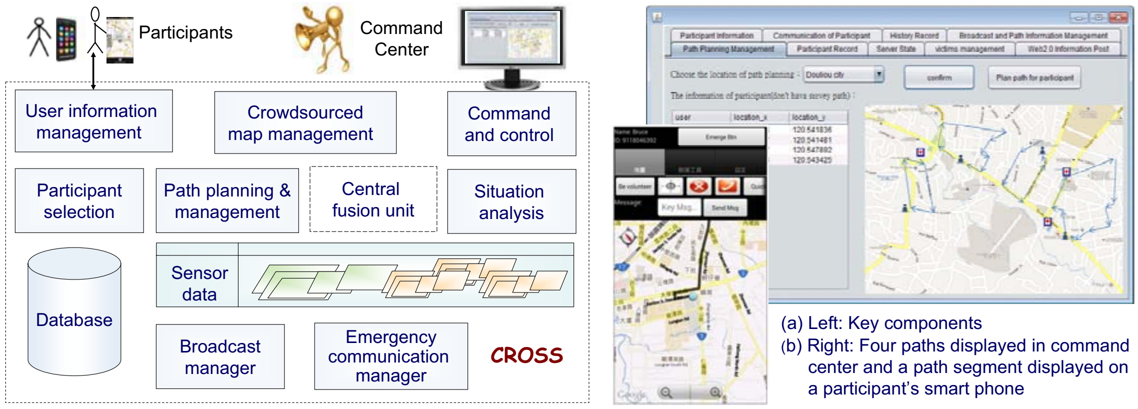



Figure 1 CROSS components

To start a CDC process, the emergency manager uses the broadcast manager to post on Facebook a call for participation, containing as a volunteer requirement the locations needing visits. A short message containing the requirement pops up on registered volunteers' smart phones. If after viewing the message, a volunteer decides to participate, he or she signs in CROSS to report his/her identity and current location. Having thus volunteered, he/she waits for the path planned by the system for him/her or a message stating that his/her help is not needed at the current time. The broadcast manager then passes the locations of all the signed in volunteers to the participant selection and path planning modules, as well as the crowdsourcing map manager to have the locations displayed.

For the purposes of selecting participants of a CDC process, the threatened area is divided into $R$ ($\geq$ 1) regions, each of which has a value (> 0). Associated with each responded volunteer (say the $j$-th volunteer) are benefit $b_{j,k}$ ($\geq$ 0) and cost $c_{j,k}$ ($\geq$ 0) if he/she is selected and assigned to visit virtual sensors in the $k$-th region, for $k$ = 1, 2, … $R$. The variant of participant selection problem currently in use can be stated in general as follows: Given as input parameters, $R$ regions, each of which is defined by its value and locations of virtual sensors in it; $K$ responded volunteers, each of whom is defined by $R$ benefit and cost factors; and a total budget available to spend on all selected participants. The objective is to select participants from volunteers and assign them each to a region in such a way that the total benefit achieved by all selected participants is maximized subject to the constraints that for every region, the total benefit of all participants assigned to the region is no greater than the value of the region and the total costs of all selected participants does not exceed the total budget. This problem is an extended special case of generalized assignment problem [38], which is known to be NP-hard. CROSS solves the problem heuristically, using efficient algorithms such as the one described in [11].

We note that the system can choose to make values of regions and benefits/costs of participants represent different things. In particular, the system can make the benefit of a participant assigned to a region equal to the number of virtual sensors in the region and the value of the region equal to the total number of virtual sensors in it times the number of human sensors required to visit each virtual sensor. The data on the quality of sensor coverage as a function of the number of human sensors per virtual sensor presented in Section 6 will enable this choice of input parameters. Similarly, the cost $c_{j,k}$ ($\geq 0$) can be set based on the travel time from the initial position of participant $j$ to the $k$-th region, or simply the amount of reward to be given to the participant for his/her help.

After participants are chosen and assigned, the path planning module computes for all the participants assigned to each region a set of exploration paths, a path for each participant. The constraints are that the path of each participant starts and ends at his/her initial location and visit each virtual sensor in the region at most once. For this computation, the path planning module characterizes the region by a directed graph in which nodes represent virtual sensors and lengths of edges represent minimum travel times between virtual sensors. Given the graph and initial locations of participants, path planning module computes all the paths satisfying the above mentioned constraints. The objective is either to maximize the visited virtual sensors by all participants within a specified length of time or to have every virtual sensors visited by a specified number of participants in minimum time. The problems are new variants of the well-known multiple traveling salesman problem [39]. The formal definition of the problem and the heuristic algorithm currently used by the module can be found in [9].

From time to time, the system needs to redirect some participants. This is done by providing them with new paths. For example, Figure 2(b) shows four paths displayed on a monitor within the command center. The figure also shows a segment of a participant's path displayed on his/her smart phone. In addition to providing the command center whereabouts and progresses of all participants, the crowdsourced map manager integrates the messages received from them and displays them on a map. Participants may find blocked road, building fires, falling trees and victims along their routes. They may also be trapped by incidents during their exploration. CROSS provides them with an emergency communication function to report the incidents and communicates with participants nearby.

### 3.2 Representative Scenarios

Figure 2 shows three representative scenarios [10, 11]. Only part (a) shows physical sensors: They are surveillance cameras in this scenario. Physical sensors in other parts are omitted in order to keep the figure simple. Small circles in the figure labeled by symbols $S_1$, $S_2$, … $S_k$ and so on represent virtual sensors. The exact locations of the sensors depend on the type of disaster, the condition of interest, and physical sensors used to monitor the condition, and so on. In all scenarios considered here, data from all sensors, physical and virtual, within a disaster threatened area are used together to detect a specified condition. The fusion unit assumes that all sensors are well placed but does not use their locations as an input.

(a) Human sensor data: presence/absence of tar balls

(b) Human sensor data: {wind direction & speed, temperature, humidity}

(c) Human sensor data: water level in centimeters

Figure 2 Representative scenarios

In the oil spill scenario, the area threatened by an oil spill is the stretch of beach delimited by two high resolution surveillance cameras on the beach. Periodically, each camera scans and captures images of the beach. The bandwidth required to send captured images to the system being prohibitive, the camera processes the images locally to decide whether the images indicate the presence of tar balls and sends its local decision to the surveillance system. Crowdsourcing human sensor data is used when visibility hampers the coverage of the cameras and degrades the quality of their decisions. Similar to the litters-in-park case study presented in [35, 36], each human sensor is asked to report a binary value indicating the presence or absence of tar balls at his/her location. Based on values in their reports and local decisions of the cameras, the system decides whether the stretch of the beach monitored by the sensors is threatened by oil spill and preventive clean up operations should be launched. Using the terms from [15, 16], we say that the system does *decision fusion* (i.e., working with local decisions of individual sensors) for physical sensors, but does *value fusion* (i.e., working with values of sensor data) for all human sensors or human sensors assigned to each virtual sensor.

The park scenario in Figure 2(b) assumes that physical sensors in the few weather stations scattered in a national park routinely measure local wind speed and direction, temperature and humidity. When a raging wildfire is threatening the park, the surveillance system asks human sensors camping in the park to report values of these same types of data observed by them at their locations. In the scenario shown in Figure 2(c), human sensors along several city blocks threatened by flood during an unexpectedly heavy downpour may be asked to report the depth of water on roadway(s) in front of them. In these cases, human sensor data have arbitrary values, similar to physical sensors. The system can, in principle, do value fusion, making its determination of whether the campsite is in danger of the wildfire or the streets will soon become impassable, and hence should be closed immediately, based on the data.

