Group Identification in Crowded Environments Using Proximity Sensing

Shaden Shaar
and Saquib Razak
Carnegie Mellon University in Qatar
Email: {sshaar,srazak}@cmu.edu

Fahim Dalvi
Computer Science
Stanford University
Email: fdalvi@cs.stanford.edu

Syed Ali Hashim Moosavi
College of Computing
Georgia Institute of Technology
Email: hashim.moosavi@gatech.edu

***Abstract*—Children and elderly separating from their family members is a common phenomenon, especially in crowded environments. In order to avoid this problem, places like Disney World and pilgrimage officials have developed systems like wearable tags to determine groups or families. These tags require information about families to be entered manually, either by the users or the facility organizers. The information, if correct, can then be used to help identify and locate a lost person's group. Manually entering information is inefficient, and usually leads to either long waiting times during entry, or partial information entry within the tags. In this paper, we propose a system that uses proximity sensing to determine groups and families without any input or interaction with the user. In our system, each user is given a wearable device that keeps track of it's neighbors using bluetooth transmissions. The system then uses this proximity data to predict cliques that represent family members.**

I. Introduction

One of the biggest fears of parents at crowded places like airports, malls, theme parks, etc. is to lose sight of one of their children. Children at these places are easily distracted due to the numerous sights around them and may not realize that they have been separated from their families. The problem is not limited to children, as older people also get disoriented and lose track of their families. As a result, many families at crowded places end up looking for lost family members. The time this process takes can range from few minutes to several hours of agony depending on the size of the facility and it's crowd's density.

According to International Association of Amusement Parks and Attractions (IAAPA), there are more than 400 amusement parks in the United States alone, and in 2010, more than 290 million people visited these theme parks [1]. With this many people visiting, the likelihood of people getting separated from their families is high. According to parentguide, 2000 children get lost every day for some period of time at places like beaches, malls, airports, and amusement parks [2].

Annual religious gatherings and sports event are another place where people get separated. At the Hindu festival of Kumbh Mela and Muslim annual pilgrimage to Makkah, millions of people attend several days of worship within a wide area [3] [4]. During these events, several thousand people are reported missing, adding to operational cost for the organizers and grief to the attendees [4] [5]. While some of these people are reunited with their families within a few hours, several of them stay missing for days [4]. The diverse backgrounds of the people attending these events adds to the problem.

When a lost child is discovered in a facility, there are two problems that need to be solved: somehow, determine who are the family members of the child and then, locate these family members. Existing manual or technological solutions to the problem of identifying family members of a lost child rely on either, some kind of information entry, or some level of technical expertise and awareness from the users. These solutions lack answers to the following challenges:

- Users or organizers cannot spend time entering information about individuals on a device. Any such entry causes a barrier to entry resulting in long waiting lines.
- Manual data entry by individuals is error prone resulting in inaccurate or incomplete information.
- People visiting these places, belong to diverse backgrounds and a large number of them have no experience with technology.
- Use of smart phones, although very effective, is cost prohibitive for larger families in terms of cost of equipment, as well as, cost of service.

In this paper, we propose a system that uses proximity data to help identify members of a family or group without any technical expertise or input from the user. As people enter into a facility, they are handed a wearable bracelet. Except wearing this bracelet, users do not need to perform any other interaction with the device. Each device transmits a periodic heartbeat message. Devices hearing this message keep track of their neighbors during their lifetime. At any given time, a device is able to identify its neighbors (family). Our hypothesis is that frequency of interactions, amount of time spent together, and chronology of spent time are good indicators for predicting families - even when family members get separated. Once we have identified the devices that belong to a person's family members, the second problem is to locate these devices within the service area. In this paper, we will only attempt solving the first problem. We are not concerned with solving the second problem since there are many existing localization solutions that use bluetooth and ad-hoc networks as discussed in section II.

We show the effectiveness of our technique by simulating people movement in different environments like malls and bookfairs. We show that our algorithms are able to correctly predict 92% of the families when there is a loss rate of 25% and 96% with a loss rate of 10% . For lower loss rates, our success rates are almost 100%.

II. Related Work

For outdoor environments, GPS based solutions provide accurate localization of devices embedded within family members' clothing or smart-devices they might carry with them [6], [7]. Smartphone applications allow you to track the location and movement of these family members. These solutions require each family member to own a device that might be cost prohibitive and inconvenient to carry - especially for small children.

Other solutions employ Personal Area Communications (PAN) devices like RFIDs and Bluetooth peripherals. These devices alert the parents if their child is not in their close proximity [8]. Since this solution provides very coarse localization, recent work has extended these solutions by using crowd sourced collaborative localization to obtain accurate location information [6].

