Context Aware Adhoc Network for Mitigation of Crowd Disasters

3 authors:

Maneesha Vinodini Ramesh
Amrita Vishwa Vidyapeetham, Amritapuri, Ko…
117 PUBLICATIONS   **431** CITATIONS

Shanmughan Anjitha
4 PUBLICATIONS   **17** CITATIONS

Rekha Prabha
Amrita Vishwa Vidyapeetham
22 PUBLICATIONS   **47** CITATIONS

Some of the authors of this publication are also working on these related projects:

Project   Healthcare data analytics and communications View project

Project   Mobile Infrastructure for Coastal Region Offshore Communications & Networks (MICRONET) View project

Context aware ad hoc network for mitigation of crowd disasters

Maneesha Vinodini Ramesh *, Anjitha Shanmughan, Rekha Prabha

AMRITA Center for Wireless Networks & Applications, AMRITA Vishwa Vidyapeetham (AMRITA University), India

ABSTRACT

Our research works focuses on the design and implementation of a novel ubiquitous multi context-aware mobile phone sensing network for mitigation of crowd disasters using machine-to-machine (M2M) communications. A mobile sensor network system integrated with wireless multimedia sensor networks (WMSNs) was designed for effective prediction of a stampede during crowd disasters. This proposed sensor network consists of mobile devices that are used as crowd monitoring participant nodes that employ light sensors, accelerometers, as well as audio and video sensors to collect the relevant data. Real-time crowd dynamics modeling and real-time activity modeling have been achieved by implementing the algorithms developed for Context Acquisition and multi-context fusion. Dynamic crowd monitoring was achieved by implementing the context based region identification and grouping of participants, distributed crowd behavior estimation, and stampede prediction based on distributed consensus. The implementation of the proposed architecture in Android smartphone provides light-weight, easy to deploy, context aware wireless services for effective crowd disaster mitigation and generation of an in time alert to take measures to avoid the occurrence of a stampede. The system has been tested and illustrated within a group of people for stampede prediction by using empirically collected data.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Wireless Sensor Networks (WSNs) consists of large number of sensor nodes which are capable of linking the physical world with the digital world by capturing and revealing real-world phenomena and converting these occurrences into a format that can be processed, stored and acted upon [31]. These sensors are low power devices that can be used for a wide range of applications beneficial for society. The accurateness and the timeliness of this sensed information is extremely important to take critical action in various real world situations. Hence, the key consideration of our recent research work in WSN is to improve the quality and reliability of the sensed information by utilizing the potential of Context Aware Computing (CAC). This context awareness helps to configure the WSN to adapt and react according to the real time situation of the physical space in which the sensors are deployed. The key focus of our research work is to integrate the concept of CAC for WSN (CAC-WSN) to predict the onset of abnormality within a crowd.

Context aware computing is the key enabling technology for pervasive computing. The major classification of context includes computing context, user context and physical context [30]. Computing context includes network connectivity, communication costs, communication bandwidth and nearby resources such as printers, displays and workstations. User context specifies the user's profile, location, and people nearby, even the current social situation. Physical context includes lighting, noise levels, traffic conditions and temperature.

Design and deployment of context aware systems in open and dynamic environments is raising a new set of research challenges such as Sensor Data Acquisition, Context

* Corresponding author. Address: AMRITA Center for Wireless Networks & Applications, Amrita Vishwa Vidyapeetham (AMRITA University), Amritapuri Campus, Clappana P.O., Kollam 690 525, Kerala, India.

E-mail addresses: maneesha@am.amrita.edu (M.V. Ramesh), anjithas@gmail.com (A. Shanmughan), rekhap@am.amrita.edu (R. Prabha).

Data Fusion, Context Modeling and Reasoning, Service Discovery, and Execution of Services for the user. One of the crucial requirements of context-aware applications is real-time collection and aggregation of sensor data from sensors distributed in the environment.

The proposed Ubiquitous Multi-Context Model (UMM) consists of three major phases, namely the Context Acquisition phase, the Context Modeling and Inference phase, and the Context based Action Generation phase. The key problem under consideration of our model is to develop a distributed context aware architecture with key features such as reusability, non-redundancy and wide area coverage of context information.

The recent advancements in the field of WSN make use of mobile phones, specifically smartphones, to act as sensor nodes. The mobile phone penetration rate is increasing tremendously all over the world, and in India it is expected to reach up to 97% of the total population in the year 2014 [10]. The increased penetration rate of smartphones in everyday life has resulted in the development of many promising applications utilizing low cost embedded sensors such as accelerometers, microphone, camera, gyroscope, and light sensors. This new research area is referred to as mobile phone sensing.

The implementation of our UMM is performed on an Android framework using mobile phone sensing and CAC-WSN. The objective of our research work is to make use of UMM for applications suitable for both personal sensing (e.g. individual activity) and group sensing (e.g. crowd monitoring).

The demand for security and safety within public spaces is gaining attention nowadays due to the increase in crowd disasters. The brief history of crowd disasters in public gatherings for pilgrimage, sports events, etc. from 1989 to 2011 is reported in the reference paper [9]. India experiences more than 100 deaths per year due to crowd disasters. The major motivation for our research work is the stampede of pilgrims that occurred at the hilltop Sabarimala shrine in the state of Kerala in southern India, which took the life of 102 devotees [9]. The stampede was set off when a jeep drove into the crowd of pilgrims. The pilgrimage area was flooded with people and the situation went uncontrollable. The other causes of crowd disasters are terrorist activities that are focused on public gatherings resulting in the loss of life of many innocent people.

Integration of context aware computing with mobile phones makes the devices capable of sensing the physical world, process the current scenario, and adapt and react to dynamic changes of the environment. The proposed system uses the above-mentioned capability to develop an application suitable for crowd management, real-time monitoring of crowd behavior, and dissemination alerts and instructions for crowd control.

2. Context and context aware computing

Researchers defined the term context in a slightly different manner such as, Schmidt defined context as knowledge about the state of the user's IT devices, along with information on the surroundings, situation, and to a less extent, location [3]. Chen and Kotz defined context as the set of environmental states and settings that either determined an application's behavior or in which an application event occurred and is interesting to the user [4]. Dey et al. gave the definition of context as any information that is useful to highlight the interaction between a user and an application and also used to characterize the situation of an entity such as a person, place or object [5]. The knowledge of data along with its context will provide insights into the complex patterns hidden in those data. This understanding lead to the development of the new area of "context awareness" in ubiquitous computing.

The term "context-awareness" in ubiquitous computing was initially introduced by Schilit in 1994 [1,2]. According to Schilit [1], context awareness is the capability of a system to "adapt according to the location of the user, the collection of information about nearby people, hosts, accessible devices, as well as the changes to such things over time". Pascoe et al. [6] define context-awareness as the ability to detect, gather, interpret and react to context changes in the user's surrounding and changes in its device.

The research paper [6] discusses the importance of context-aware sensing and provides a general overview of acquiring context information and various context aware applications.

These context aware frameworks detailed several open research challenges such as:

- Context Acquisition Issues
  - Determination of appropriate sensors and the type of context to be acquired. Some of the sensors should initially be chosen for recognizing context because most of the applications do not require context data from all the available sensors. Use redundant sensors only in case of uncertainty. In this way, data transmission over the network decreases together with the processing demand.
  - Real time management of the sensors. Providing feedback to the sensors and dynamically managing them during the operational phase can improve the application's performance, improving the overall result.
  - Determination of who the context capturers will be and the number required. Assess the need for predetermined participants or new users of the application and determine how many smartphones are required for the best context inference. Develop automatic techniques to prevent redundant information capture due to close proximity of smartphones.
  - Determination of the most suitable time-period of monitoring with respect to the application.
- Context Modeling & Inference Issues
  - A distributed context aware inference formulating framework is required.
  - Since new contexts are continuously added in the sensor network, the requirement of continuous adaptive learning algorithm is an issue that draws attention.
  - Context inference is a complex task that requires a good mechanism for mapping simple captured context data to a higher level of data. This is not an easy

task and requires using various context modeling techniques. Research is still ongoing to find the best method that will disregard errors, automatically adapt the system to new types of data, learn autonomously, and reason correctly.

- Context-based Action Generation Issues
  - Visualizing results and incorporating results on the screens of hand-held devices.
  - Overcoming low bandwidth in wireless networks by introducing algorithms that minimize communication as well as the number of hops.
  - Limited battery power.
  - Minimizing the data-sampling frequency to save energy and communication cost.

