Crowd (Soft) Control

Moving Beyond The Opportunistic

John Rula and Fabián E. Bustamante
Northwestern University
{john.rula|fabianb}@eecs.northwestern.edu

ABSTRACT

A number of novel wireless networked services, ranging from participatory sensing to social networking, leverage the increasing capabilities of mobile devices and the movement of the individuals carrying them. For many of these systems, their effectiveness fundamentally depends on coverage and the particular mobility patterns of the participants. Given the strong spatial and temporal regularity of human mobility, the needed coverage can typically only be attained through a large participant base.

In this paper we explore an alternative approach to attain coverage without scale – (soft) controlling the movement of participants. We present Crowd Soft Control (CSC), an approach to exert limited control over the temporal and spatial movements of mobile users by leveraging the built-in incentives of location-based gaming and social applications. By pairing network services with these location-based apps, CSC allows researchers to use an application's incentives (e.g. games objectives) to control the movement of participating users, increasing the effectiveness and efficiency of the associated network service. After outlining the case for Crowd Soft Control, we present an initial prototype of our ideas and discuss potential benefits and costs in the context of two case studies.

1. INTRODUCTION

A large class of novel wireless networked services, ranging from delay-tolerant networking [2, 23], to digital geo-tagging [22] and participatory sensing [3,8,13], leverages the increasing capabilities of mobile devices and the movement of the individuals carrying them. Whether aimed at providing connectivity in challenging settings, entertainment, or support for citizen science, the effectiveness of these proposed systems depends on coverage that can typically only be attained through a critical mass of participants.

For instance, several projects have explored the use of data carriers to bridge the gaps in intermittently connected mobile networks [2, 6, 23]. To effectively overcome the lack of end-to-end connectivity, the proposed approaches must either assume a large participatory base or introduce some form of control over the mobility of participants. In the context of community sensing, accurately characterizing a distributed phenomena, such as pollution levels or vehicular road conditions, similarly requires sufficient coverage to capture the variations of the metrics of interest across space and over time. The uncoordinated movement of participants, however, with typical strong spatial and temporal regularity [4], means that *certain areas of potential interest may not be covered or the collected data may never reach its destination due to limited connectivity*.

Recently, a number of research efforts have proposed addressing this problem by using either the social networks of large populations or their previously recorded mobility patterns [7, 11, 16]. Hui et al. [7], for instance, suggests leveraging participants' social network connections to predict movements; these predictions are then used to aid data forwarding decisions delay tolerant networks (DTNs). Reddy et al. [16] proposes building a recruitment system for participatory sensing based on a large participant base and their previously collected mobility traces. Thus, by carefully selecting participants from a sufficiently large pool based on their recorded movement or social network, a researcher can piece individual mobility patterns together into a quilt of the necessary coverage.

In this paper, we present an alternative approach *to attain coverage without scale by (soft) controlling the movement of participants*. We introduce *Crowd Soft Control (CSC)*, an approach to exert limited control over the temporal and spatial movements of mobile users by leveraging the built-in incentives of location-based gaming and social applications. Location-based applications rely on a users' position and orientation to enhance their game play or social networking experience. For instance, a class of games known as augmented reality games use a mobile device's GPS location along with its camera and compass to overlay virtual information on top of a user's camera view [21]. Social networking games like Foursquare [1] and Gowalla [2] encourage user's to "check-in" at locations to alert their friends to their whereabouts and collect virtual rewards. We argue that by leveraging the incentive of such location-based applications (e.g. offering bonus points for visiting a certain location), users' actions can be manipulated to achieve a network service's goal (e.g. taking a measurement at that location).

The following section expands on the case for Crowd Soft Control and provides an overview of a general architecture. Sections 3 and 4 present an initial prototype of our ideas and illustrate them using two case studies. We review related work in Sec. 5 and discuss open issues and future work in Sec. 6 and 7.

HotMobile'12 February 28–29, 2012, San Diego, CA, USA.