Techniques for activity and face recognition are also considered to help find lost people in crowds. By analyzing the accelerometer data and then classifying their activity type, high accuracy was achieved when identifying lost person behaviour [9]. Extraction of facial features from stored images and subsequent comparison with the images from CCTV cameras is also another solution used to locate lost people in a crowded area [10]. This solution requires a huge infrastructure setup of multiple cameras all around the facility.

Independent of the actual method of detecting proximity, work has also been done in improving the detection and clustering of groups in large crowds. Although the definition of a group varies across different studies, the techniques themselves lend naturally to identifying a set of people with some defined commonality [11].

III. System overview

We start this section with what we imagine to be a typical use case of our proposed system. We imagine that administrators at crowded areas like malls and theme parks, have a desire to provide lost and found service to their visitors. As customers walk into a facility, they are handed a device that they will keep on their persons. These devices need no data entry from the customers when they receive them. We argue that any form of data entry at entrance of facilities, no matter how trivial, will cause long waiting lines and result adverse customer experience. The devices are equipped with Bluetooth low energy (BLE) modules that transmit an "I am here" message periodically. The use of BLE helps us in minimizing the power consumption. The range of a typical Bluetooth device is around 10 meters [12]. This limited range helps in ignoring devices that are not within our immediate proximity. Based on this range, any device that is within this area will receive the message from its neighbors. When a device receives a message from a neighbor, it notes the hardware id of the device and saves it internally.

As the devices borne by their carriers move around the facility, they come in contact (within reception range) of other devices and record their presence. This way, a device collects information about all the other devices it has come in contact with. The device also records the frequency and duration of each contact with other devices.

IV. System Design

In this section, we describe the information our system learns over time and the algorithms we use to define groups. A *node* refers to an individual device, capable of recording some information as it comes in proximity of other nodes. All of the information is updated in units of a cycle, which is set to some length of time. The information for neighboring nodes is collected by each node as it moves around in the facility, coming in contact with several other devices. The set of nodes belonging to one family are in general together more often and for a longer period of time. When a device is lost (a user bearing a device is lost), it will come in random contact with other devices. The behavior of these contacts is very different than the usual behavior around a node's family, and our system exploits this phenomenon to predict the families of a node.

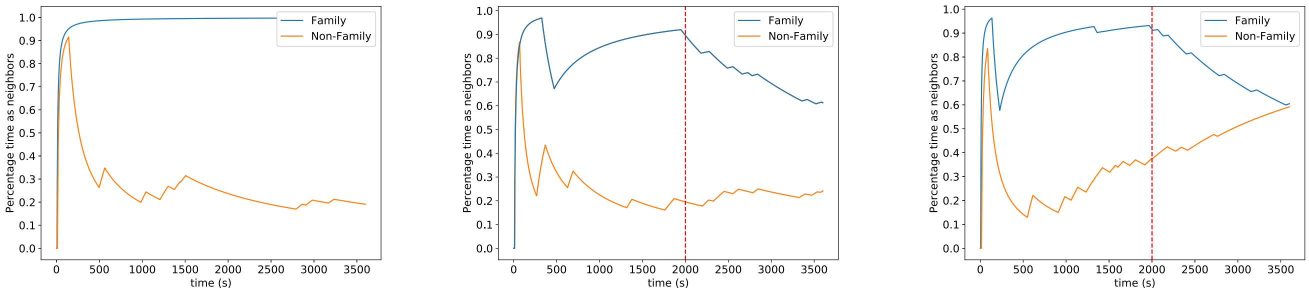
Let us first consider three characteristic scenarios which our nodes can encounter:

1) Figure 1a shows the first scenario, i.e. the interaction of a *node* with a *family node* compared to a *non-family node*. The graph shows the percentage of time the *node* has seen the other two nodes. We verify our intuition that if a *node* spends substantial amount of time with specific nodes, we can conclude that these nodes must be members of the same family.
2) In the second scenario, as shown in figure 1b, where a *node* gets separated from its family after several minutes and then roams around freely. We see that the while in contact with the family, Node 1 had a higher percentage of time as neighbors with the family member. Once a node gets separated from its family, it has the same interaction behavior as with non-family members.
3) The third scenario is when a *node* gets separated from its family and it adopts another family. In this case, as shown in figure 1c, after some period of time, the total time spent for the two categories of nodes is similar, which poses a challenge for identifying families.

We now present several criteria we used to identify the group a *node* belongs to. In section V, we present the accuracy of each criteria.

A. Time Duration:

Each node keeps track of the amount of time it has seen a specific node. Each time a message is received, the node updates the timer for the transmitter. If a transmission is missed from a specific node for less than *maxMissedCycles* (which we set to 5 based on experimental evaluation), the node is still considered a neighbor. Any subsequent transmission



(a) Total Time spend together for a family node compared to a non-family node.

(b) Percentage Time spend together with a family node compared to a non-family node. The node loses contact with the family after 2000 seconds.