Based on the analysis of various context aware frameworks, shown in Table 1, the observations that can be extended to our system under development for multiple application support include: the agent based architecture of CoBrA and the distributed middleware architecture of SOCAM and Gaia.

The recent studies illustrated in [7] highlight the use of smartphones for context-aware applications because they are relatively powerful in processing and they also support various embedded sensors. Various smartphone platforms available are Android, iPhone, Symbian, RIM, Windows phone and Linux. The two most promising platforms were iPhone and Android because of their popularity, high usability, powerful CPUs, and available sensors.

The key features of Android includes support for machine learning based programming in Java, access to more core OS functionality, no requirement of any certification or developer registration to deploy the software to the hardware, and the Android SDK is available on multiple platforms [8]. The Android application framework has built-in context support. The main two parts are the raw context sources and context processing [7].

The raw context source support includes packages and classes for Bluetooth, a camera, and a sensor manager for collecting data from the embedded sensors on the Android device (location data, time, etc.). Context sources provide necessary information about the entity it is associated with. Smart context sources supported by Android can be categorized as hard and soft sensors [8]. The hard sensors include:

- Accelerometer: A tri-axial accelerometer is a sensor that returns a real valued estimate of acceleration along the x, y and z axis from which velocity and displacement

can also be estimated. Accelerometers can be used for motion detection, position, and posture sensing.
- Digital Compass: The digital compass provides two measures; the orientation and the ambient magnetic field. The values for orientation are in radians/second and measure the rate of rotation around the X (roll), Y (pitch) and Z (yaw or Azimuth). The compass measures the ambient magnetic field in the X, Y and Z axis, and those values are in micro-Tesla (μT).
- Gyroscope: Gyroscopes are used for measuring angular velocity, angular rotation, and the rate of rotation around the X, Y and Z axis. Gyroscopes are used for navigation and homing applications.
- Location sensor: Smartphone provides two major modes of location tracking: an Assisted Global Positioning System (A-GPS) and Global System for Mobile Communications (GSM) cell tower triangulation.

Social Networks Sites (SNSs) are becoming increasingly popular and are referred to as soft sensors. SNS users share their daily personal information via SNS. Context aware applications are progressing to utilize the user specific information collection from SNS.

The context processing enables the management of this raw context data, from hard and soft sensors, into useful contextual data such as speech recognition, face recognition, text-to-speech, and location proximity.

Using an analysis of existing frameworks, we developed the new idea of integrating mobile phone sensing with the potential of wireless multimedia sensor networks (WMSNs) to reduce the geographical restriction of monitoring. To achieve reliable alert generation with real time values and ease of deployment and maintenance, our proposed crowd abnormality detection system utilizes the capabilities of WMSN and the embedded smartphone sensors. Our research work also brings in multi-sensor modalities for activity recognition. Another key objective is to estimate the probability of stampede occurrence using mobile phone sensing.

3. Crowd monitoring systems

Based on the study we conducted on crowd disasters, we identified two major problems that contribute to crowd disasters. They are inadequate space and individual loss of physiological and psychological control. Conventional manual crowd monitoring is a tedious effort, requiring the dedicated participation of many security officers. Manual monitoring has been replaced by closed circuit televi-

Table 1
Study of existing context aware frameworks.

Context aware framework	Specific features analyzed for the development of new multiple application supporting framework: UMM
CoBrA [1]	"Context Broker" component spans multiple application context aware agents by providing shared model of contexts
SOCAM [2]	Distributed with centralized server and it performs context modeling based on ontologies
Gaia project [3]	Distributed middleware infrastructure which is intended to coordinate the development and execution of mobile applications
Cenceme [4]	Sharing of context information in web portal
InContexto [5]	Obtain context from a smartphone user

sion systems (CCTV). The research paper [11] discusses crowd monitoring using an image processing approach that includes an automated CCTV based technique. The cost of deployment and maintenance of these systems was very costly which led to the development of Wireless Sensor Networks (WSNs) based crowd monitoring, which uses low cost sensors and real time wireless communications.

A prototype for stadium surveillance based on WSN, integrated with increased situational awareness, was proposed for better knowledge of real-time incidents within a stadium [12]. However, this system led to a high rate of false alarms and was less reliable as it only used temperature and acoustic sensors to develop situational awareness [12]. Due to the geographical restriction of monitoring due to fixed deployment of WSN, the researchers brought in the idea of mobile phone sensing for effective crowd behavior monitoring. The research work in this area is in progress and opens up very promising studies for wide area sensing and communication utilizing the potential of the embedded sensors in smartphones.

The use of mobile phone sensors for effective activity recognition is illustrated in [13,14]. The disaster evacuation methods using the sensing capabilities of mobile phones were given in [15,16]. One of the major research studies in the area of mobile phone sensing is the decentralized detection of group formation using wearable sensors in [17]. In that work Wirz et al. details the experimental results and detection of walking by a group of people based on body worn accelerometer sensors and proved the applicability of mobile phone sensing for crowd monitoring in real life situations. The experimental studies were based on accelerometer sensor alone, and the authors suggested the additional integration of multi-sensor modalities for better behavior recognition. A mobile phone sensing framework for crowd behavior estimation is described, and an experimentation illustration of effectiveness of mobile phone sensing for group activity recognition is given in [28].

4. Context aware ad hoc network for mitigation of crowd disasters

Stampedes are one of the major reasons for recent crowd disasters that have taken the lives of hundreds of people. Our proposed crowd disaster mitigation sys-tem focuses on estimating the probability of occurrence of a stampede in a crowded area based on the distributed sensor data fusion and analysis of sensor values from the tri-axial accelerometer, GPS, acoustic sensors and video sensors, thereby aiming to avoid or decrease the impact of imminent stampede. In the stampede suspicious area, this system will provide the service to make the crowd controlling station more vigilant and provide alerts on facilitating evacuation schemes in case of emergency.

4.1. Crowd dynamics and modeling

The scientific definition of a crowd is given as a concentration of a large number people in the same area at the same time. The high density of a crowd causes continuous interactions with or reactions to other individuals. There are different types of crowds: homogeneous, heterogeneous, expressive and aggressive. Aggressive crowds get more attention. Examples of two types of crowd are shown in Fig. 1. The pilgrimage crowd scenario is a heterogeneous crowd that is composed of dissimilar elements. They have no common bonds other than the event that brought them together.

Now-a-days, mass gatherings are prone to crowd disasters. The studies related to crowd disasters indicate the presence of crowd stampede as the major life threatening cause behind the onset of such disasters. The major anomalies in the crowd that leads to stampede are

(a) Unpredictable obstacle in the flow path of the crowd resulting in increased pressure.
(b) 'Flight' or 'Panic', where people experience either a real or perceived threat. This triggers crowd movement which leads to the occurrence of the event.
(c) A competitive rush to gain a highly valued goal.

Fruin [18] provides a detailed report of the causes and prevention of crowd disasters. According to Fruin, real-time information and communication are key factors in preventing crowd disasters. He proposed a simple crowd model, know as FIST model, which specifies crowd force, information upon which the crowd acts, physical space, and time (duration of incident).

The FIST model designed by Fruin provides the scientific explanation of the impact of the crowd force.

Fig. 1. (a) Homogenous crowd and (b) aggressive crowd.

"When the density of the crowd increases and becomes equal to plan area of the human body, the group controls the individuals" [18]. When crowd density is about seven persons per square meter, the crowd becomes almost a fluid mass. This fluid mass generates shock waves that propagate through the crowd and are capable of lifting people off their feet. The rise in crowd pressure results in an increase in the temperature and feelings of asphyxia.

The studies in the field of crowd modeling identified the similarity of the characteristics of individuals within the crowd with that of the movements of particles. This comparison resulted in the modeling of the crowd based on particles by Helbing [20]. Helbing identified the key features of escape panics and some of them are as follows: (1) During panic people try to move considerably faster than normal. (2) Pushing among individuals makes the interactions among people become physical in nature. (3) Clogging and jamming effect occurs at the exit. (4) The jamming effect multiplies and results in dangerous pressures of up to 4450 N m^{-1} which can bend steel barriers. (5) Injured people become 'obstacles' in the escape path.