Value fusion of arbitrary valued human sensor data can be problematic, however. We will discuss technical factors that complicate the fusion process in Section 5. A practical problem is that human sensors may not have tools to measure wind direction, water level, etc. and must rely on their own estimations. Consequently, considerable effort is required to collect and validate the statistical characteristics of noise in arbitrary valued human sensor data. So, in practice, the system is likely to request from human sensors binary valued reports such as whether “wind in the direction of fire” and “water level rising” is true or false. In essence, the system is collecting from human sensors binary decisions based on their individual observation on wind direction and water level. Rather than fusing such binary valued human sensor data with arbitrary valued physical sensor data, the fusion unit may first process data from each physical sensor to make a local binary decision regarding whether wind is in direction of fire or water level is rising and then fuse decisions of physical sensors with the human sensor data. This is the design choice of the simple and general centralized fusion unit described in the next section, along with tradeoffs of the design.

### 3.3 Models of Physical and Human Sensors

Our models of the symbiotic sensors (i.e., physical sensors and human sensors) in a disaster surveillance system make the five types of assumptions. First, the command center uses data from all sensors to monitor and detect alarming phenomena or conditions that may occur in the disaster threatened area and support its decisions on preparedness and response actions upon the detection of the conditions. For sake of discussion in this paper, it suffices for us to focus on sensors and sensor data processing for the monitoring and detection of a single condition of interest, and we indeed do so throughout the paper.

Second, all the physical sensors used to monitor and detect of the condition of interest are *functionally identical:* In general, each of them provides the system with the same $n$ (≥1) types of data. For example, in the scenario in Figure 2(b), the physical sensor in every weather station provides 4 types of data: wind speed, wind direction, temperature and humidity. Sensors are read at discrete instants of time. Without loss of generality, we assume that physical sensors are read immediately before each CDC process starts. Because of ever-presence of random noise, the values obtained by reading any physical sensor $S_i$ is a *sample* $\boldsymbol{x_i} = (x_{i,1}, x_{i,2}, \dots, x_{i,n})$ of a $n$-dimensional random variable $\boldsymbol{X_i} = (X_{i,1}, X_{i,2}, \dots, X_{i,n})$.

Third, the surveillance system aims to provide the command center with a comprehensive view of a disaster threatened area. Ideally, the system would have a sufficient number of physical sensors of the right kind and at the right locations to achieve a sufficient good coverage of the area. For reasons including deployment costs, damages and poor operating conditions, $v$ physical sensors are missing. When existing physical sensors cannot provide the command center with sufficiently good and complete coverage, it starts a CDC process in order to acquire human sensor data on conditions around the location of each missing physical sensor. We assume that the system (hence the command center) knows the identity and location of each missing sensor and has a *virtual sensor* at the location of the sensor. We use $S_1, S_2, \dots, S_v$ to denote the virtual sensors, and denote existing physical sensors by $S_{v+1}, S_{v+2}, \dots, S_\sigma$ when the threatened area should be covered by a total of $\sigma$ sensors.

The command center may solicit from human sensors data of the same types as that of the physical sensors. In this case, we denote the sample (i.e., data) reported by the $k$-th human sensor (participant) from locations around the virtual sensor $S_i$ during the current CDC process by the vector $\boldsymbol{x}_i(k) = (x_{i,1}(k), x_{i,2}(k), \dots, x_{i,n}(k))$, for $i = 1, 2, \dots, v$. The fusion unit computes for each virtual sensor $S_i$ a sample $\boldsymbol{x}_i$ for the virtual sensor based on samples $\boldsymbol{x}_i(k)$'s from all human sensors assigned to $S_i$ and statistical distributions of $\boldsymbol{x}_i(k)$'s. Figure 3 shows an abstract view of such a system of symbiotic sensors as seen by the command center and fusion unit.

Fourth, the sample $\boldsymbol{x}_i$ from every physical sensor $S_i$, for $i = v+1, v+2, \dots, \sigma$, is the sum of a vector of noise-free values (i.e., true values of sensor readings) plus a vector of noise components, the latter being a sample of random noise $\boldsymbol{\Theta}_i = (\Theta_{i,1}, \Theta_{i,2},\ldots, \Theta_{i,n})$ of the sensor. The noise $\Theta_{i,j}$ in the $j$-th component $x_{i,j}$ of $\boldsymbol{x}_i$ of all physical sensors (i.e., for all $i = \nu+1, \nu+2, \ldots, \sigma$) are assumed to be statistically independent, identically distributed. Consequently, the random variables $\boldsymbol{X_i}$ for all $i = \nu+1, \nu+2, \ldots, \sigma$ are statistically independent. This assumption is valid for functionally identical sensors that operate independently from either other under similar operating conditions. For every physical sensor $S_i$, the noise components $\Theta_{i,j}$ and $\Theta_{i,k}$ of $\boldsymbol{\Theta}_i$, and hence the component random variables $X_{i,j}$ and $X_{i,k}$ in $\boldsymbol{X_i}$, for $j, k = 1, 2, \ldots n$ and $j \neq k$, are also assumed to be statistically independent. This simplifying assumption is valid when the sensor does not have any systematic problem (e.g., power fluctuation) that introduces correlated noises to component values of its sample. It follows from these assumptions that the noise $\boldsymbol{\Theta}_i$ in samples of all physical sensors can be characterized by distribution functions $A_j(t)$ of $\Theta_{i,j}$ (i.e., the probability of $\Theta_{i,j} \leq t$) for $j = 1, 2, \ldots n,$ where $t$ is from a scenario-specific set of values. The fusion unit knows these distribution functions.

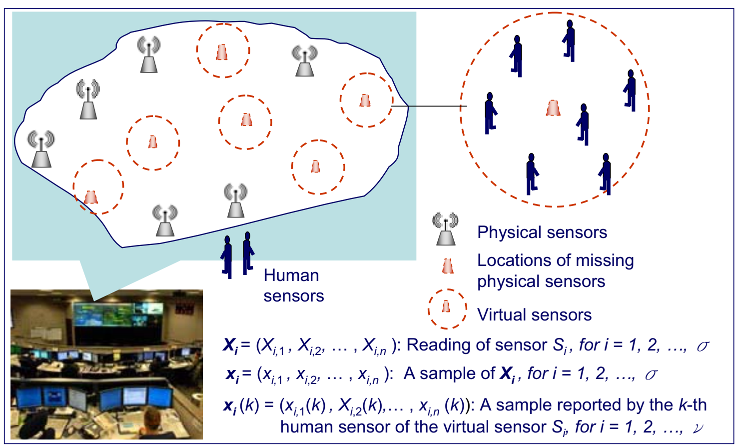



Figure 3 An abstract view of symbiotic sensors

The fifth and final type of assumption is on characterization of uncertainties in samples from human sensors. As stated in previous sections, the system uses only registered volunteers. The system knows each of them and knows that he/she will not lie and have promised to work independently of other participants. However, the person may lack the skills, training and tools to get such data as wind speed and water level accurately. Hence, the sample reported by him/her may contain errors. Specifically, the sample reported by the $k$-th human sensor at $S_i$ contains an additive error component $\Theta_i(k) = (\Theta_{i,1}(k), \Theta_{i,2}(k),\ldots, \Theta_{i,n}(k))$ and that errors of different human sensors are statistically independent and identically distributed. Moreover, virtual sensors being functionally identical, $\Theta_{i,j}(k)$, for each $j = 1, 2, \ldots, n$, is a random variable with known distribution function $B_j(t)$ for all human sensors.

To keep notations used in subsequent discussions simple, we consider only the case of $n = 1$

except where is stated otherwise. In other words, every sensor provides only one type of data. Extension to the general case of $n > 1$ is straightforward when the assumptions of statistical independence of noise components stated above are valid.

The model of human sensors can be further simplified when samples from them are binary values: In terminology of binary hypothesis testing, each human sensor is asked to report whether the hypothesis $H_1$ (e.g., water level is rising) is true or the opposite hypothesis $H_0$ is true. The sample $x$ from the sensor is either equal to 1 or 0, indicating the answer from him/her being $H_1$ or $H_0$, respectively. A sample $x = 1$ (i.e., $H_1$ is true) is a *false alarm* if in fact $H_1$ is not true and is a *detection* if $H_1$ is in fact true. Rather than a statistical distribution in general, the error and uncertainty in a binary valued sample can be characterized by the probabilities of the sample being a false alarm and detection. Subsequent sections use this characterization of binary valued sensors (i.e., sensors with binary valued samples). The *quality* of such a sensor is characterized in terms of a 2-tuple $(f, d)$ of threshold probabilities $f$ and $d$: The probability of a sample from a binary valued sensor of quality $(f, d)$ being a false alarm is at most equal to $f$ and the probability of the sample being a detection is at least equal to $d$.