---

[1] foursquare.com

[2] gowalla.com

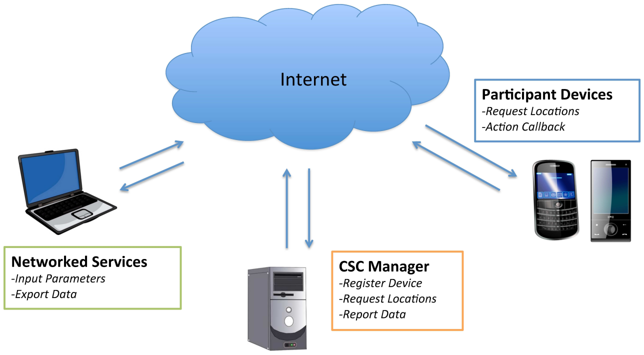


Figure 1: Crowd Soft Control at work – The CSC manager translates the networked service targets into game objectives and matches players with particular targets.

2. CROWD SOFT CONTROL

The proliferation of resource-rich mobile devices, such as phones, PDAs and tablets, has brought a wide-rage of proposals for community-supported networked services. Whether for monitoring atmospheric conditions or to provide connectivity to nomadic communities, this class of networked services leverage the increasing capabilities of mobile devices and the movement of the individuals carrying them.

Consider the case of a community sensing service. Community sensing, either participatory or opportunistic, is motivated by the observation that new mobile devices include not only computing and communication capabilities, but also embed a number of sensing systems such as microphone, cameras, GPS and accelerators. By recruiting a number of these devices, one could consider tracking a phenomena of interest (e.g. noise levels) purely through software running on them. By either controlling the mobility of the device carriers [5, 23] or ensuring a sufficiently large population of them [16], a researcher could plan to cover reasonable areas of interest.

Most deployments, however, tend to be limited in scale and only a few support controlling the mobility of carriers [5]. Even if some of the proposed sensing platforms were to see wide adoption, one could easily imagine enough interesting services to deploy over the participating nodes as to make any scale of adoption insufficient!

Without sufficient scale or the ability to control the carrier's mobility, the typically strong spatial and temporal regularity [4] of human movements, implies that certain areas of potential interest may not be covered or the collected data may never reach its destination due to limited connectivity.

CSC goal is to attain coverage without scale by leveraging the incentives of location-based applications, such as gaming, to control human mobility. In the context of a community sensing service, a point of interest in the covered area will be introduced as a game's objective (perhaps an item in a virtual scavenger hunt) and associated with sufficient game credit as to drive players toward it.

In the following paragraphs, we outline a general architecture for CSC and illustrate our ideas in the context of some potential location-based applications and networked services.

2.1 Crowd Soft Control at Work

A CSC setting includes three key elements: a networked service, e.g. a noise pollution monitoring service, the set of mobile devices taking part of a game, and the CSC manager mediating between them. Figure 1 illustrate these components and their interaction.

Networked service. A third-party networked service that submits requests for assignment to participating devices.

CSC participants devices. Mobile devices participants, such as players in a third-party location-based game, registered with the CSC manager and accepting requests from it.

CSC manager. The manager, acting as a matchmaker, that assigns networked service's requests to devices according to a device's capabilities and physical and temporal location.

A networked service is any system which can benefit from the controlled movement of mobile devices. Examples include community sensing services, such as noise pollution monitoring, environmental sensing and traffic monitoring [9, 12, 13], and delay-tolerant services such as pocket switched networks and location based media sharing [6, 11]. The networked service submits (i) a set of requirements for devices and locations and (ii) a number of actions to carry on at the specified locations and times by the selected devices. Example requirements include type of sensors available in the devices, as well as temporal or spatial preferences or constraints. Specified actions can range from recording sensors' readings to exchanging content with a collocated devices.

Networked services are supported, at least in part, by the participating mobile devices. These participating devices are host to third-party location-based applications, such as an augmented reality game or a mobile social networking service. These applications include incentives to direct users toward particular locations as either part of the game or the social interaction model. For a virtual scavenger game, for instance, this could be the coordinates of a virtual item or a challenge [17]. For a geo-social networking application such as FourSquare[3], it could be a mayorship with its associated discounts. These location-based applications are extended to interact with CSC[4] – to incorporate CSC's locations into the application logic, carry on associated actions and reporting their outcomes.

Matching the needs of networked services with the compatible mobile devices is the *CSC manager*. Pairings are based on the spatial and temporal location of the device, as well as the device's capabilities.

2.2 Example Networked Services

The control, even limited, enable by Crowd Soft Control can be potentially beneficial to a wide range of networked services. The following are just some clear, illustrative examples including community sensing, pocket switched networks and network alleviation through bulk transfer.