The focus of Helbing's model was on pedestrian dynamics rather than crowd dynamics as a whole. Taking this into account, Hughes [21] modeled a crowd of individuals as a "continuum" with a set of equations to improve the crowd flow of individuals. Helbing's Social Force model [19] incorporates psychological and physical forces. The Social Force model given in Eq. (1) takes into account the pedestrian dynamics based on their motivation, destination, and environmental constraints. v_i^0 indicates the desired velocity of motion, and $\mathbf{e}_i^0(t)$ indicates the direction. $f_{ij}(t)$ and $f_{iw}(t)$ are the interaction forces.

$$m_i \frac{d\mathbf{v}_i}{dt} = m_i \frac{v_i^0(t)\mathbf{e}_i^0 - \mathbf{v}_i(t)}{\tau_i} + \sum_{j(\neq i)} \mathbf{f}_{ij} + \sum_W \mathbf{f}_{iW} \qquad (1)$$

The challenge in performing the analysis of crowd dynamics is the inability to completely grab the real time in-place situation awareness. The studies of crowd behavior is performed based on the crowd disaster video datasets collected at different time spans from different crowd scenarios such as pilgrimage, and sports events. Based on the analysis of crowd movements, they were able to identify three general patterns of crowd flow. They are an ordered pattern (smooth) in normal crowd density and surrounding conditions, the stop and go pattern when the crowd density become dense and obstacles are encountered in the free flow of the crowd, and the extreme flow condition referred as "Crowd Turbulence" due to very high crowd density and critical crowd conditions. This high-density crowd can become "turbulent" and can result in injury or falling.

"Crowd Turbulence" results in displacement of individuals in the crowd in all directions. The crowd flow pattern "turbulence" resembles the fluid flows which show a instability and randomness in the flow pattern. The scientific studies of "Crowd Turbulence" reveal the presence of sharply peaked probability density function of velocity incre-
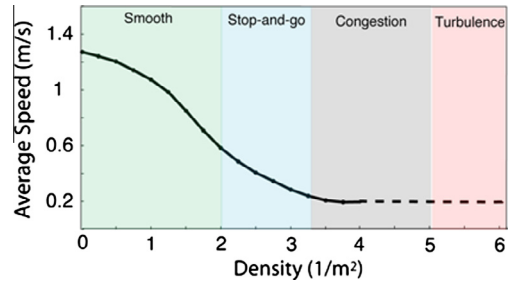

Fig. 2. Flow pattern variation with respect to increased crowd density.

ments V_x, given by Eq. (2), when the value of the time shift τ is small and r represents the location.

$$V_x^\tau = V_x(r, t + \tau) - V_x(r, t) \qquad (2)$$

Based on the video dataset analysis of Mecca pilgrimage, the major conclusions brought in were the crowd accident occurred after 10 min of onset of "Crowd Turbulence", more specifically when the crowd pressure, $P(t)$, exceeded $0.02/s^2$.

Crowd pressure, $P(t)$, is a function of time t, which is obtained by multiplying the density with the variance of the speed. The analysis also highlights that crowd accident occurred 30 min after the change in ordered flow pattern to stop and go. The analysis can be shown graphically in Fig. 2.

The major focus of our research work is to predict the onset of stampede to take safety measures such as crowd flow control, pressure relief strategies, and techniques to block the propagation of crowd pressure generated shockwaves.

4.2. Participatory sensing

Participatory sensing [22] is an emerging technology that makes use of mobile devices to form participatory sensor networks that acquire, process, and share the local information. It supports many applications such as health-care, surveillance, environmental monitoring, and civic management. Even though there are different usage models for different applications, a general workflow in participatory sensing is described in [23].

The essential components of participatory sensing are ubiquitous data capture and leveraged data processing. Ubiquitous data capture in a wide area can be achieved using mobile phones to sense, compute, and communicate the required data effectively. Leveraged data processing can be achieved by using various models and algorithms to analyze the data captured by the smartphone sensors, thus understanding complex phenomena related to the individuals or groups.

This typical capability of participatory sensing is used in the development of multi-context framework has been implemented and tested for crowd monitoring.
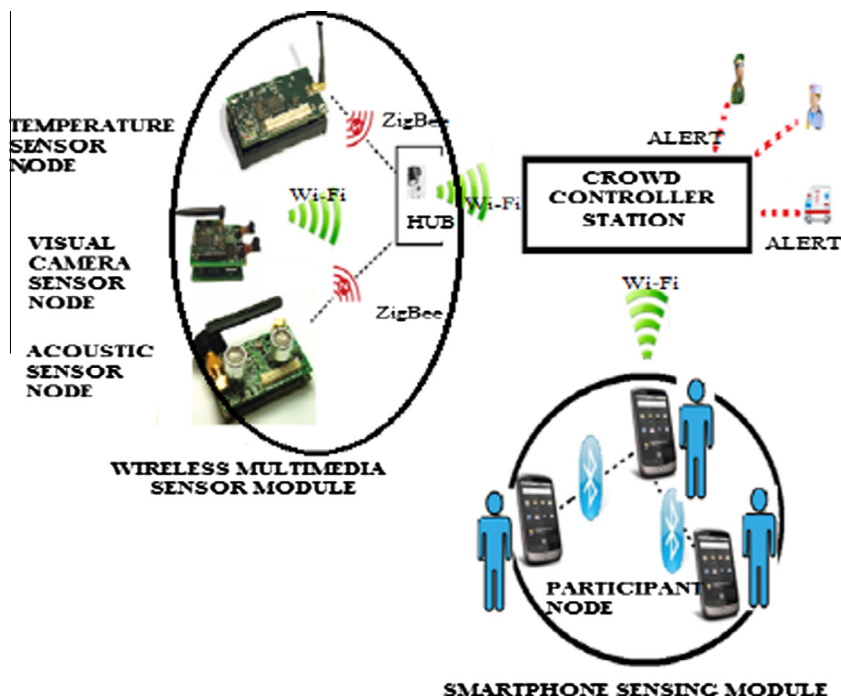
Fig. 3. Participatory sensing based mobile ad hoc network for mitigation of crowd disasters.

4.3. Architecture for mitigation of crowd disasters

In our proposed system shown in Fig. 3, in the area we are crowd monitoring, we assume that we are aware of: the geographical and topographical area, the acceptable crowd capacity of the chosen area, the time of the crowd gathering, and the motivation of the crowd. With these assumptions integrated, the objective of our proposed work is: to estimate the Stampede Occurrence Probability in a specific area, based on correlation of distributed heterogeneous and homogeneous sensor values with known space and time values.

This application explores the use of a participatory sensing approach, where our assumption is that the personal mobile devices and web services are used to systematically explore the key aspects of the current context in a crowd. The personal mobile devices in the crowd act as participant sensor nodes.

For the estimation of the Stampede Occurrence Probability, when the crowd density is 'high' and crowd flow pattern becomes 'stop and go', we perform real-time data mining on heterogeneous and homogeneous sensor values.

Sensor data is analyzed and mapped to activities based on the distributed consensus of 'n' neighboring participant nodes. Sensor data is analyzed to determine the following activities:

(i) Forward force in the crowd towards the common destination.
(ii) Backward force in the crowd against the common destination.
(iii) Crowd force on the sidewalls, or building structures.
(iv) Fall detection in a crowd.

(v) Heat and thermal rise due to increased crowd density.
(vi) High noise level due to screaming.

The correlation of Multiple Datasets from Multiple Sources is required to appropriately estimate the Stampede Occurrence Probability.

An explanation of the system architecture follows:

- The Wireless Multimedia Sensing Module consists of:
  - Temperature sensors – capture heat and thermal rise due to increased crowd density.
  - Acoustic sensors – capture high noise level due to screaming.
  - Visual sensors – capture crowd density and flow pattern estimation.
- The Smartphone Sensing Module consists of:
  - Accelerometer and gyroscope – determine activity recognition and flow pattern inference.
  - Location sensors – provide location context.

4.4. Implementation of Smartphone Sensing Module for crowd disaster mitigation

Participatory sensing using smartphones of the individuals in the crowd act as the key component in the proposed model. The task of estimation of crowd dynamics and estimation of behavior patterns is very challenging to implement in real world situations. But our system makes use of crowd activity recognition chain [28] and distributed consensus of the participant nodes to derive a reliable decision about the onset of the stampede.

The distributed consensus is achieved based on initial individual behavior estimation, then pair-wise Bluetooth ad hoc network based behavior estimation, and finally behavior estimation of the GPS based geographical clusters of participant nodes.

4.4.1. Individual participant node behavior estimation by the participant node

The crowd behavior estimation is carried out by the analysis of sensor data for each participant node individually. Based on the individual sensor data analysis, the "participant behavior" is estimated. The mapping between multi-modal sensors and activities is learned using machine-learning techniques. The individual participant node behavior can be represented as B_{Si}. The block diagram for individual behavior estimation is shown in Fig. 4.