In the subsequent section, we also use the 2-tuple $(f, d)$ to denote the quality of binary decisions of individual sensors. A binary decision of quality $(f, d)$ means that the decision has a false alarm probability no greater than $f$ and detection probability no less than $d$.

## 4 FUSION AND DECISION PROCEDURE

This section describes the procedure which the central fusion unit such as the one shown in Figure 1 calls to determine whether a condition monitored by the surveillance system has occurred and explain some of the work a fusion center needs to do to fuse and process all sensor data during a CDC process. The procedure, called the *centralized decision fusion* (*CDF*) *procedure*, is built on several principles, approaches and methods from literatures [12-16]. The next section will present the specifics on the techniques used by it. Hereafter, we refer to the fusion unit also as the system when there is no need to be specific.

### 4.1 Design Choices and Rationales

The version of the CDF procedure to be described shortly is for a central fusion unit that functions as a decision module: Its mission is to decide whether an object is present, a phenomenon has occurred, or a specified condition is true, and so on. The system takes action according to the decision. The procedure is presented in terms of binary hypothesis testing and uses $(\alpha, \beta)$-coverage as the desired quality measure. They can be easily replaced by other commonly used methods (e.g., maximum *a posterior* (MAP) and maximum likelihood (ML) decisions and multiple hypothesis testing), and quality criteria (e.g., probability of error and Bayesian costs). A fusion unit in a disaster surveillance system that monitors diverse conditions for diverse purposes will need to have an extensible library of fusion methods, including codes that implement rules for multiple-hypothesis testing to minimize decision error and other optimization criteria.

Alternatively, the surveillance system makes decisions on the basis of not only sensor data but also other information. In that case, the system wants to get from the fusion unit estimates of some parameters. Then, the unit needs to solve an estimation problem or a combined detection and estimation problem. The next section will present examples to illustrate when the capability to perform combined detection and estimation are essential, but leaves the treatments of these problems, as well as multiple-hypothesis testing, to a future paper.

A major design choice of the CDF procedure is that it does decision fusion, making the overall decision on the basis of decisions of individual physical and virtual sensors, not value (data) fusion on the basis of samples (data) from the sensors. Pros and cons of decision fusion versus value fusion have been studied extensively. In general, when compared according to factors such as communication cost, energy consumption, fault tolerance, etc., the relative merit of decision fusion versus value fusion is application domain dependent. In many cases (e.g., [40-42]), decision fusion out performs value fusion. When compared solely on the basis of decision quality, intuition tells us that value fusion should perform better. There is no general proof. A case study on 4 binary sensors with Gaussian additive noise [15] and quality $(f, d)$ = (0.05, 0.95) demonstrated that the best value fusion scheme can achieve a decision quality of $(f, d)$ = (0.001, 0.9998) while the quality achieved by the best decision fusion scheme is $(f, d)$ = (0.014, 0.9995). For our application, even the poorer quality is acceptable, however.

The CDF procedure uses decision fusion despite the apparent disadvantage in result quality. This design choice is made in favor of simplicity and general applicability of the central fusion unit. As stated in Section 3.2, limitation in communication bandwidth makes decision fusion the only option for sensors such as high-resolution camera, radar and so on that generate large volumes of data and have sufficient resources to perform value fusion locally.

Decision fusion is used even for physical sensors (e.g., temperature sensors) that do not generate large volumes of data and do not have sufficient resources to perform fusion. For each of such sensors or a group of such sensors, the central fusion unit first computes a local decision based on values of samples from them using a fusion routine that comes with the sensor. Similarly, it computes a decision for human sensors assigned to each virtual sensor and then fuses the decision with decisions of other sensors. To explain the reason for doing so, we note that noise statistics of physical sensors (e.g., Gaussian additive noise and Poison noise models) differ for different types of sensors and they differ significantly from characteristics of noise in human sensor data (e.g., uniformly distributed or beta distributed errors). Value fusion for all physical and human sensors together is doable in principle but requires fusion routines, each tailored to the signal and noise models of a specific type of physical sensors. Until such routines become available, decision fusion offers a way to build a simple central fusion unit that works for fusing inputs from diverse physical sensors and human sensors.

The version of CDF procedure described below uses the N-P test (i.e., Neyman-Pearson criterion for binary hypothesis test [17]) in decision fusion steps, and whenever applicable, also for value fusion. The next section will describe the test. A reason for using the test is that it does not require a priori probability of each hypothesis. This is important since in almost all cases considered here, the a priori probability of whether a condition of interest is true is not known. Another reason is that the N-P test is optimal (i.e., the most powerful test) in the sense that it maximizes the detection probability for a given acceptable false alarm probability. The test provides the system with control over the tradeoff between these conflicting quality measures. This is an also an important advantage for our application.

### 4.2 Centralize Decision Fusion Procedure.

The CDF procedure is described by the pseudo code in Figure 4. Many terms used in the figure are not yet defined. They will be defined in the next section.

The procedure is called by the central fusion unit whenever the surveillance and early warning system needs to update its information on a condition, renewal its situation awareness and make a decision. Before calling the procedure, the unit polls all physical sensors used to monitor the condition with the help of the communication manager: It acquires from sensors that perform value fusion locally their local decisions and decision qualities. It gets from sensors without value fusion capability their latest readings and computes from the readings the local decisions and decision qualities of these physical sensors. After updating the decisions and decision qualities of all physical sensors, the unit calls the CDF procedure. The fusion unit periodically pools all physical sensors in this way during crowdsourcing processes.

In the pseudo code, as well as in subsequent discussions, lower case $u$, $f$ and $d$ are used to denote the local decision, upper bound to false alarm probability and lower bound to detection probability of individual sensors, respectively. For example, the local decision and decision quality of an individual sensor $S_i$ are $u_i$ and $(f_i, d_i)$, respectively. Capital $U$ and $(F, D)$ are used to denote overall decision and overall decision quality produced by system. Again, the desired decision quality is $(\alpha, \beta)$. In other words, the requirement is $F \leq \alpha$ and $D \geq \beta$ for given $\alpha < 0.5$

and $\beta > 0.5$. The bounds are scenario and situation specific and are typically chosen by someone at the command center.

Centralized Decision Fusion Procedure:

// Check whether existing physical sensors can achieve ($\alpha$, $\beta$)-coverage for given threshold probabilities $\alpha$ and $\beta$

1 do decision fusion for physical sensors using Neyman-Pearson hypothesis testing:
   a. for $i = v+1, v+2, \ldots \sigma$, get local decision $u_i$ and decision quality ($f_i$, $d_i$) of sensor $S_i$;
   b. generate likelihood ratio from $u_i$, $f_i$ and $d_i$ for all $i = v+1, \ldots \sigma$;
   c. for an overall threshold false alarm probability $F = \alpha$, compute decision thresholds $\eta_p \geq \eta^*_p$;
   c. compute overall decision $U$ and detection probability $D$;
2. if $D \geq \beta$, return decision $U$ and decision quality ($F$, $D$) ; // Coverage sufficiently good, terminiate CDF procedure.

// Start a CDC process

3. request broadcast manager to broadcast a Call-For-Participation; wait for responses;
4. wait for selection of participants from responded human sensors and their assignments to $v$ virtual sensors;
5. wait for $M_i$ or more samples $x_i(k)$, for $k = 1, 2, \ldots M_i$ …sent by human sensors assigned to $v_i$, for all $i = 1, 2, \ldots v$;

// Do fusions and then check whether coverage of all sensors is sufficiently good.