(c) Percentage Time spend together with a family node compared to a non-family node. The node loses contact with its own family and starts following the non-family node.

Fig. 1: Percentage time spend in various scenarios

will consider the missed cycles to be part of the *Time Duration*. If a node is out of range for longer, the intermediate cycles will not be counted.

B. Time Duration and Frequency:

The second criteria is to consider time duration and frequency together. The frequency of a neighbor is incremented each time a node receives a message from this neighbor. Since this is a monotonically increasing function, we normalize it for comparison purposes. We combine the two metrics to get a new $score_i (i$ stands for nodeid of the neighbor) as shown in Eqn 1.

$$score_i = \alpha_1 * \frac{freq_i}{\max_{i \in n} f_i} + \alpha_2 * \frac{Duration_i}{Totaltime} \tag{1}$$

$$\text{where } n \in N, \text{ the number of } \textit{nodes}$$

$$\alpha_1 \text{ and } \alpha_2 \text{ are weights assigned to each factor}$$

C. Area under the curve:

Our earlier observation of node behavior in figure 1c, indicates that earlier interactions with nodes should also be part of evaluation criteria under certain circumstances. In order to account for this, we observe that the area under the curve for percentage of time together is substantially higher for family nodes than those of non-family ones. We use this as an additional criteria, extending equation 1 to calculate the new value of score:

$$\begin{aligned} score_i = \alpha_1 * \frac{freq_i}{\max_{i \in n} f_i} + \alpha_2 * \frac{Duration_i}{Totaltime} \\ + \alpha_3 \int_0^t \frac{Duration_i}{Totaltime} dt \end{aligned} \tag{2}$$

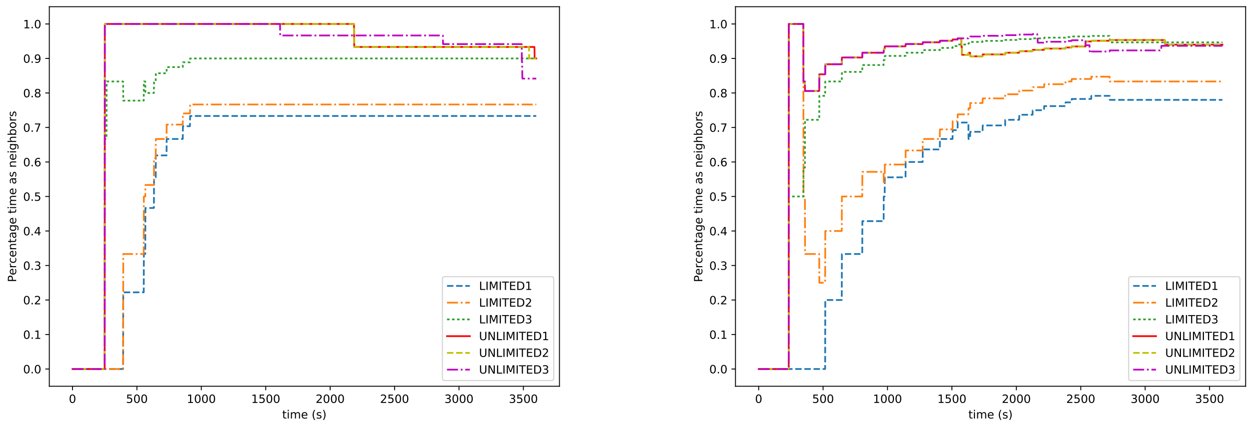
D. Memory Limitations

The above algorithms assume that each node has complete knowledge of all interactions from the beginning of time. Given that small wearable devices have limited memory, keeping record of every interaction with all nodes is not a reasonable expectation. We extend our existing techniques to algorithms that keep track of a limited number of nodes throughout their lifetime. Here, we introduce the concept of *buckets*, where each *bucket* can store information for a single *node*. The number of *buckets* is based on the available memory on the device itself. These buckets are then divided into multiple levels, with higher levels having a higher retention ratio. Hence, moving in to and out of higher levels would require more concrete evidence of particular *nodes* belonging to the same group. In our algorithms, we divide the buckets into two levels. The higher level only holds predicted group members, while the lower level holds potential group members. We propose the following algorithms that define different criteria for evicting a node from our list or updating a node to be a neighbor in our list.

- $\texttt{LIMITED}_1$: Evict a node with the oldest timestamp. Upgrade a node in the list to be a neighbor if its total time is double the total time of any node in a higher bucket.
- $\texttt{LIMITED}_2$: The eviction function depends on eq 1. The lowest score node is evicted only if the score is below a threshold. When the score of a node in lower level bucket exceeds a node in higher level by a certain threshold, the nodes in the two buckets are swapped.
- $\texttt{LIMITED}_3$: Similar to $\texttt{LIMITED}_2$ but with eq 2.