The area of crowd formation is divided into a set of subareas by the crowd controller station. When a participant node joins the crowd, the crowd controller station adds that node into a specific subarea based on its current location. This region identification of a participant node can be performed by using smartphone's GPS, and the steps involved in it are depicted in Algorithm 1. Algorithm 2 explains the steps involved in grouping a participant node into a subarea list. It is necessary to periodically perform a region based grouping algorithm to take into account the mobility of the participant nodes whenever the distance traversed by the participant is greater than a threshold value. This threshold value is around 20 m from the Wi-Fi hotspot for indoor conditions and is of greater value for outdoor conditions. Algorithm 2 also determines the neighboring nodes in a subarea by estimating the distance between two participant nodes. If the distance between them is less than 10 m we add the new participant node to the same area. The 10 m is selected to enable Bluetooth communication in case of an emergency situation. This distance estimation is based on Vincenty's formula using latitude and longitude co-ordinate points.

Algorithm 1 Region Identification of Participant Node, *S* using GPS

1: Begin
2: Input: S; Output: <Latitude, Longitude, Time>
3: while S in power-on state do
4: set locationUpdateWindowSize
5: set minTime i.e. the Location Manager could potentially rest for minTime milliseconds between location updates to conserve power.
6: set minDistance i.e. a location will only be broadcasted if the device moves by minDistance meters.
7: timeDeviation=location.getTime()-currentBestLocation.getTime()
8: if timeDeviation>timeWindow then the user has likely moved and transmit the new location to crowd controller station with timestamp.
9: end while
10: End

Algorithm 2 Region based Grouping of Participant Node, *S* by Crowd Controller Station

1: Begin
2: Input: *SubAreaList={SA₁,SA₂,SA₃...}*, <*Latitude, Longitude, Time*>; Output: *S* grouped in a *SubArea, SA*
3: while *S* in power-on state do
4: for each *SAᵢ* in *SubAreaList* do
   4.1: if <*Latitude, Longitude*> location context matches with *SAᵢ*, group the participant node in *SAᵢ* and incrementCrowdCount$_{SAi}$.
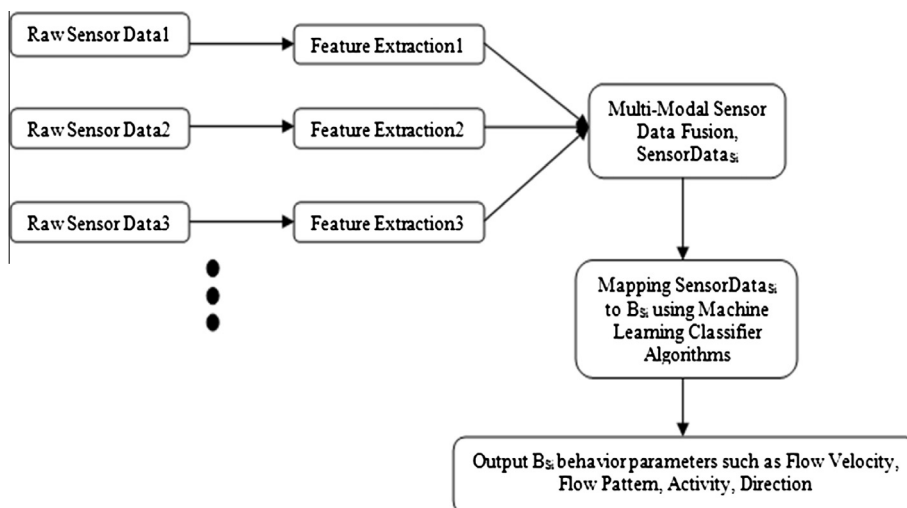5: end for
6: end while
7: End

Fig. 4. Individual participant node behavior estimation.

4.4.1.1. Distance estimation using Vincenty's formulae. Vincenty's formulae [24] are two related iterative methods used in geodesy to calculate the distance between two points on the surface of a spheroid, developed by Vincenty. Vincenty's formula is accurate to within 0.5 mm, on the ellipsoid used.

The key features extracted from individual behavior estimation are activity, flow velocity and flow direction. The flow direction extracted can be used in determining the flow pattern of the crowd.

Algorithm 3 Flow Velocity Estimation and Flow Direction Identification of Participant Node, S

1: Begin
2: Input: <*Latitude, Longitude, Time*>, *x_acceleration, y_acceleration, z_acceleration*; Output: V_{Si} and D_{Si}
3: while *S* in power-on state do
4. Calculate velocity using GPS data as V_{gps}.
  4.1 V_{gps}= *(currentGPSPoint - lastGPSPoint) / (time between GPS points)*
5: Calculate velocity using accelerometer readings as V_{acc}.
  5.1 V_{acc} = *acceleration* ∗ *Time*
6: Obtain V_{Si} from V_{gps} and V_{acc} values
7: Analyze accelerometer and digital compass data for identification of participant flow direction, D_{Si}.
7: end for
8: end while
9: End

4.4.1.2. Feature extraction and activity recognition. Fig. et al. [25] provides the detailed description of different signal processing techniques that can be used for context recognition from accelerometer data. The techniques that can be implemented in mobile devices range from classical signal processing techniques such as FFT to contemporary string-based methods. The paper [25] presented the experimental results to compare and evaluate the accuracy of the various techniques using real data sets collected from daily activities.

(i) Time domain techniques

Simple mathematical and statistical metrics can be used to extract basic signal information from raw sensor data. In addition, these metrics are often used as preprocessing steps for metrics in other domains as a way to select key signal characteristics or features.

(a) Statistical metrics: mean, variance and standard deviation

By determining the mean of a window or range of data samples, meaningful information from almost every kind of sensor can be retrieved. This metric can be calculated with small computational cost and demands minimal memory requirement. The mean is usually applied in order to preprocess raw data by removing random spikes and noise (both mechanical and electrical) from sensor signals,

smoothing the overall dataset. Another important statistical metric is the variance (s^2) defined as the average of the squared differences from the mean. The standard deviation (s) is the square root of the variance and represents both the variability of a data set and a probability distribution. The standard deviation can give an indication of the stability of a signal. These two statistical metrics can be used as an input to a classifier or to threshold-based algorithms. μ indicates the mean, s indicates the standard deviation, and w is the sampling window size.

$$\mu_x = \frac{1}{w}\sum_{i=1}^{w} x_i \tag{3}$$

$$\mu_y = \frac{1}{w}\sum_{i=1}^{w} y_i \tag{4}$$

$$\mu_z = \frac{1}{w}\sum_{i=1}^{w} z_i \tag{5}$$

$$S_x = \sqrt{\frac{\sum_{i=1}^{w}(x_i - \mu_x)^2}{(w-1)}} \tag{6}$$

$$S_y = \sqrt{\frac{\sum_{i=1}^{w}(y_i - \mu_y)^2}{(w-1)}} \tag{7}$$

$$S_z = \sqrt{\frac{\sum_{i=1}^{w}(z_i - \mu_z)^2}{(w-1)}} \tag{8}$$

(b) Root Mean Square (RMS) metric

The feature extraction is performed by taking average of a_j values in the 'w' sized sample window using (1) where *i* can be *x, y* or *z* axis. RMS value estimation can be used to distinguish the walking patterns.

$$a_{average} = \sqrt{\left(\left(\sum_{j=0}^{w}(a_{ij})^2\right)/w\right)} \tag{9}$$

(c) Correlation coefficients

The estimation of correlation coefficients, *r* is useful for distinguishing between activities.

$$r_{xy} = \frac{\sum_{i}^{w}(x_i - \mu_x)(y_i - \mu_y)}{(w-1)s_x s_y} \tag{10}$$

$$r_{xz} = \frac{\sum_{i}^{w}(x_i - \mu_x)(z_i - \mu_y)}{(w-1)s_x s_z} \tag{11}$$

$$r_{yz} = \frac{\sum_{i}^{w}(y_i - \mu_y)(z_i - \mu_z)}{(w-1)s_y s_z} \tag{12}$$

(ii) Frequency domain technique

Fast Fourier Transform (FFT) is a computationally fast alternative for Discrete Fourier Transform. FFT takes a discrete signal in the time domain and transforms that signal into its discrete frequency domain representation. Activity recognition workflow based on FFT coefficients extraction is illustrated in Fig. 5.