6. do value fusion to get local decisions of virtual sensors: for each virtual sensor $S_i$, for $i = 1, 2, \ldots v$, do the following
   a. compute test statistics from samples $x_i(k)$, for $k = 1, 2, \ldots M_i$ …;
   b. compute local decision $u_i$ and decision quality ($f_i$, $d_i$);
7. get local decision $u_i$ and decision quality ($f_i$, $d_i$) for every physical sensor $S_i$;
8. do decision fusion for all sensors using Neyman-Peason hypothesis testing
   a. compute likelihood ratio from $u_i$, $f_i$ and $d_i$, for all $i = 1, 2, \ldots \sigma$;
   b. for threshold false alarm probability $F=\alpha$, set detection thresholds $\eta \geq \eta^*$;
   c. compute overall decision $U$ and detection probability $D$;,
9. if $D < \beta$, request the system to send updated instructions to human sensors; goto step 5; // Current process continues.
10. terminate the current CDC process; return decision $U$ and decision quality ($F$, $D$) ; // Terminate CDF procedure

Figure 4 Centralized decision fusion (CDF) procedure

Steps 1 and 2 in Figure 4 first check whether a decision of sufficiently good quality can be made based on physical sensor data alone. It requests the start of a CDC process only when the quality of the decision is not satisfactory.

In Step 6, the system does for each virtual server *value fusion* of the samples reported by human sensors from a neighborhood around the sensor: The system makes a local decision on whether the specified condition is true based on the sample values.

The system does decision fusion in Step 1 and 7. The system generates the overall decision $U$ and assesses the overall decision quality ($F$, $D$) based on local decisions and decision qualities of physical sensors in Step 1 and of all sensors in Step 7. If after Step 7, the system is not satisfied with the quality of the overall decision, it continues to collect human sensor data, using additional participants and/or redirecting existing participants if necessary.

The pseudo code is explicit about the use of the N-P test in Steps 1 and 7. When the system uses only human sensors selected from registered volunteers with known quality and asked them to send binary valued reports, the test can also be used for value fusion in Step 6. If and when the system must use unknown human sensors, the distributions of the noise components in their samples are unknown. Other methods, including the EM algorithm [36], are warranted to iteratively estimate the model parameters and then make local decisions. Section 6 will discuss the relative performance of the algorithm and N-P test.

## 5 SYMBIOTIC DATA FUSION AND PROCESSING

To state the detection problem for both decision fusion and value fusion, we let the $M$-dimensional random vector $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_M)$ represent the inputs from $M$ independent sensors. The fusion unit receives a sample $\boldsymbol{y} = (y_1, y_2, \ldots, y_M)$ of $\boldsymbol{Y}$ containing a sample $y_i$ of $Y_i$ for every $i = 1, 2, \ldots, M$. In most cases of practical interest, $y_i$'s are discrete valued.

In the context of the CDF procedure, $M$ is equal to the number $\sigma - \nu$ of physical sensors and the number $\sigma$ of all sensors in Steps 1 and 7, respectively. For these decision-fusion steps, $y_i$'s are local decisions of the sensors that are independently made based on their samples. In Step 6, $M$ is equal to the number of human sensors reporting from a virtual sensor. For each value-fusion step, $y_i$'s are the samples reported by the human sensors.

### 5.1 Binary Hypothesis Testing with Neyman-Pearson Criterion

In case of binary hypothesis testing, the fusion unit decides whether a hypothesis $H_1$ (e.g., tar balls on beach) is true or the alternative hypothesis $H_0$ (e.g., no tar ball on beach) is true based on the received input $\boldsymbol{y}$.

As stated earlier, a *false alarm* occurs when the unit decides in favor of $H_1$ when in fact $H_0$ is true, and the unit successes in *detection* of $H_1$when it decides on $H_1$ when $H_1$ is indeed true. The *a priori* probabilities of the hypotheses are unknown typically. The fusion unit works with the conditional probability mass functions, which give the conditional probabilities of seeing $\boldsymbol{y}$ given $H_1$ or $H_0$ is true, respectively. These functions are known.

$$P(\boldsymbol{y} \mid H_1) = P(y_1, y_2, \ldots, y_M \mid H_1) \equiv \Pr[\boldsymbol{Y} = \mathbf{y} \mid H_1] = \Pi_{1 \le i \le M} P(y_i \mid H_1) \tag{1}$$
$$P(\boldsymbol{y} \mid H_0) = P(y_1, y_2, \ldots, y_M \mid H_0) \equiv \Pr[\boldsymbol{Y} = \mathbf{y} \mid H_0] = \Pi_{1 \le i \le M} P(y_i \mid H_0)$$

The last equality in each line follows from the fact that $Y_i$'s are statistically independent.

The CDP procedure aims to maximize the probability of detection for a given threshold false alarm probability $\alpha$. It uses the N-P test for reasons stated earlier. The test statistics is the *likelihood ratio* $L(\boldsymbol{y})$,

$$L(\boldsymbol{y}) = P(y_1, y_2, \ldots, y_M \mid H_1) \,/\, P(y_1, y_2, \ldots, y_M \mid H_0)$$
$$= \Pi_{1 \le i \le M} P(y_i \mid H_1) \,/\, P(y_i \mid H_0) \tag{2}$$

or equivalently, the log function of the likelihood ratio.

The N-P test has two commonly used decision rules, a deterministic rule and a randomized rule [14]. To state these rules, we let $\Lambda$ denote the set of possible values of $L(\boldsymbol{y})$ for all values of

$\boldsymbol{y}$, and let $\eta$ and $\eta^*$ be two adjacent values in $\Lambda$ which are such that $\eta > \eta^*$ and

$$\Pr[L(\boldsymbol{y}) \geq \eta \mid H_0] \leq \alpha \quad (3a)$$

$$\Pr[L(\boldsymbol{y}) \geq \eta^* \mid H_0] > \alpha \quad (3b)$$

Figure 5 illustrate the relationship between $\eta$ and $\eta^*$ as well as their relationship with other values in $\Lambda$ for discrete valued $\boldsymbol{y}$. The dotted and dashed curves are envelopes of conditional probabilities $\Pr[L(\boldsymbol{y}) \mid H_0]$ and $\Pr[L(\boldsymbol{y}) \mid H_1]$, respectively, for all values of $L(\boldsymbol{y})$.

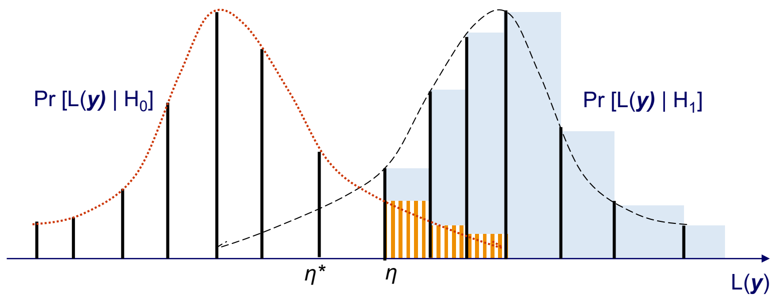



Figure 5 An illustrative example

***Deterministic Rule:*** The *deterministic rule* uses $\eta$ as the detection threshold and selects

$$\begin{aligned} &H_1, \text{ if } L(\boldsymbol{y}) \geq \eta \\ &H_0, \text{ if } L(\boldsymbol{y}) < \eta \end{aligned} \quad (4)$$

The false alarm probability $F$ and detection probability $D$ achieved by rule (4) and detection threshold $\eta$ are given by

$$F = \Pr[L(\boldsymbol{y}) \geq \eta \mid H_0] = \Sigma_{\Lambda(\boldsymbol{y}) \geq \eta} P(\boldsymbol{y} \mid H_0) \quad (5a)$$

$$D = \Pr[L(\boldsymbol{y}) \geq \eta \mid H_1] = \Sigma_{\Lambda(\boldsymbol{y}) \geq \eta} P(\boldsymbol{y} \mid H_1) \quad (5b)$$

As Figure 5 illustrates, $F$ is equal to the sum of $P(\boldsymbol{y} \mid H_0)$ over values of $\boldsymbol{y}$ for which $L(\boldsymbol{y})$ is in the area marked with vertical lines under $\Pr[L(\boldsymbol{y}) \mid H_0]$. D is equal to the sum of $P(\boldsymbol{y} \mid H_1)$ over values of $\boldsymbol{y}$ for which $L(\boldsymbol{y})$ is in the shaded area under $\Pr[L(\boldsymbol{y}) \mid H_1]$. By definition of $\eta$, $F \leq \alpha$.