Community Sensing. Crowd soft control can benefit community sensing through finer control of the location and time of sensors. This makes possible driving carriers into areas with little to no sensing coverage, allowing for greater coverage with fewer participants or higher densities of measurements in a specific region.

Pocket Switched Networks. Pocket switched networks (PSN) exploit human mobility in order to transfer data between mobile users' devices. Crowd soft control allows a PSN to regain some of that control from opportunistic interactions. This control can be leveraged, for instance, to coordinate a hand-off of data between two carriers, thus creating more efficient and reliable routing mechanisms.

---

[3] http://www.foursquare.com

[4] With our current prototype, extending a game can be done with fewer than 25 lines of code.

Mobile Network Alleviation. Most mobile devices today support a number of possible connection types [14]. With cellular networks, services are provided by discrete cells, causing services to differ based on location for signal strength, latency and throughput. Therefore, it is in the best interest of both the network provider and user to be able to pinpoint areas of higher throughput or cheaper transit costs. Research in 3G data-offloading through delay tolerant transmissions has emerged as a potential solution to mobile network congestion. Trestian et. al outlines the use of "Drop Zones" as a means for network alleviation specifically for uploading user-generated content [19]. In this context, crowd soft control would make possible to direct movement toward these "Drop Zones" at peak traffic times.

3. A CSC PROTOTYPE

To explore the proposed ideas behind CSC, we constructed a prototype of a CSC system, including the mobile device middleware and a CSC manager, as well as a number of networked services and location-based applications. Our two example networked services are all in the context of community sensing, for simplicity of implementation and presentation. The mechanisms of mobile sensing allow a simpler translation between sensing requirements and location-based applications. Experimenting with other classes of networked services is part of our future work. We also implemented one location-based application, an augmented reality game for the Android platform. The CSC prototype – the mobile device middleware and the CSC manager, includes nearly 5,000 lines of code in both python (1,020) and Java (4,258).

3.1 CSC Manager

The CSC Manager accepts the requirements sent to it by the different network services, and pairs these requirements to requesting mobile devices. The CSC Manager is implemented as a python web server running a REST webservice over a serialized XML format.

Each researcher uploads their requirements to the CSC Manager. These requirements consist of a structured XML file containing geographic areas, times, and sensing actions to occur. These requirements are then paired to active mobile devices when a request occurs.

3.2 CSC Middleware

Our middleware library, built for the Android mobile operating system interfaces between the CSC server and the location-based application running on the mobile device. The middleware enables device registration, location information exchange and action execution. It executes any required tasks and passes spatial and timing variables to the location based game application.

When the CSC is first accessed, it goes through a registration procedure of the device. During this process, it first determines the sensing capabilities of the phone, recording the types and availability of different phone sensors. It then assigns the device a unique ID, which is stored for future phone identification. This information is then uploaded to the CSC server. The unique ID is used to provide complete anonymity to participating mobile users. Registration data is stored so that a phone's sensor capabilities, and therefore its ability to provide the requested sensing or action response, can be quickly determined.

To request location suggestions, the game makes a call to our CSC middleware passing its current location, the maximum distance the points can be from the current location, and number of points to be returned. These parameters limit the scope of the requirements returned, as well as provide a way for the developer to tailor to their individual application. The middleware then contacts the CSC Manager, and receives a set of network service requirements for the area around the current location.

The library translates these requirements into a set of points, each of which contain the following information:

Location: The location desired stated in terms of latitude and longitude, and optionally altitude and heading.

Action: The type of action to be triggered at the particular location and time.

Expiration Time: The time when the request is no longer valid; this is used to control the timing and relevancy of actions.

Ranking: The relative importance of the location. This is used by the game to control differentiate between incentives.

3.2.1 CSC Integration

The location-based application interacts with CSC through the CSC device API. This interaction can be split in two stages, *point generation* and *action*. At *point generation*, the location-based application queries CSC to obtain and incorporate points based on the service sensing requirements. In the *action* stage, the device has reached the location and the associated action is triggered.

Point Generation. The Point Generation stage, the CSC client creates a series of location objects based on the requirements it received. For simplicity, the CSC API provides a single call to generate locations.