4.4.2. Distributed pair wise behavior estimation based on Bluetooth ad hoc networking and geographical clustering based behavior estimation

The block diagram for the distributed behavior estimation is shown in Fig. 6. B_{si} indicates the individual participant node behavior and S indicates the participant node. The behavioral pattern specifically B_{si} changes with time. Hence we need to compute the measure of behavior disparity between a pair of individuals participating in a common crowd. The disparity D is computed based on function f, correlation $Corr$, and g provides the mapping to a disparity value. f, $Corr$ and g are computed based on the training data.

Algorithm 4 Distributed Pair Wise Behavior Estimation

1: Begin
2: Input: B_{si}, B_{si+1}, T; Output: *Distributed Pair Wise Behavior Estimation*
3: while S_i *and* S_{i+1} in power-on state do
4: for each S_i in SA_i do
   4.1: Select the neighboring node of S_i say S_{i+1} and mark them as selected so that they will not be considered for further pair wise analysis.
   4.2: Establish Bluetooth Communication for data exchange between paired participant nodes.
   4.3: Estimate Disparity Matrix for the selected participant nodes, $D_{(u;vT)}$.
     4.3.1: Determine the windowing function, f, *Correlation* and g
     4.3.2: $D_{(u;vT)}$=$g(Correlation(f(B_S; T); f(B_{si+1}; T)))$
     4.3.3: if $D_{(u;vT)}$ has lower values u and v take part in the same crowd.
5: end for
6: end while
7: End

4.4.2.1. Distributed consensus algorithm for stampede prediction in a geographic cluster. We are given with a set of participant nodes or processors P. Consensus is said to be achieved if a good set of processors agree on an outcome regardless of the bad processors. Participant node starts with an initial binary value stored in a variable V. Protocol solves the binary distributed consensus problem, in our case decision of whether there is chance of stampede or not, if it always terminates and satisfies the following two conditions:

Agreement: at termination $Vp = Vq$ for all p and q in P
Validity: if at the beginning of the component $Vp = V$ for every p in P, then the same holds at the end.

- *Step 1*: Initially, set stampede prediction decision value as 0 for all participant nodes.
- *Step 2*: Perform activity recognition and flow pattern extraction. If the condition for stampede depicted in Section 4.1 is satisfied regarding flow pattern variation in a high-density crowd, set the decision value to 1 for the participant group.
- *Step 3*: The leader election algorithm, based on residual battery energy of smart phones, selects the master node. The remaining nodes act as slave nodes. Slave nodes in a Bluetooth ad hoc network establish communication with the master node and transfer the decision value to it.
- *Step 4*: When a set of participant nodes agrees on the decision value as 1, send an alert message to the crowd controller station to generate appropriate control measures.

5. Experimentation and result analysis

We illustrate the proposed context aware ad hoc network for mitigation of crowd disasters by detecting
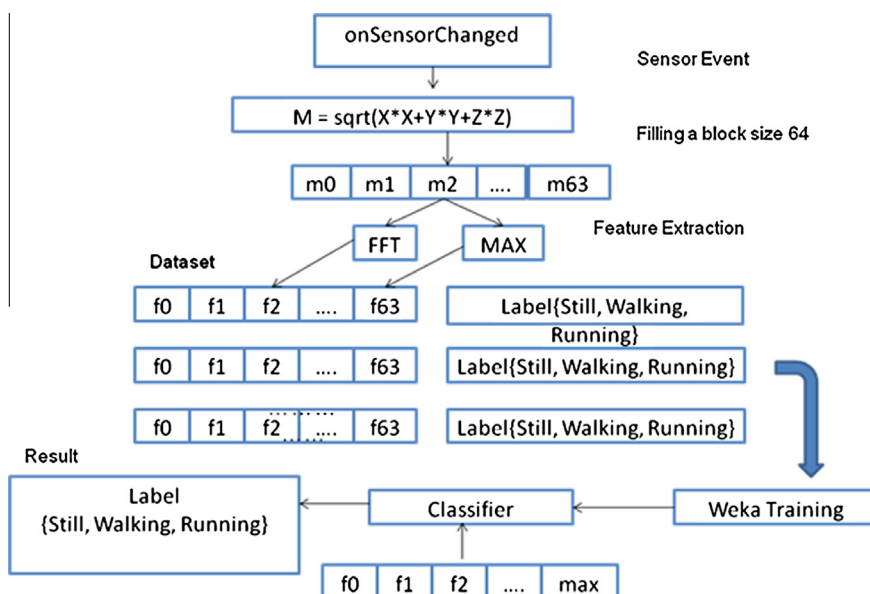

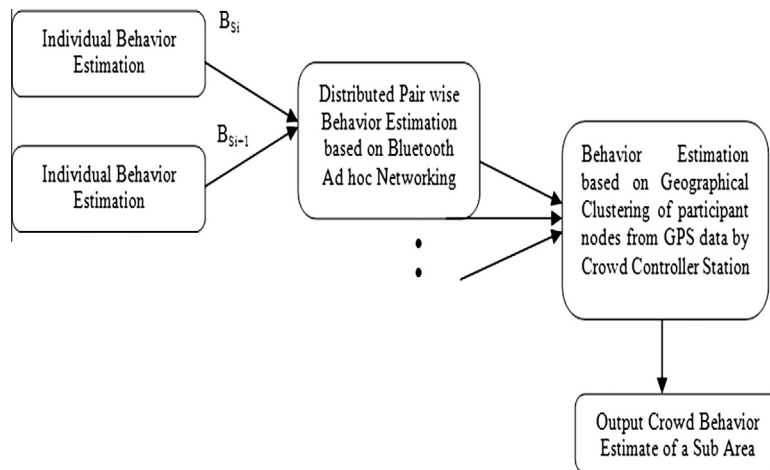**Fig. 5.** Activity recognition workflow.

Fig. 6. Distributed behavior estimation.

abnormal human movements, specifically erratic shake and push, in a group of participants moving through a narrow pathway. Based on the analysis of crowd flow patterns, for the purpose of experimentation, the group of participants followed an ordered flow pattern initially. Later on, due to insufficient space for free movement when they reached the narrow pathway, pushing and leaning of people against each other resulted in the development of crowd forces, which are collected by using accelerometers.

The push and erratic shake experienced by the group is extracted by the signal processing of raw tri-axis accelerometers embedded in the smartphones. The decision on the prediction of stampede is based on crowd activity recognition chain and distributed consensus algorithm. For crowd behavior estimation, it is important to estimate individual behavior initially. Hence our experimentation focuses on individual activity recognition, ad hoc networking for information spreading and distributed consensus for prediction of stampede.

5.1. Experimental setup

5.1.1. Activity recognition of an individual
5.1.1.1. Data acquisition. Activity recognition phase required data acquisition from raw accelerometer sensor, embedded in the smartphone, to use as datasets for training and testing. The smartphones used for data acquisition are HTC Nexus One, Samsung Galaxy Y and Samsung Galaxy Ace. The activities under consideration are:

  (a) Standing.
  (b) Walking.
  (c) Slow running/jogging.
  (d) Climbing up.
  (e) Climbing down.
  (f) Fall.
  (g) Peak shake while standing.

 Based on the result of Bao and Intille [26] research work, we were able to identify that the thigh is the most

suitable position for the placement of accelerometers for identifying various movements. Hence our experimentation with smartphone was carried out by placing the phone in the pant pocket. Twenty participants spent variable time for each activity at their convenience so the data could be acquired.

We created an Android application with a graphical user interface, shown in Fig. 7, to select, start and stop control options for each activity. The collected data was written to the smartphone's SD card in Attribute Relation File Format (ARFF). The application also graphed the sampled acceleration data, shown in Fig. 8, allowing for a real-time preview of the data. Android SDK supports four abstract sampling rates Fastest, Normal, Game and UI [8]. The Android framework allows the acquisition of accelerometer samples when OnSensorChanged() event gets triggered. Following the sampling rates provided by Android framework, we could collect samples approximately every 50 ms, i.e. 20 samples per second.

5.1.1.2. Signal processing and feature extraction. The raw values obtained from the smartphone's accelerometer are preprocessed prior to feature extraction. For the performing preprocessing of data, we have used a window size of 256 with 50% overlap. Each of the three axes is analyzed individually and the statistical metrics such as mean, standard deviation and correlations were obtained. The significance of these statistical metrics is specified in Section 4.5.1.

The feature extraction values for various activities such as walking, slow running, and falling is collected at the crowd controller station. Activity recognition can be considered as a classification problem and hence we can make use of the potentials of machine learning algorithms. Machine learning algorithms are used to make decisions based on the gathered information from sensors to make the initial scientific hypothesis. In order to classify the human activities in a crowded area, decision tree based machine learning algorithm was used.