***Randomized Rule*** The randomized rule uses $\eta$ and $\eta^*$ as detection thresholds together with a random selection probability $p$ which is the solution of the equation

$$p \Pr[L(\boldsymbol{y}) \geq \eta^* \mid H_0] + (1 - p) \Pr[L(\boldsymbol{y}) \geq \eta \mid H_0] = \alpha \quad (6a)$$

The randomized rule selects

$$\begin{aligned} &H_1, \text{ if } L(\boldsymbol{y}) \geq \eta \\ &H_1 \text{ with probability } p \text{ and } H_0 \text{ with probability } 1 - p, \text{ if } L(\boldsymbol{y}) = \eta^* \\ &H_0, \text{ if } L(\boldsymbol{y}) < \eta^* \end{aligned} \quad (6b)$$

The false alarm probability achieved by rule (6) is $\alpha$ by definition of the rule. The detection probability is given by

$$D = p \Pr[L(\boldsymbol{y}) \geq \eta^* \mid H_1] + (1 - p) \Pr[L(\boldsymbol{y}) \geq \eta \mid H_1] \quad (7)$$

***Optimality*** It has been shown in [14] that the randomized rule (6) is optimal among all rules, including complex rules using more than two detection thresholds: It maximizes the detection probability under the constraint of $F \leq \alpha$.

When $\boldsymbol{y}$ is continuous valued, $\eta$ and $\eta^*$ are equal. The deterministic rule (4) is same as the probabilistic rule and hence, is also optimal. It is not optimal when $\boldsymbol{y}$ and hence $L(\boldsymbol{y})$ are discrete valued. The reason is that there may not be a value in the set of all possible values of $L(\boldsymbol{y})$ for which the equality in (3a) holds. In that case, false alarm probability $F$ achieved by rule (4) is less than $\alpha$, and the detection probability $D$ may not be the maximum possible under the constraint $F \leq \alpha$. No other deterministic rules out perform rule (4), however.

### 5.2 Binary Hypothesis Testing Based on Binary Valued Inputs

In the multiple sensor fusion problem treated in [15], local decisions of individual sensors and overall decision of the fusion center are all binary valued. This is assumed by the pseudo code in Figure 4. The special case of hypothesis testing based on binary-valued samples is also of practical importance. In scenarios similar to the one shown in Figure 2(a), samples are naturally binary valued. For practical reasons discussed earlier, the system may also request binary valued reports from human sensors instead of arbitrary valued ones in other scenarios.

To state of the problem of fusing binary decisions formally, we let $y_i = 1$ when the sensor $S_i$ decides in favor of $H_1$ and $y_i = 0$ if it chooses $H_0$. In addition to $y_i$, the fusion unit also knows the decision quality ($f_i$, $d_i$) of the sensor's decision where

$$f_i = \Pr[y_i = 1 | H_0], \qquad d_i = \Pr[y_i = 1 | H_1] \tag{8}$$

for all $i = 1, 2, \ldots, M$ sensors.

Similarly, we let $y_i = 1$ and $y_i = 0$ be the possible values of a sample $Y_i$ reported by the $i$-th human sensor and say that $H_1$ is true if $Y_i = 1$ and $H_0$ is true if $Y_i = 0$ and use the conditional probabilities $f_i$ and $d_i$ defined in (8) as quality measures of the $i$-th human sensor. At risk of abusing the terms, we call them detection and false alarm probabilities, respectively, of the human sensor and call ($f_i$, $d_i$) his/her quality. It is easy to see that the problem of fusing binary-valued samples from human sensors is the same as the problem of binary decision fusion when the quality ($f_i$ , $d_i$ ) is known for every human sensor.

A surveillance system is likely to use *similar sensors*. In addition to being functionally identical, *similar sensors* operate at the same quality level ($f$, $d$), meaning that they use the same threshold false alarm probability $f_i = f$ and achieve detection probability $d_i \geq d$ for all $S_i$. In this case, it suffices for the fusion unit to compute the test statistics from the number $K$ of 1's among the $M$ inputs $y_i$'s. The probability mass functions of $K$ conditional on $H_0$ and $H_1$ are the binomial distributions $B(M, f)$ and $B(M, d)$, respectively. Let $k$ be the sample value of $K$ observed by the fusion unit. The likelihood ratio is given by

$$L(k) = \Pr [K = k \mid H_1] / \Pr [K = k \mid H_0]$$
$$= d^k(1 - d)^{M-k} / f^k(1 - f)^{M-k} \quad (9)$$

By working with log $L(k)$, the randomized rule of the N-P test simplifies to the following: Let $t$ be an integer in $(0, M)$ which is such that

$$\Pr [\, k \geq t \mid H_0] \leq \alpha, \quad \Pr [\, k \geq t - 1 \mid H_0] > \alpha \quad (10a)$$

The simplified randomized rule is: select

$H_1$, if $k \geq t$ (10b)

$H_1$ with probability $p$ and $H_0$ with probability $1 - p$, if $k = t - 1$

$H_0$, if $k < t - 1$

where the selection probability $p$ is given by $p \Pr [\, k \geq t - 1 \mid H_0] + (1 - p) \Pr [\, k \geq t \mid H_0] = \alpha$.

The following theorem states that the fusion center can conclude that it can achieve the desired $(\alpha, \beta)$-coverage if three or more sensors can achieve detection probability $\beta$ or better with the threshold false alarm probability $\alpha$:

**Theorem** *In a system containing $M \geq 3$ similar sensors all of which operate with threshold false alarm probability $\alpha$, $(\alpha, \beta)$-coverage can be achieved using the randomized N-P-test when the detection probability of 3 or more sensors is equal to or higher than $\beta$.*

The theorem is based on the theorem and its proof in [15].

### 5.3 Binary Decisions Based on Arbitrary-Valued Samples

It is straightforward to apply N-P test rules to make optimum binary decisions based on arbitrary-valued samples when their joint distributions are known under both hypotheses. This is especially so when samples are continuous valued, because the simple deterministic rule (4) is optimal. A special case of practical importance is when the data are jointly Gaussian under each hypothesis. This model has been widely used to characterize physical sensor data and has been treated extensively in literature, including [12-16].

The problem of making decisions based on human sensor data of arbitrary values is made complicated by two factors. First, errors (e.g., additive noises) in samples from human sensors are typically not Gaussian. Uniform distribution and some beta distribution are closer models, especially for data from a few (e.g., <5) human sensors. These distributions of noise make application of optimal decision rules and evaluation of their performance complicated.

Second, a more challenging complication is the fact that the system often does not know the values of noise-free components of samples from individual sensors and sometimes, not even their distributions. In other words, each sample $Y_i = V_i + \Theta_i$ from a human sensor is the sum of a random noise-free component $V_i$ and a random additive noise $\Theta_i$. The distribution of noise $\Theta_i$ is known for reasons stated in previous sections. The distribution of $V_i$ may not be fully known under each of the hypotheses.

We would encounter this case in scenarios shown in Figure 2(b) and (c) when human sensors are asked to report wind direction with respect to the direction of wildfire or water depth instead of reporting whether "wind-is-in-direction-of-fire" or "water-is-over-curb" is true or false. Due to the effect of microclimate, wind direction, ambient temperature, etc. at each virtual sensor may differ significantly from that of surrounding area. During a downpour, water may accumulate on some road segments due to poor local drainage but not elsewhere. In both cases, the system may not be able to compute the value of the "signal" $V_i$ contained in the sample from samples from surrounding physical sensors and virtual sensors.