V. Evaluation

We evaluate the performance of our algorithms using multiple realistic scenarios. Due to space limitations, we only present the evaluation inside a mall. For a mall, we mark points of interest - like entrances to shops and locations inside shops. Families of sizes 3-6 are generated at the entrances of the mall and each family then picks a random point of interest within the mall and walks there. Each family moves within the area using Reference Point Group Mobility Model (RPGM) [13] along with Random Waypoint mobility model [14]. Each family is generated at an entrance to the mall. Each family of nodes is represented by a ”logical center”. During simulation, this center picks a point of interest as destination within the



(a) High density of crowds when 10% of families lose a person (b) High density of crowds when 25% of families lose a person

Fig. 2: Accuracy of predicting family members by lost nodes.

mall and moves towards it at an average walking speed of $1.2 \pm \varepsilon\ m/sec$ [15]. When the center reaches the destination, it picks another location at random as destination using random walk model. Each family member attempts to stay within a circular area around the center. The area of the circle is a function of family size. Each node moves at its own speed, slowing down if it is ahead of the circle and speeding up if it falls behind.

Figure 2 shows the performance of the six algorithms presented in section IV. Here we present results for high density of crowds. We have similar results for medium- and low- density crowds.

For the first three algorithms, we use the criteria defined in section IV-D. The last three algorithms, we have complete information about all interactions a node has had during its lifetime and we use the criteria:

- $\texttt{UNLIMITED}_1$: Time only
- $\texttt{UNLIMITED}_2$: Frequency and time
- $\texttt{UNLIMITED}_3$: Frequency, time, and area under the time curve

For the rest of the three algorithms, we use the criteria defined in section IV-D.

As we can see, algorithms using limited memory are able to predict family members with 90% accuracy, even in the presence of high percentage loss.

VI. Future Work and Conclusions

In this paper we presented a novel technological solution to determine groups of families in a crowded environment that requires no input or expertise from the user. We presented several algorithms that use proximity to other nodes to predict family members. We evaluated our algorithms under a realistic scenario and showed that even when severely limited by memory, we are able to predict families of a lost member with high accuracy. Implementing these algorithms as a real system is part of our future work.

References

[1] "Amusement park and attractions industry statistics," http://www.iaapa.org/resources/by-park-type/amusement-parks-and-attractions/industry-statistics, accessed: 2015-04-4.

[2] "Preventing kids from getting really lost. and what to do if it happens!" http://www.parentguidenews.com/Articles/PreventingKidsFromGettingReallyLost, accessed: 2015-04-4.

[3] S. Sridhar, P. Gautret, and P. Brouqui, "A comprehensive review of the kumbh mela: identifying risks for spread of infectious diseases," *Clinical Microbiology and Infection*, 2014.

[4] "Lost and found in a sea of tens of millions at india festival," http://articles.latimes.com/2013/mar/02/world/la-fg-india-festival-lost-20130303, accessed: 2015-04-4.

[5] "Hajjlocator: Mobile interactive monitoring and tracking system for hajj pilgrims," http://isif.asia/projects/projects/view/167, accessed: 2015-04-4.

[6] K. Liu and X. Li, "Finding nemo: Finding your lost child in crowds via mobile crowd sensing," in *Mobile Ad Hoc and Sensor Systems (MASS), 2014 IEEE 11th International Conference on*, Oct 2014, pp. 1–9.

[7] "Child tracking: What parents should know," http://www.cozi.com/live-simply/child-tracking-what-parents-should-know, accessed: 2015-04-4.

[8] "Find what matters," https://www.thetileapp.com/, accessed: 2015-04-4.

[9] M. H. O. Balfas, "Ubitrack: A study on lost person activity estimation using accelerometer data from smartphones," Ph.D. dissertation, MARQUETTE UNIVERSITY, 2015. [Online]. Available: http://gradworks.umi.com/15/86/1586539.html

[10] S. A. Aly, "A missing and found recognition system for hajj and umrah," *CoRR*, vol. abs/1208.5365, 2012. [Online]. Available: http://arxiv.org/abs/1208.5365

[11] D. Roggen, M. Wirz, G. Tröster, and D. Helbing, "Recognition of crowd behavior from mobile sensors with pattern analysis and graph clustering methods," *arXiv preprint arXiv:1109.1664*, 2011.

[12] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology," *Sensors*, vol. 12, no. 9, pp. 11 734–11 753, 2012.

[13] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, "A group mobility model for ad hoc wireless networks," in *Proceedings of the 2Nd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '99. New York, NY, USA: ACM, 1999, pp. 53–60. [Online]. Available: http://doi.acm.org/10.1145/313237.313248

[14] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile computing*, pp. 153–181, 1996.

[15] K. Fitzpatrick, M. A. Brewer, and S. Turner, "Another look at pedestrian walking speed," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1982, no. 1, pp. 21–29, 2006.