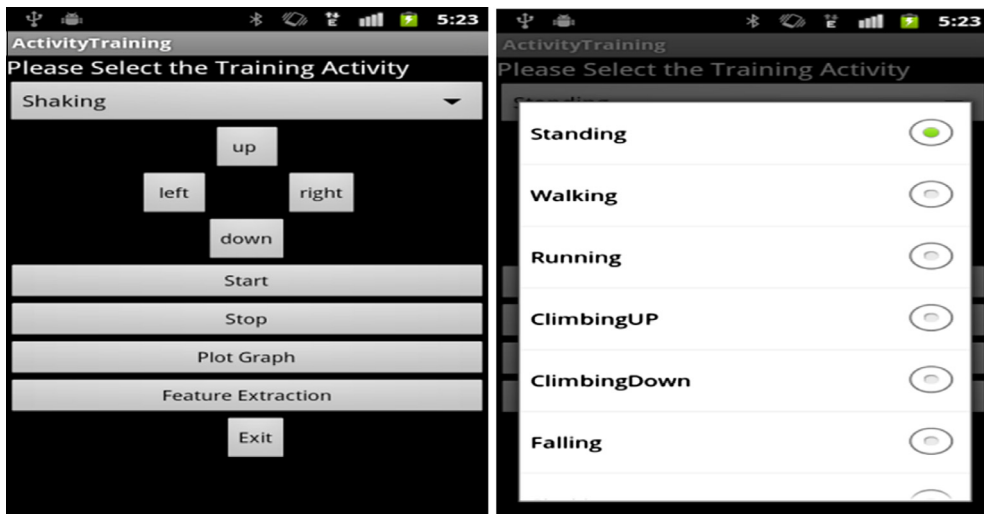C4.5 generates decision trees from a set of data obtained from the training phase. These training data set

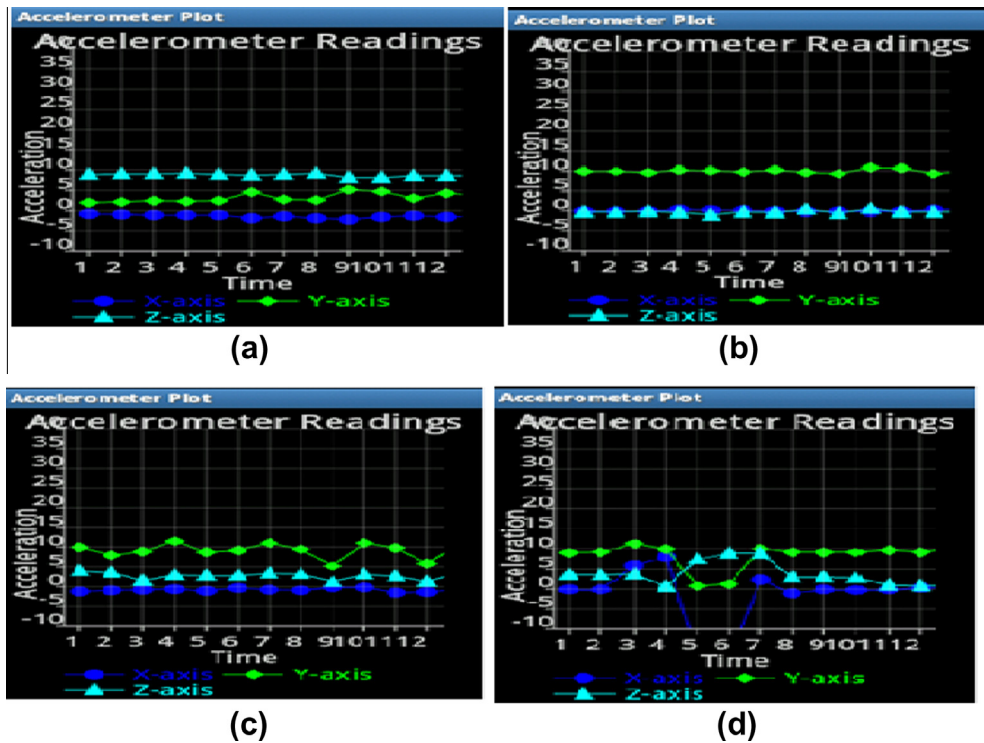**Fig. 7.** User interface for activity training and accelerometer based analysis.

Fig. 8. Accelerometer plot (a) standing, (b) forward walk, (c) slow-run, and (d) peak shake.

contain samples which are features extracted. These features are augmented with a specific vector to indicate to which class it belongs to C4.5 chooses one key feature from the set to split it into subsets. The feature with highest normalized information gain is selected to make the decision. For the purpose of analysis of the performance of various machine learning algorithms, we make use of the Weka toolkit. Weka [27] is data mining software which provides a collection of machine learning algorithms. J48 is

the implementation of C4.5 algorithm in Weka. For the activity classification, Fast Fourier Transform values were also obtained. The method of extraction is detailed in Fig. 5.

5.1.2. Prediction of stampede in a group

Based on the assumption of known geographical area specifically an open space for crowd gathering, the raw values were obtained from the participant smartphones. Several repeated experiments with change of locations were

Fig. 9. Group of participants for the simulation of clogging in a narrow pathway.

Fig. 10. Real time detection of clogging.

conducted to gather training data as well as decision threshold. The experiments were conducted in the University premises with sample size of 20 smartphones. For the purpose of deciding on the decision threshold, a simulation of clogging in a narrow pathway inside the university building was conducted, shown in Fig. 9. The participants were instrumented with smartphones in their thighs. The group was instructed to follow an ordered flow pattern before reaching the narrow pathway.

Later on, in order to check the correctness of the decision threshold, the experiment was conducted in a real time scenario. The participants with smartphones installed with our Crowd Abnormality Monitor (CAM) application formed the part of a group of people moving out of narrow gateway at peak rush time shown in Fig. 10.
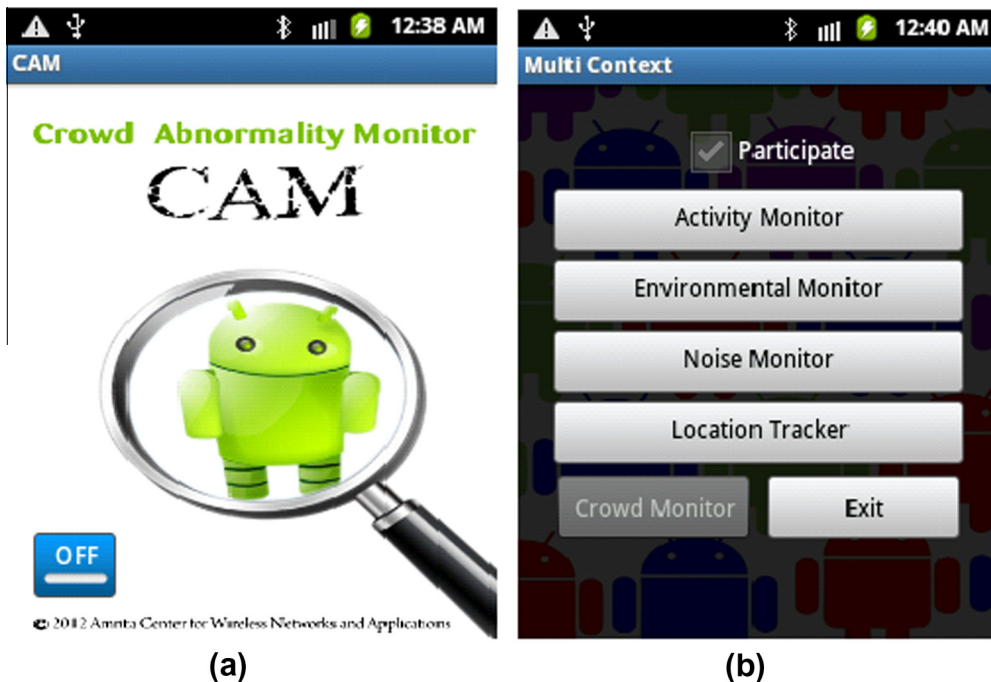
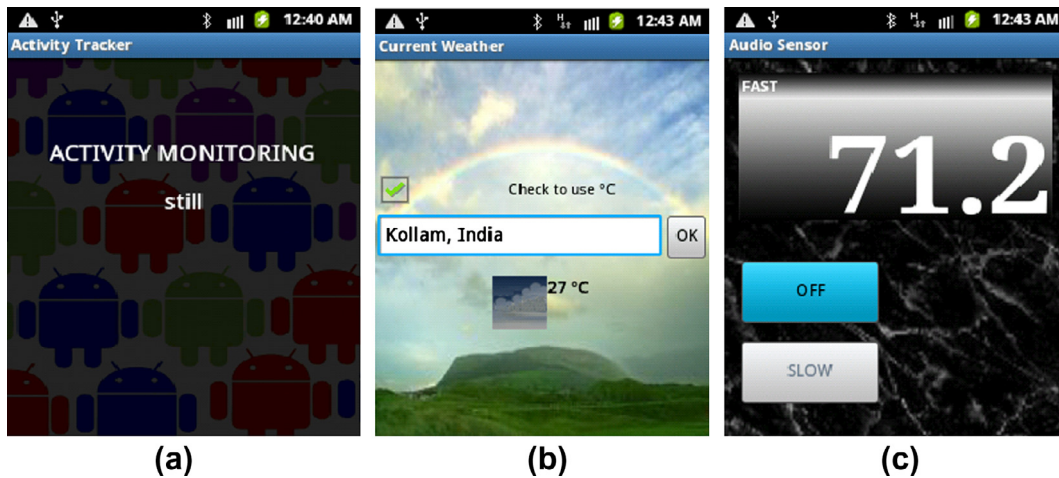**Fig. 11.** (a) CAM launch window and (b) multi-context framework.