We can formulate the problem of fusing such samples as a joint binary hypothesis testing and estimation problem treated in [43]: We are given the conditional distribution of the samples $\boldsymbol{Y}$ from $M$ sensors under each hypothesis $P(\boldsymbol{y} \mid H_0)$ and $P(\boldsymbol{y} \mid H_1, \boldsymbol{\Phi})$, where $\boldsymbol{\Phi}$ is a random parameter with a known probability density (or mass) function. The solution of the problem gives us a rule to decide in favor of $H_0$ or $H_1$, and if the decision is in favor of $H_1$, compute an estimate of $\boldsymbol{\Phi}$. As an illustrative example, supposed that the human sensors in Figure 2(c) are asked to report water depth on a street in a small number of city blocks. $\boldsymbol{\Phi}$ is the amount of local rainfall or actual water depth. Its probability density function can be derived or estimated from data on measured or forecast rainfall of the surrounding area and historical records. $H_0$ is "no flooding danger" and $H_1$ is "flooding possible" and the action to be taken by the system depends on the estimate of $\boldsymbol{\Phi}$. We will apply and evaluate the schemes described in [43] to this and similar scenarios and report the result in a future paper.

## 6 PERFORMANCE ISSUES

This section first present data on how decision quality of a virtual sensor depends on the number and qualities of human sensors assigned to it. The participant selection module in Figure 1 can use this data as input. The performance of the N-P test is then compared with the performance of EM algorithm [36] for fusing binary-valued human sensor data.

### 6.1 Dependence of Minimum Required Human Sensors on Sensor Quality

For our application, the acceptable false alarm probability is in order of 10% for some scenarios and 1 % for other scenarios. This quality criterion is not met by typical human sensors, and some physical sensors. It is well known that fusion center can improve the false alarm probability and detection probability of the overall decision by using the N-P-test on samples from multiple sensors [14, 15]. A question of practical interest is how many sensors are required to get the overall false alarm probability $F \leq \alpha$ and overall detection probability $D \geq \beta$ as a function of the sensor quality $(f, d)$ and required decision quality $(\alpha, \beta)$.

We can find answers to this question from the expressions of the overall quality measures $F$ and $D$ in terms individual quality measures $f$ and $d$ of $M$ similar sensors. To illustrate, suppose that the fusion center uses the simple deterministic rule (4) with a single detection threshold $t$. Then, the overall $F$ and $D$ are given in closed form by

$$F = \sum_{t \leq k \leq M} C(M, k) f^k (1-f)^{M-k} \tag{11}$$
$$D = \sum_{t \leq k \leq M} C(M, k)\, d^k (1-d)^{M-k}$$

where $C(M, k)$ denotes the binomial coefficient $M!/k!(M-k)!$. We can obtain from the equations above the minimum number of sensors of quality $(f, d)$ required to achieve the overall decision quality $(F, D)$ by solving these equations when $f < 0.5$ and $d > 0.5$.

Figure 6 plots the minimum number of sensors of quality $(f, d)$ required to achieved $(\alpha, \beta)$-coverage for several likely combinations of sensor quality $(f, d)$ and required quality $(\alpha, \beta)$ when the optimal probabilistic rule (6) is used. The combinations in Figure 6 tell us what intuition tells us all along: It is better to use a relatively small number of high quality human sensors (e.g., people with $(f, d) = (0.2, 0.8)$) than a big crowd of poor human sensors (e.g., people with $(f, d) = (0.4, 0.6)$). By being better, we mean that fewer human sensors are needed to achieve the specified overall quality.

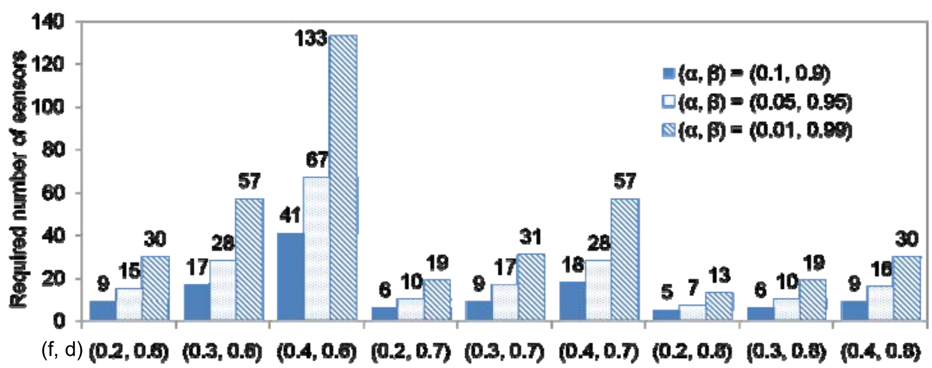



Figure 5 Required numbers of sensors of quality $(f, d)$ to achieve $(\alpha, \beta)$-coverage

While the deterministic rule is not optimal, it does not lead to a larger number of sensors required to achieve the required quality for most of the combinations plotted in Figure 6. Exceptions are listed in Table 1: In these cases, one or two more sensors are required when the fusion center uses the simpler deterministic rule. Values of $F$ and $D$ in bold and red font give the overall decision quality $(F, D)$ when a sufficient number of human sensors is used to meet the required quality. In each case, the probability rule indeed enables the fusion center to achieve the required quality with fewer sensors. On the other hand, when the additional one or two sensors are used, the deterministic rule often overshoots the quality requirement. For this reason and because that the rule is simple to evaluate, we use the rule and equations (11) to compute the overall $F$ and $D$ achieved by the N-P test.

Table 1 Performance Degradation by Using Deterministic Rules

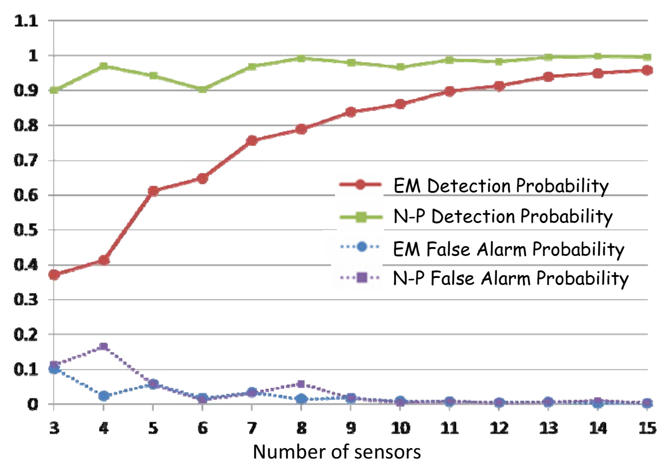
| Sensor quality (f, d) | Required quality ($\alpha$, $\beta$) | No of sensors | Deterministic rule | | Probabilistic rule | |
|---|---|---|---|---|---|---|
| | | | F | D | F | D |
| (0.2, 0.6) | (0.05, 0.95) | 15 | 0.01805881 | 0.90495259 | ***0.05*** | ***0.95043136*** |
| | | 16 | 0.02665733 | 0.94168106 | 0.05 | 0.95829903 |
| | | 17 | ***0.03766344*** | ***0.96518727*** | 0.05 | 0.96957856 |
| (0.2, 0.7) | (0.01, 0.99) | 19 | 0.00665766 | 0.98945884 | ***0.01*** | ***0.99101102*** |
| | | 20 | ***0.00998177*** | ***0.99486184*** | 0.01 | 0.99486501 |
| (0.3, 0.6) | (0.1, 0.9) | 17 | 0.04027694 | 0.80106351 | ***0.10*** | ***0.90038417*** |
| | | 18 | 0.05958588 | 0.86528585 | 0.10 | 0.90369128 |
| | | 19 | ***0.08391516*** | ***0.91152594*** | 0.10 | 0.92026044 |
| (0.3, 0.8) | (0.01, 0.99) | 19 | 0.00282259 | 0.97672169 | ***0.01*** | ***0.99217705*** |
| | | 20 | ***0.00513816*** | ***0.99001821*** | 0.01 | 0.99300941 |
| (0.4, 0.7) | (0.1, 0.9) | 18 | 0.05764734 | 0.85931654 | ***0.10*** | ***0.90388434*** |
| | | 19 | ***0.08847406*** | ***0.91608484*** | 0.10 | 0.92214923 |
| (0.4, 0.8) | (0.05, 0.95) | 16 | 0.01914192 | 0.91831211 | ***0.05*** | ***0.96165735*** |
| | | 17 | ***0.03481273*** | ***0.96233656*** | 0.05 | 0.96944756 |