`getClosestNLocations(lat, lng, radius, N)`: Returns the specified N locations. Each location is derived from the sensing requirements described earlier. lat and lng represent the current location, $radius$ is the distance representing the game area, N is the number of locations requested.

Action. In the action stage, the mobile device has reached the destination, and the application objective is completed. When this occurs, a callback is fired to the CSC middleware to initiate the appropriate sensing action as defined in the requirements.

`performAction()`: Fires a callback to the CSC library to execute the predetermined sensing action. The CSC middleware references the location context and performs the appropriate sensing action.

3.3 Ghost Hunter Game

We implemented a proof-of-concept augmented reality game for Android called Ghost Hunter. In Ghost Hunter, a player chases ghosts and other monsters around her neighborhood and "zaps" them by capturing their photo through an augmented reality display, for which she is awarded points. We chose augmented reality for our implementation because it allows us the most control over a user's device; in addition to a user's location, augmented reality allows us to control the heading and orientation of the device.

The game consists of two parts: a map screen and an augmented reality display. To play, the user walks around toward the location of "ghosts" which are overlaid on the map screen. Once close enough, the game switches to an augmented reality mode where the player follows arrows on the screen to place the ghost in their cross hairs. Once the ghost is within their sights, the player is able to capture it. Ghost Hunter is implemented in about 3,500 lines of Java code.

CSC integrates into existing applications with a minimal amount of code. Table 1 shows the number of lines needed to integrate our CSC middleware into our Ghost Hunter Game. All external code changes come to only 25 lines of code. Although the specific

Function	Num. of lines
Location Request	2
Device Action	10
Other	13

Table 1: Cost to integrate CSC into our example game in terms of number of additional lines of code used. CSC was integrated with only 25 lines of additional code.

number of lines to be modified would be different for different games, we do not expect the task to be significantly more complex. We are currently porting other open source games and location-based applications over CSC.

4. CASE STUDIES

In this section, we evaluate the feasibility of the proposed ideas by applying our prototype implementation of CSC to two separate case studies. Each case study emulates an existing mobile sensing system. The first one crowdsources three-dimensional models of buildings [20] while the second monitors noise pollution around a university campus [9]. These examples rely on different sensors and have significantly different requirements on data collection. The first one requires complete angular coverage of a specific object, while the second one needs broad coverage over a relatively large area.

We use a common case study framework – the prototype system we describe in Section 3, to experiment with these mobile sensing applications. We implemented an extensible mobile middleware for the Android operating system (Sec. 3.2) along with a web-based pairing service (Sec. 3.1). To evaluate the benefits of CSC over traditional sensing applications, we designed and built a location-based augmented reality game (*Ghost Hunter*) as well as a more traditional community sensing service (our *control* service). In our control sensing service, users were instructed to randomly take measurements around a college campus. We use the data provided by the users of this application to establish a baseline of coverage.

4.1 Case Study 1: Photo Hunter

Recently, online mapping tools have utilized large collections of ground-level urban imagery. These images have new potential applications in novel augmented reality applications, monitoring and analyzing building infrastructure, and urban planning. Current acquisition techniques are expensive, and only available to large enterprise companies. Recent work by Tuite el al. [20] has attempted to crowdsource images to create these 3D models using competition as an incentive. We recreate the objectives of their experiments through CSC.

The 3D modeling experiment requires strong control over the geographic and angular position of the photos being taken. A 3D model requires complete angular coverage around a building's perimeter, for instance, along with several other photos taken from different distances and different angles. Using CSC, photos were captured by our Ghost Hunter game by placing a ghost in the exact heading and zenith of the preferred photo from each location. When a ghost is "captured" as part of the game, a photo is taken in the background. To generate a baseline for comparison in this case, we follow the approach described by Tuite et al, crawling Flickr for photos of the same landmark.

Figure 3 shows the location of both sets of pictures (using X to label the location of Flickr pictures and a $circle$ to label those collected through Ghost Hunter. As can be seen from the figure, Flickr photos tend to be clustered (in the "most" interesting angles from a visitor's point of view), compared with the relative isotropy of images obtained through CSC via Ghost Hunter.