Fig. 12. (a) Activity monitoring, (b) environmental monitoring, and (c) noise monitor.

5.2. Results and validation

5.2.1. Crowd Abnormality Monitor (CAM) on Android platform

An application in Android framework was developed for crowd monitoring, named as Crowd Abnormality Monitor (CAM). The CAM application facilitates people in the crowd to participate in crowd monitoring by registering with the crowd control station. On joining the crowd monitoring group, the information regarding the participant such as the location context, unique MAC address and the activity are communicated to the crowd control station. Based on the location context, the crowd control station performs the region identification and grouping algorithm. The situational awareness is generated by extracting the context data such as environmental noise level in the Sound Pressure Level (SPL), activity tracker, location tracker, and environmental monitor using the multi-context fusion framework, which is shown in Figs. 11 and 12.

5.2.1.1. Multi-context fusion procedure.
smartphones are equipped with a wide variety of sensors. Let these sensor set be represented as S, where $S = \{S_i\}$, S_i's are individual sensors such as audio, location, and light sensor. Our application makes use of say n, number of sensors from S for making the critical decisions.

- *Step 1*: For a particular time period, the basic levels of crowd monitoring sensors are activated and the values are extracted. The basic level set id obtained based on the battery usage.
- *Step 2*: When the basic level threshold is exceeded, trigger the activation of next level of sensors which are higher in battery consumption.
- *Step 3*: Fuse the extracted sensor data from multiple specific sensors to make critical decisions. Generate alert according to the critical threshold.

5.2.1.2. Crowd monitoring module.
The user of CAM application can decide whether to join crowd monitoring by

Fig. 13. Crowd monitor in listening mode and monitor status.

checking the participate checkbox. When the participant joins crowd monitoring, the application registers with crowd controller station and switches to listening mode. The crowd monitoring module is shown in Fig. 13.

5.2.2. Activity recognition of an individual

5.2.2.1. Performance analysis using J48 machine learning algorithm.
The trained dataset collected by using data acquisition module is analyzed by using the Weka machine learning tool. Tables 2 and 3 show the results obtained as part of data analysis. The J48 based classification was performed and a decision threshold for individual activity recognition was determined. True Positive Rate (TP rate) indicates the rate of predicted correctly to total possibilities and False Positive Rate (FP rate) indicates the vice versa. Recall and precision are the parameters for indicating relevance. F-measure is the harmonic mean of precision and recall. ROC area gives the performance of a system as the threshold is

Table 2
Performance analysis of activity recognition.

Class	TP rate	FP rate	Precision	Recall	F-measure	ROC area
Still	0.965	0.042	0.94	0.965	0.952	0.968
Walking	0.903	0.049	0.903	0.903	0.903	0.948
Running	0.889	0.024	0.928	0.889	0.908	0.949

Table 3
Confusion matrix for activity recognition.

Still	Walking	Running
109	3	1
5	84	4
2	6	64

varied. The area under the ROC indicates the accuracy. Kappa statistic is a measure of agreement.

- Correctly Classified Instances: 92.446%.
- Incorrectly Classified Instances: 7.554%.
- Kappa statistic: 0.8845.
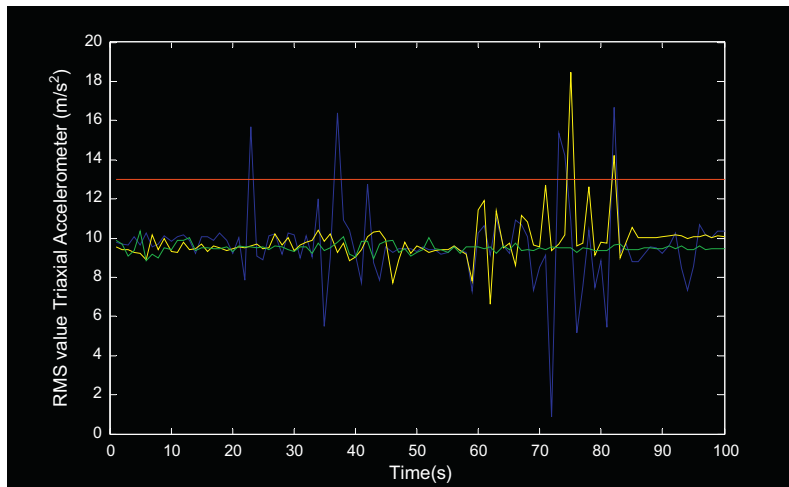
Fig. 14. Accelerometer signal analysis from clogging simulation.
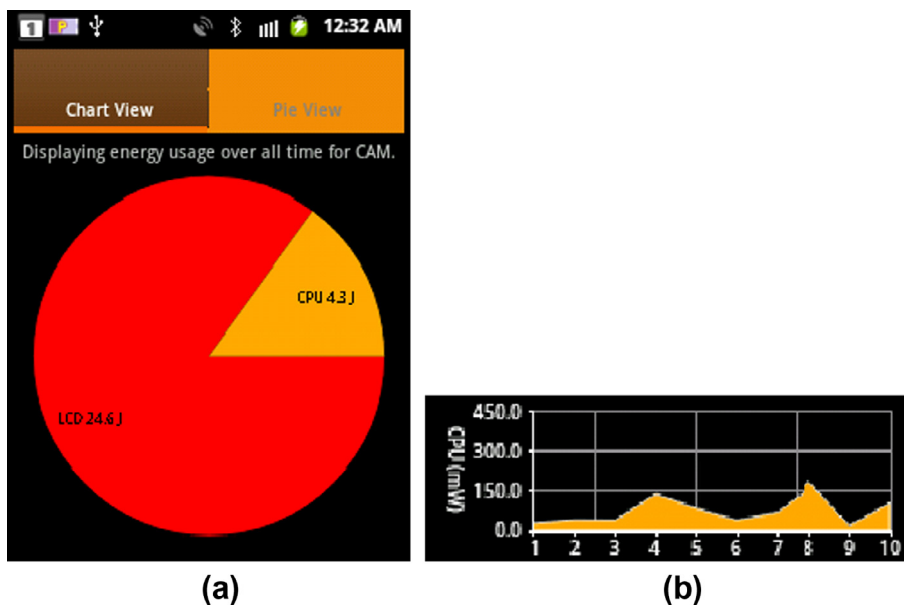
Fig. 15. Energy estimation (a) energy utilization by CAM application and (b) CPU utilization of the entire system when CAM application is active.

- Mean absolute error: 0.0569.
- Root mean squared error: 0.2169.
- Relative absolute error: 13.0096%.
- Root relative squared error: 46.3803%.
- Coverage of cases (0.95 level): 93.1655%.
- Mean rel. region size (0.95 level): 33.5731%.

The major criteria for the detection of fall is when the lower threshold value of tri-axis acceleration vector sum is approximately around 5 m/s^2 and the upper threshold value greater than or equal to 20 m/s^2.