### 6.2 N-P Test versus EM Algorithm

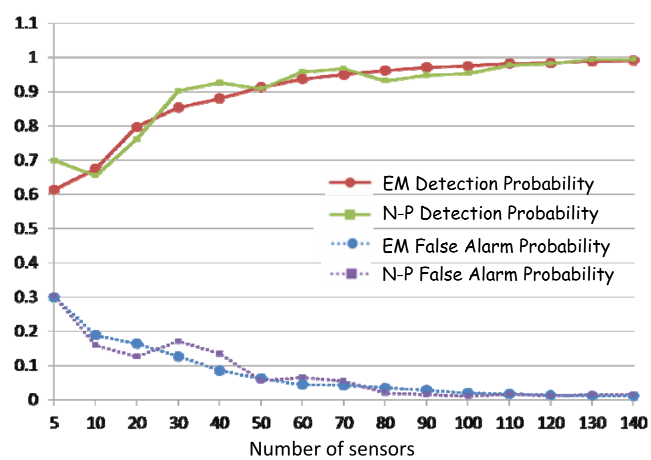
In particular, we use the deterministic rule in a study to compare the N-P test and the EM algorithm [36]. The algorithm was used by Wang, *et al*. [34] for binary hypothesis testing based on binary valued human sensor data collected from unknown crowd. A distinct advantage of the EM algorithm over the N-P test is that it does not required knowledge of sensor quality. Assuming that human sensors are similar, the algorithm treats $f$ and $d$ as unknown model parameters. Starting from initial guesses of values $f < 0.5$ and $d > 0.5$, the algorithm iteratively computes estimates of these model parameters, uses the estimates to compute likelihood ratio, makes decision and computes new estimates of the parameters from the likelihood ratio, and so on. A major advantage of the N-P test is its simplicity, especially the deterministic rule for processing samples from similar sensors. In addition, it is optimal in the sense that it maximizes detection probability for given threshold false alarm probability.

The goal of our comparison study is to quantify their relative merits in terms of quality of overall decision as a function number of sensors. Specifically, we compared the performance of N-P test and EM algorithm using a scenario similar to the one used in [34]: Human sensor data are binary valued. Hypotheses $H_1$ and $H_0$ are equal probable. Human sensors are similar with the same quality ($f$, $d$). They report binary valued samples, indicating their choices in favor of $H_1$ and $H_0$ with a "1" or "0", respectively.

We simulated the scenario. During each simulation run, the number of sensors and their quality ($f$, $d$) are fixed. The value of the sample from each sensor is chosen at random with probability $f$ being false alarm and $d$ being detection. After collecting samples from all sensors, the N-P test computes numerically from the values of the samples, the achievable overall false alarm probability $F$ and detection probability $D$ with the given number of sensors using equations (11) for the deterministic rule. The EM algorithm estimates $F$ and $D$ iteratively starting from some initial guesses of $f < 0.5$ and $d > 0.5$. After repeating a sufficient number of runs to get a data point, we increased the number of sensors by 1 and repeated the runs to get another data point. This process was repeated until the overall decision quality ($F$, $D$) meets the desired overall quality of ($\alpha$, $\beta$) = (0.01 and 0.99). Figure 7 shows the results thus obtained.




(a) Sensor quality ($f$, $d$) = (0.2, 0.8)




(b) Sensor quality ($f$, $d$) = (0.4, 0.6)

Figure 7 Relative performance of N-P test and EM algorithm

Specifically, Figure 7 shows how the average overall false alarm probability and detection probability achieved by the schemes depend on the number of sensors. The value plotted at each point is the average of the values obtained in 100 simulation runs. The plots in part (a) were obtained from simulation of good sensors with quality (0.2, 0.8), while sensors simulated to generate plots in part (b) have quality (0.4, 0.6).

We found that the results produced by the EM algorithm are not sensitive to the initial guesses of $f$ and $d$, provided that $f < 0.5$ and $d > 0.5$; otherwise, its estimates of $F$ and $D$ do not converge to the desired quality as the number of sensors increases.

From these plots, one can see that except for where there are only a few human sensors (say < 10), the performance of N-P test and EM algorithm are comparable. When only sensors of known quality are used, the N-P test not only performs as well or better but also is much simpler to implement and efficient to run.

Finally, we note that because the deterministic rule of the N-P test is used, the overall false alarm probability $F$ in the figure is not a constant $\alpha = 0.01$. That would be the value achieved independent of number of sensors if the probabilistic rule were used, by definition of the rule. The non-monotonic behavior of false alarm and detection probabilities achieved by the N-P test as the number of sensors increases is due to variations in the degradation from the optimal suffered by the deterministic rule. Table 1 shows examples illustrating that the degradation is larger for some combinations of sensor quality and sensor number than other combinations.

## 7 SUMMARY AND FUTURE WORK

A crowdsourcing support system such as CROSS for disaster warning and response purposes not only provides mechanisms and tools for managing crowdsourcing human sensor data collection. It must also provide supports for fusion of data from physical and human sensors. We described in the previous sections the work done by a central fusion unit to process and fuse inputs from physical surveillance sensors together with human sensor data collected from participants during a system-driven crowdsourcing process. It may use a combination of value fusion and decision fusion in ways exemplified by the CDF procedure. The goal is to reach a decision of some specified quality or better on action(s) to be taken by the system, and to do so with the fewer human sensor reports, the better.

By taking into account realistic restrictions on how human sensors are used for collecting disaster surveillance data purposes and requiring the system to know bounds to their qualities as sensors, we are able to formulate the problems of fusing surveillance physical sensor data and crowdsourced human sensor data as classical statistical detection and estimation problems. By doing so, we are able exploit well established principles and techniques for fusion in multiple physical sensor systems and focus our attention on incorporating the fusion of human sensor data within a coherent framework with fusion for physical sensors.

In the immediate future, we will evaluate via numerical computations and simulation experiments alternative solutions based on this approach, including the ones described in the previous sections, for the types of physical and human sensors that are likely to be used in different disaster scenarios, including the ones used in scenarios shown in Figure 2. We will add to CROSS a prototype central fusion unit using the CDF procedure and an extensible library of fusion methods as a starting point. .

Thus far, our effort has been focused on binary hypothesis testing. A natural next step is to provide the fusion unit with code that applies the *maximum a posteriori* (*MAP*) rule for multiple-hypothesis testing to minimize probability of error and rules for computing parameter estimates according to specified optimization criteria.

It is also urgent to provide the system with the capability of making decisions and estimates based on data with incomplete models because the system does not know either the noise-free values in the samples or the distributions of noises. The former arises in scenarios such as the ones in Figure 2(b) and (c) for reasons discussed in Section 5.3. We will exploit the optimum tests (e.g., combining N-P test and ML estimation) for joint detection and estimation proposed recently by Moustakides, *et.al*., [43] to build solutions for these scenarios. We have the latter case in scenarios when the system has no choice but to use unknown participants with unknown noise characteristics. Other methods, including the EM algorithm [36], are warranted to estimate model parameters and make local decisions or estimations.

**ACKNOWLEDGEMENT**

This work was supported by the Taiwan Academia Sinica thematic project OpenISDM (Open Information Systems for Disaster Management).