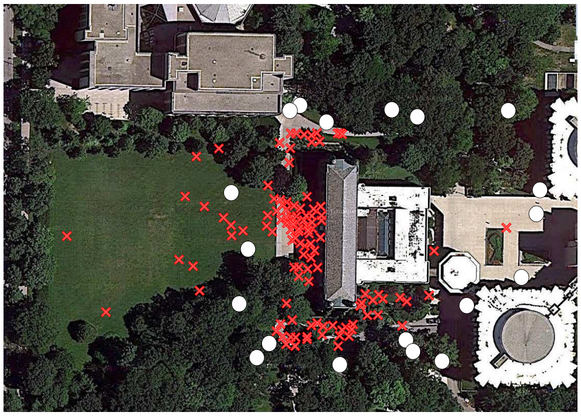


Figure 3: Photos of Deering Library. Red X's represent the location of Flickr photos. White circles represent photos taken through CSC.

4.2 Case Study 2: Noise Pollution

Noise pollution has become a major problem in major metropolitan areas. Recent studies have shown that a significant portion of urban dwellers are regularly exposed to harmful levels of noise pollution. In order to better understand this phenomena, a system which monitors noise pollution across a city must be able to obtain measurements at specific times and specific places across an urban landscape.

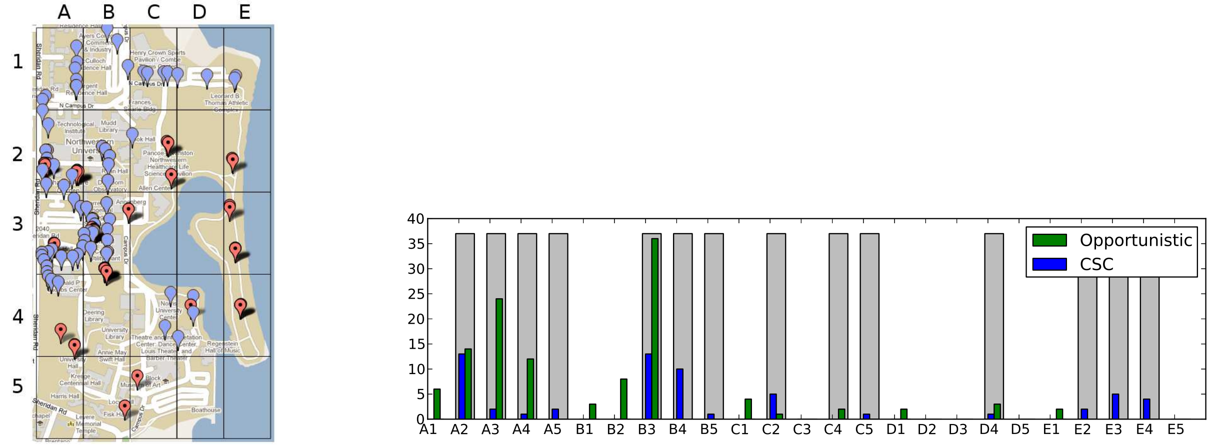
In our experiments, we have users replicate these requirements by scheduling measurements across an area with both spatial and temporal specification. For comparison, we ran our experiments with both our CSC enabled game as well as a *traditional* sensing application. This traditional sensing application is modeled after several participatory sensing applications which ask a user to voluntarily take measurements as they follow their typical daily lives. Our application asked users to take noise measurements throughout their time spent on campus.

The Opportunistic and CSC measurements are plotted on a gridded map in Figure 2a. The figure illustrates the problems with opportunistic measurement approach. While the CSC measurements are spread over heavily trafficked areas, those collected opportunistically tend to be concentrated on a few regions. This disparity is clearly shown in the bar graph of Figure 2b. Of the 13 grid spaces where measurements were requested, opportunistic sensing was only able to achieve any level of coverage in 7 of them (54%). In comparison, CSC provided much better coverage with 93% (12 of 13) of requested areas.

5. RELATED WORK

CSC builds on an extensive set of research involving human mobility and its effect on mobile network services. Research ranging from community sensing [3] to delay-tolerant networks [23] have utilized the mobility of hand held devices to provide new network services.

Research in several areas uses mobility information through either location traces or social networking develop intelligent mobile network services. Pietiläinen et al. proposes a system to use the information from social networks to form ad-hoc routing



(a) Map of measurements taken.

(b) Graph of measurement coverage.