5.2.3. Performance analysis and validation of stampede prediction algorithm

Signal processing on raw accelerometer data obtained from simulation of the clogging effect in a narrow pathway was performed to determine the decision threshold value. The Root Mean Square (RMS) value of the accelerometer values obtained by performing three different simulations is shown in the figure. Based on the analysis of the RMS values, we were able to take 13 m/s^2 as the abnormality detection threshold (shown as red[1] line in Fig. 14) for our distributed consensus algorithm, i.e. a prediction alert for stampede (in the simulation purpose clogging effect) occurs when all the participant nodes the group exceeds the specified threshold value. The decision threshold was set in the Android application according to the analysis and was performed in a real time scenario for clogging detection and alert generation.

5.2.4. Energy based cost evaluation of the system

The energy of the proposed framework, CAM, is estimated by making use of PowerTutor application [29]. The battery usage of the application including the usage of CPU and communication cost is approximately 28.7 J. The analysis of energy and power are shown in Fig. 15.

6. Conclusion and future work

This research work has designed a mobile sensor network system integrated with wireless multimedia sensor networks (WMS) for effective prediction of a stampede for crowd disaster mitigation. This resourceful innovation, with an eye on the future for other applications, designed a multi context aware framework and made use of the prolific ownership of smartphones. For the implementation of a crowd monitoring system, we developed and evaluated individual behavior estimation, distributed behavior estimation, and a distributed consensus algorithm for stampede prediction. This work has developed an Android application for capturing real time dynamics of crowd behavior, space configuration, space capacity, traffic processing capabilities, etc. Our activity recognition algorithm has an accuracy of 92%, and the stampede prediction algorithm makes use of this activity recognition algorithm along with individual behavior estimation.

Congestion, one of the stampede aspects, was experienced in the university building and was used to determine the decision threshold. The Crowd Abnormality Monitor (CAM) application was developed based on this threshold. The CAM was tested in a real time situation, and the stampede prediction alert was generated.

Our research work predicts the onset of a stampede in order to take safety measures such as crowd flow control, pressure relief strategies, and techniques to block the propagation of crowd pressure shockwaves.

In the future, we plan to extend this research to integrate multimedia processing with mobile phone sensing for better accuracy of stampede detection. Since modeling of a stampede using mobile phones is a new area, more research work needs to be brought in for crowd force estimation and modeling using mobile phones for stampede prediction.

The ultimate objective of this research work is to develop a non-redundant and data sharing context aware framework for multiple applications including healthcare, traffic monitoring, etc.

References

[1] Bill Schilit, Norman Adams, Roy Want, Context aware computing applications, in: Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, California, December 1994, pp. 85–90.
[2] Bill Schilit, Marvin Theimer, Disseminating active map information to mobile hosts, IEEE Network 8 (5) (1994) 22–32.
[3] Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, Walter Van de Velde, Advanced interaction in context, in: Proceedings of First International Symposium on Handheld and Ubiquitous Computing, Karlsruhe, Germany, September 1999, pp. 89–101.
[4] Guanling Chen, David Kotz, A survey of contextaware mobile computing research, Technical Report TR2000-381, Computer Science Department, Dartmouth College, Hanover, New Hampshire, November 2000.
[5] Anind K. Dey, Understanding and using context, Personal and Ubiquitous Computing 5 (1) (2001) 4–7.
[6] J. Pascoe, N. Ryan, D. Morse, Using while moving: HCI issues in fieldwork, ACM Transactions on Computer–Human Interaction (TOCHI) 7 (3) (2000) 417–437.
[7] S.P. Hall, E. Anderson, Operating systems for mobile computing, Journal for Computing in Small Colleges 25 (2009) 64–71.
[8] Google, Android Developers, 2010. <http://developer.android.com/>.
[9] Safer Crowds, 2012. <http://www.safercrowds.com/>.
[10] JagdishRebello, India Cell Phone Penetration to Reach 97% in 2012. <http://www.isuppli.com/Mobile-and-Wireless-Communications/News/Pages/India-Cell-Phone-Penetration-to-Reach-97-Percent-in-2014.aspx>.
[11] Anthony C. Davies, Jia Hong Yin, Sergio A. Velastin, Crowd monitoring using image processing, IEE Electronic and Communications Engineering Journal 7 (1) (1995) 37–47.
[12] L. Gomez, A. Laube, C. Ulmer, Secure sensor networks for public safety command and control system, IEEE Technologies for Homeland Security (2009).

[13] Jennifer R. Kwapisz, Gary M. Weiss, Samuel A. Moore, Activity recognition using cell phone accelerometers, ACM SensorKDD, Washington, DC, USA, 2010.

[14] T. Ryan Burchfield, S. Venkatesan, Accelerometer based Human Abnormal Movement Detection in Wireless Sensor Networks, 2006.

[15] J.C. Puras, C.A. Iglesias, Disasters2.0 Application of Web2.0 Technologies in Emergency Situations.

[16] Y. Nakajima, H. Shiina, S. Yamane, T. Ishida, H. Yamaki, Disaster evacuation guide: using a massively multiagent server and GPS mobile phones, Applications and the Internet (2007).

[17] Martin Wirz, Daniel Roggen, Gerhard Tröster, Decentralized detection of group formations from wearable acceleration sensors, CSE (4) (2009).

[18] John J. Fruin, The causes and prevention of crowd disasters, in: First International Conference on Engineering for Crowd Safety, 2002.

[19] D. Helbing, P. Mulnar, Social force model for pedestrian dynamics, Physical Review E 51 (1995) 4284–4286.

[20] D. Helbing, I. Farkas, T. Vicsek, Simulating dynamical features of escape panic, Nature 407 (2000) 487–490.

[21] R.L. Hughes, The flow of human crowds, Annual Reviews of Fluid Mechanics 35 (2003) 169–183.

[22] J. Burke, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, M.B. Srivastava, Participatory sensing, in: WSW'06 at SenSys, ACM, Boulder, 2006.

[23] Deborah Estrin, Participatory sensing: applications and architecture, IEEE Internet Computing, 2010. <www.computer.org/Internet>.

[24] T. Vincenty, Direct and Inverse Solutions of Geodesics on the Ellipsoid with application of nested equations, April 1975.

[25] Davide Figo, Pedro C. Diniz, Diogo R. Ferreira, Joao M.P. Cardoso, Preprocessing techniques for context recognition from accelerometer data, Personal and Ubiquitous Computing 14 (7) (2010) 645–662.

[26] L. Bao, S. Intille, Activity recognition from user-annotated acceleration data, Lecture Notes Computer Science 3001 (2004) 1–17.

[27] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, second ed., Morgan Kaufmann, 2005.

[28] Martin Wirz, Daniel Roggen, Gerhard Troster, Recognition of crowd behavior from mobile sensors with pattern analysis and graph clustering methods, Networks and Heterogeneous Media 6 (2011).

[29] PowerTutor, 2012. <http://ziyang.eecs.umich.edu/projects/powertutor/>.

[30] James J. Park, Jong Hyuk Park. Embedded and Multimedia Computing Technology and Service: Emc 2012, Springer, 2012.

[31] Maneesha V. Ramesh, Design, development, and deployment of a wireless sensor network for detection of landslides, Ad Hoc Networks, Elsevier, 2012, accepted for publication.

Anjitha Shanmughan is a Master's student in AMRITA Center for Wireless Networks & Applications, Amrita University. She holds a Bachelors' degree in Computer Science and Engineering. Her research interests are wireless sensor networks, context aware computing and crowd dynamics modeling.

Rekha Prabha is a research associate in AMRITA Center for Wireless Networks & Applications, Amrita University. She is pursuing PhD in the area of heterogeneous wireless networks. She holds a Masters degree in Applied Electronics and posse's industrial experience in product development, testing and documentation activities in the embedded domain. She was involved in the project WINSOC, India's first ever wireless sensor network to detect rain induced landslides. Her research interests are on wireless networks, energy optimization, and embedded systems. Her particular interest is on overlay communication and self-organized wireless networks.

Maneesha V. Ramesh is the Director & Associate Professor of Amrita Centre for Wireless Network and Applications (www.amrita.edu/awna) and is the Director of Amrita Center for International Programs in the Amritapuri Campus of Amrita University, India. She received her PhD in Computer Science from Amrita University on "Wireless Sensor Network to detect rainfall induced landslides". She is the pioneer in building up and launching, India's first ever landslide detection through WINSOC', the European funded project, in which she was the Principal Investigator. Data is continuously real streamed to the web page www.winsoc.org. The project had issued a real time warning in July 2009 for a potential landslide in Kerala, Southern India. The landslide system won the rural innovation award from NABARD, Govt. of India. Her research areas include wireless sensor networks for disaster management, wireless networks for bio-nano-medical applications, wireless sensor network algorithms, wireless network design and development, context aware systems, participatory sensing, mobile computing, smart grid etc.