**REFERENCES**

[1] Aggarwal, C. C. and T. Abdelzaher, "Integrating Sensors and Social Networks," in *Social Network Data Analysis,* Ed. by C. Aggarawal, Springer, 2011.

[2] Rosi, A., *et al.*, "Social Sensors and Pervasive Services: Approaches and Perspectives," *Proc. of PERCOM*, March 2011.

[3] Beach, A., *et al.*, "Fusing Mobile, Sensor, and Social Data to Fully Enable Context-Aware Computing," *Proc. of the 12th ACM Workshop on Mobile Computing Systems and Application* (HOTMOBILE 2011), March 2011.

[4] Demirbas, M., *et al*., "Crowd-Sourced and Collaboration Using Twitter," *Proceedings of*

*IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, (WOWMOM), 2010

[5] List of Crowdsourcing Projects, http://en.wikipedia.org/wiki/List_of_crowdsourcing_projects.

[6] Boulos, M. N. K., *et al.*, “Crowdsourcing, Citizen Sensing and Sensor Web Technologies for Public and Environment Health Surveillance and Crisis Management: Trend, OGC Standards and Application Examples,” *International Journal on Health Geographics,* pp 1-67, 2011.

[7] Ferrell, J., “Crowdsourcing Snowstorm's Westward Model Shift,” at http://www.accuweather.com/en/weather-blogs/weathermatrix/highres-models-shift-tonights-snow-west/58754 and NoiseTube, at http://noisetube.net/

[8] Saenz, A. “Japan’s Nuclear Woes Give Rise to Crowd-Sourced Radiation Maps in Asia and US,” at http://singularityhub.com/2011/03/24/, 2011

[9] Chu, E. T. H., Y. L. Chen, J. Y. Lin and J. W. S. Liu, "Crowdsourcing Support System for Disaster Surveillance and Response," *Proceedings of the 15th International Conference on Wireless Personal Multimedia Communication*, September 2012.

[10] Chu, E. T.-H., Y.-L. Chen, J. W. S. Liu and J. K. Zao, “Strategies for Crowdsourcing for Disaster Situation Information,” *WIT Transactions on the Built Environment*, Vol.119, pp. 257-269, 2011.

[11] Chu, E. T., C.-Y. Lin, P. H. Tsai and J. W. S. Liu, “Participant Selection for Crowdsourcing Disaster Information,” to appear in *Proceedings of Disaster Management 2013*, the 3rd International Conference on Disaster Management and Human Health, July 2013

[12] Xing, G., R. Tan, B., *et al.*, “Data Fusion Improves the Coverage of Wireless Sensor Networks,” *Proceedings of MibiCom*, 2009.

[13] Gallager, R. G.,*Stochastic Processes: Theory for Applications,* November 21, 2012, http://www.rle.mit.edu/rgallager/notes.htm,

[14] A. S. Willsky, G. W. Wornell, and J. H. Shapiro, “Stochastic processes, detection and estimation,” MIT 6.432 Course Note, Fall 2003.

[15] Thomopoulos, S. C. A., R, Viswanathan, and D. C. Bougoulias, “Optimal decision fusion in multiple sensor systems,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol.23, No. 5, pp. 644 – 653, September 1987.

[16] Chair, Z. and P. K. Varshney, “Optimal Data Fusion in Multiple Sensor Detection Systems,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 22, No. 1, January 1986.

[17] Scott, C., "Neyman-Pearson criterion," h.ttp://cnx.org/content/m11548/latest/

[18] Sahana, an open source disaster management system, http://www.sahanafoundation.org/

[19] Ushahidi, an open source project for crowd sourcing crisis management, http://www.ushahidi.com/ and Mission4636, http://www.mission4636.org/

[20] Nincompoop, Statistical Tools for Extracting Ground Truth from Crowdsourced Data Sets, http://code.google.com/p/nincompoop/

[21] OpenCalais, A toolkit for Incorporating State-of-the-Art Semantic Functionality into Blog, Content Management System, Website or Application, http://www.opencalais.com/

[22] C. C. Aggarwal and T. Abedelzaher, "Social Sensing," in *Managing and Mining Sensor Data*, Springer, 2013.

[23] G. Barbier, "Maximizing Benefits of Crowdsourced Data," *Computational and Mathematical Organization Theory*, Vol. 18, No. 3, August 2012

[24]C. Wagner, "Extracting Meta-Measures from Data for Fuzzy Aggregation of Crowd Sourced Information," *Proceedings of IEEE Conference on Fuzzy Systems*, June 2012.

[25] L. P. Cox, "Truth in Crowdsourcing," *IEEE Journal on Security and Privacy*, Vol. 9, No. 3, September-October 2011.

[26] F. Gilbert, *et al.*, "YouProve, Authenticity and Fidelity in Mobile Sensing, " *Proceedings of ACM Embedded and Networked Sensor Systems,* November 2011

[27] M. Lease, "On Quality Control and Machine Learning," *Proceedings of the 3rd Human Computation Workshop*, 2011.

[28] "Developing Swift River to Validate Crowd Sourcing," http://irevolution.wordpress.com/2009/04/10/developing-swift-river-to-validate-crowdsourcing/, April 2010.

[29] Ipeirotis, P. G., *et al*, "Quality Management on Amazon Mechanical Turk," *Proceedings of ACM SIGKDD Workshop on Human Computation,* 2010.

[30] Le, J., *et al,* "Ensuring Quality in Crowdsourced Search Relevance Evaluation: The Effects of Training Question Distribution, *Proceedings of the SIGIR 2010 Workshop on Crowdsourcing for Search Evaluatio*n, 2010.

[31] Zhu, D, and B. Carterette, "An Analysis of Assessor Behavior in Crowdsourced Preference Judgments," *Proceedings of the SIGIR 2010 Workshop on Crowdsourcing for Search Evaluation,* 2010.

[32] Hsueh, P.Y., *et al*, "Data Quality from Crowdsourcing: A study of Annotation Selection Criteria," *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, 2009.

[33] X. Yin, H. Han, and P. S. Yu, "Truth Discovery with Multiple Conflicting Information

Providers on the Web," *IEEE Trans. on Knowledge and Data Engineering,* June 2008.

[34] Wang, D., H. Le, L. Kaplan, and T. Abdelzaher, "On Truth Discovery in Social Sensing: A Maximum Likelihood Estimation Approach," *Proceedings of the 11th international conference on Information Processing in Sensor Networks*, 2012.

[35] D. Wang, *et al*., "On Bayesian Interpretation of Fact-Finding in Information Networks," *Proceedings of International Conference on Information Fusion, 2011.*

[36] Gupta, M. R. and Y. Chen, "Theory and use of EM algorithm," *Foundation and Trends in Signal Processing*, Vol. 4, No. 2, 2010

[37] Open GeoSMS Standard, http://www.opengeospatial.org/standards/opengeosms

[38] Oncan, T., "A Survey of Generalized Assignment Problem and Its Applications," *Information Systems and Operation Research*, May 2008.

[39] Bektas, T., "The Multiple Traveling Salesman Problem: An Overview of Formulations and Solution Procedures," *Omega, the International Journal of Management Science,* Vol. 34, 2004.

[40] Katenka, N., E. Levina, and G. Michailidis, "Local Vote Decision Fusion for Target Detection in Wireless Sensor Networks," *IEEE Transactions on Signal Processing,* Vol. 56, No. 1, January 2008.

[41] D'Costa, A. M. and A. M. Sayeed, "Data versus Decision Fusion for Classification in Sensor Networks," *Proceedings of the Sixth International Conference on Information Fusion*, July 2003.

[42] Clouqueur, T., "Value-Fusion Versus Decision-Fusion for Fault-Tolerance in Collaborative Target Detection in Sensor Networks," *Proceedings of Fourth International Conference on Information Fusion*, 2001.

[43] Moustakides, G. V., *et al*., "Joint detection and estimation: optimum tests and applications," *IEEE Transactions on Information Theory,* Vol. 58, No. 7, July 2012.