Figure 2: Left. Measurements for the Microphone sensor. Red (dotted) represent the measurements taken by CSC. Blue (solid) represent the opportunistic measurements taken. Right. Number of measurements per grid square for each measurement type for the second case study. The gray bars indicate grid squares where measurements were requested.

overlays for content distribution [15]. Hui et al. [7] suggests using a user's social network as another means to predict future movements, also to aid in routing overlays. Bild et al. proposes uses personal mobility traces to form a *location profile* to aid in the construction of scalable MANETs [1]. Similarly, Nicholson et al. [14] uses personal mobility traces to make decisions for whether to use cellular or wifi interfaces for data transfer. This research uses the predictability of human movements to influence decisions for system behavior, whether they be routing or device operation. The difference between these and CSC is these systems make no attempt to influence the mobility of the users involved. Although, the predictability of human movement allows more intelligent decisions to be made within mobile systems, they are all still limited by their opportunistic interactions. In contrast, CSC attempts to expand on the capabilities of each user by augmenting the range of their personal movement through the incentives present in the mobile game or social application used. In this way CSC is complementary to these existing mobility based network services.

Reddy et al. [16] proposes a recruitment framework to select users based on past participation and locations for use in participatory sensing. This along with those mentioned above require a large pool of users to provide effectiveness since they still operate under opportunistic interactions between users. Further, attaining coverage still requires a large number of participants since it operates under the assumption that users' mobility cannot be changed. In contrast, CSC is able to obtain equivalent coverage with fewer individuals by manipulating their mobility.

Previous research has attempted data collection through location-based gaming, including CityExplorer [10], a game where geospatial data is collected through a location-based game, and Photocity [20], a game through which players are trained to take more *useful* photos to aid in generating three dimensional models of buildings. CSC builds on these earlier works by providing a dynamic framework for network services rather than a single game for a specific purpose.

6. DISCUSSION AND FUTURE WORK

Privacy remains a key concern, as in all mobile systems. By its design CSC retains no identifiable information from the users. All information is stored as a unique ID which is randomly assigned to each client. There is still the opportunity for identification to be inferred from geographic context [18]. However, CSC adds no additional loss of privacy beyond those in the existing location-based application.

Similarly, there is a potential risk of allowing researchers to “lure" users toward potentially dangerous areas. We argue that CSC does not extend the risks already available in any location-based application. As augmented reality gamers can be trusted to exercise their best judgement during play, users of extended location-based applications should be trusted to judge the suggestion made available through a CSC subsystem.

Our initial CSC prototype uses two example networked services both drawn from cooperative sensing. We plan to explore other networked services as well as investigate the challenges involved in running a mobile platform with diverse, concurrent network services – defining universal requirements for use across all network services, ensuring fairness across multiple mobile devices and multiple network services, and obtaining wide scale deployment across several mobile applications.

In terms of user recruitment, CSC partially moves the focus from individual users and toward developers. We believe this may be a preferable approach to more common pure crowdsourcing models that require each individual participant to be “enlisted”. With CSC, a single developer has the potential to reach a large number of users depending on the popularity of her application. Other forms of incentive, such as micro payments, are of course possible and mostly independent of the proposed ideas.

7. CONCLUSION

We presented Crowd Soft Control (CSC), an approach to control the temporal and spatial movements of mobile users by leveraging the built-in incentives of location-based gaming and social applications. By pairing network services, such as a mobile sensing application, with these location-based apps, CSC allows researchers to leverage an application's incentives (e.g. game objectives) to control the movement of participating users, increasing the effectiveness and efficiency of their network service. We made the case for CSC, described an initial prototype of these ideas, and discussed its potential benefits and costs in the context of two case studies. Preliminary evaluation of our prototype implementation shows a 72% improvement in coverage over traditional opportunistic measurements. In ongoing work, we are experimenting with other classes of network services and location-based application and investigating the degree to which users can be soft controlled.

8. ACKNOWLEDGEMENTS

We would like to thank our shepherd, Vishnu Navda, and the anonymous reviewers for their detailed and helpful feedback. This work was supported supported in part by NSF Awards CNS XXXX, CNS XXXX and by grants through Google.

9. REFERENCES

[1] BILD, D. R., LIU, Y., DICK, R. P., MAO, Z. M., AND WALLACH, D. S. Using predictable mobility patterns to support scalable and secure manets of handheld devices. In *Proc. MobiArch* (2011).

[2] DORIA, A. Providing connectivity to the Saami nomadic community. In *Proc. Int. Conf. on Open Collaborative Design for Sustainable Innovation* (2002).

[3] EISENMAN, S., MILUZZO, E., LANE, N., PETERSON, R., AHN, G., AND CAMPBELL, A. Bikenet: A mobile sensing system for cyclist experience mapping. *ACM Transactions on Sensor Networks (TOSN) 6*, 1 (2009), 6.

[4] GONZALEZ, M. C., HIDALGO, C. A., AND BARABASI, A.-L. Understanding individual human mobility patterns. *Nature 453*, 7196 (June 2008), 779–782.

[5] HOWARD, A. Experiments with large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. *International Journal of Robotics Research* (2006).

[6] HUI, P., CHAINTREAU, A., SCOTT, J., GASS, R., CROWCROFT, J., AND DIOT, C. Pocket switched networks and human mobility in conference environments. In *Proc. WDTN* (2005).

[7] HUI, P., CROWCROFT, J., AND YONEKI, E. Bubble rap: Social-based forwarding in delay-tolerant networks. *IEEE Transactions on Mobile Computing 10* (2011), 1576–1589.

[8] HULL, B., BYCHKOVSKY, V., ZHANG, Y., CHEN, K., GORACZKO, M., MIU, A., SHIH, E., BALAKRISHNAN, H., AND MADDEN, S. Cartel: a distributed mobile sensor computing system. In *Proc. SenSys* (2006).

[9] MAISONNEUVE, N., STEVENS, M., NIESSEN, M., AND STEELS, L. Noisetube: Measuring and mapping noise pollution with mobile phones. *Information Technologies in Environmental Engineering* (2009), 215–228.

[10] MATYAS, S., MATYAS, C., SCHLIEDER, C., KIEFER, P., MITARAI, H., AND KAMATA, M. Designing location-based mobile games with a purpose: collecting geospatial data with cityexplorer. In *Proc. ACE* (2008).

[11] MCNAMARA, L., MASCOLO, C., AND CAPRA, L. Media sharing based on colocation prediction in urban transport. In *Proc. MobiCom* (2008).

[12] MOHAN, P., PADMANABHAN, V., AND RAMJEE, R. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *Proc. SenSys* (2008).

[13] MUN, M., REDDY, S., SHILTON, K., YAU, N., BURKE, J., ESTRIN, D., HANSEN, M., HOWARD, E., WEST, R., AND BODA, P. Peir, the personal environmental impact report, as a platform for participatory sensing systems research. In *Proc. MobiSys* (2009).

[14] NICHOLSON, A. J., AND NOBLE, B. D. Breadcrumbs: forecasting mobile connectivity. In *Proc. MobiCom* (2008).

[15] PIETILÄINEN, A.-K., OLIVER, E., LEBRUN, J., VARGHESE, G., AND DIOT, C. MobiClique: Middleware for mobile social networking. In *Proc. 2nd Workshop on Online Social Networks* (2009).

[16] REDDY, S., ESTRIN, D., AND SRIVASTAVA, M. Recruitment Framework for Participatory Sensing Data Collections. In *Proc. of Pervasive Computing* (2010).

[17] SCVNGR, I. Scvngr. http://www.scvngr.com.

[18] SHILTON, K. Four billion little brothers?: Privacy, mobile phones, and ubiquitous data collection. *Commun. ACM 52* (November 2009), 48–53.

[19] TRESTIAN, I., RANJAN, S., KUZMANOVIC, A., AND NUCCI, A. Taming user-generated content in mobile networks via drop zones. In *Proc. IEEE INFOCOM* (2011).

[20] TUITE, K., SNAVELY, N., HSIAO, D., TABING, N., AND POPOVIC, Z. Photocity: training experts at large-scale image acquisition through a competitive game. In *Proc. CHI* (2011).

[21] WAGNER, D., AND SCHMALSTIEG, D. First steps towards handheld augmented reality. In *Proc. of Wearable Computers* (2003).

[22] WU, M.-Y., HSU, C.-C., AND LEE, J.-H. A geotagging scheme using image steganography and gps information authentication. In *Proc. IIH-MSP* (2009).

[23] ZHAO, W., AMMAR, M., AND ZEGURA, E. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *Proc. MobiHoc* (2004).