Location and Mobility in a Sensor Network of Mobile Phones

Aman Kansal
Microsoft Research
kansal@microsoft.com

Feng Zhao
Microsoft Research
zhao@microsoft.com

ABSTRACT

Mobile phones have two sensors: a camera and a microphone. Our goal in this position paper it to explore the use of these sensors for building an audio-visual sensor network that exploits the deployed base of millions of mobile phones worldwide. Among the several salient features of such a sensor network, we focus on mobility. Mobility is advantageous since it yields significant advantage in spatial coverage. However, due to the uncontrolled nature of device motion, it is difficult to sample a required region with a given device. We propose a data centric abstraction to deal with this difficulty. Rather than treating the physical devices as our sensor nodes, we introduce a layer of static virtual sensor nodes corresponding to the sampled data locations. The virtual nodes corresponding to the required region to be sensed can be queried directly to obtain data samples for that region. We discuss how the locations of the virtual sensor nodes can be enhanced, and sometimes derived, using the visual data content itself. Experiments with real data are presented to expose some of the practical considerations for our design approach.

Categories and Subject Descriptors

C.2.4 [**Computer Systems Organization**]: Computer Communication Networks—*Distributed Systems*

General Terms

Algorithms, Design, Management

Keywords

mobile sensing, network organization, coverage, spatial resolution

1. INTRODUCTION

Mobile phones can be used as sensor nodes [1]. They all have a microphone, and most have a camera. Not only can the audio-visual data be processed to derive other interesting sensing modalities, but also additional sensors can be connected to a phone [18] using Bluetooth. Mobile phones are connected to a network infrastructure and have some form of data connectivity, ranging from short message service (SMS) capability to broadband wireless connectivity (eg. GPRS, 3G). The challenges in power management that affect many unattended sensor network deployments become relatively benign here, since a phone can be easily recharged in its user's car, office, or home. These features make it feasible to use a phone as a sensor and thus we propose to build a sensor network using multiple individual owned cellular phones as its sensing substrate.

1.1 Sensing Advantages

Using mobile phones has several advantages for sensor networking applications. Firstly, a large number of cell-phones already exists around the world, providing the physical sensing infrastructure. Deploying the sensing hardware and providing it with network and power requires significant effort in other sensor networking systems.

Secondly, such a system can take advantage of the community effect. Many useful systems have been built where several contributors each develop and share a small component of a much larger system, such as Wikipedia, Linux, and other Web 2.0 applications. Such an approach, sometimes referred to as peer production [2], can leverage small amounts of sensor data contributed by mobile phone users to enable useful sensing applications. For example, a single mobile phone owner taking pictures of broken sidewalks on her fitness running route in an urban neighborhood may generate a dataset with very limited utility but if several such runners share their data, the utility grows significantly and the dataset may suddenly become useful for more activities such as damage mapping for city repair planning, new running route planning in runners' e-groups, or urban lifestyle modeling for sports goods related business dashboards.

Thirdly, mobile phones can provide coverage where static sensors are hard to deploy and maintain. No single entity may have the access rights to place sensors across the complete coverage domain required by an application, such as a combination of subway stations, public parks, and shopping malls. The number of static sensors required to cover the same spatial expanse as covered by a single mobile device[1] may be prohibitively expensive to deploy.

Further, each mobile device is associated with a human user, whose assistance can sometimes be used to enhance application functionality, such as by pointing the cellphone camera at the object of interest. Human assistance may be very limited, and may depend on application, but can often help overcome certain hurdles that are hard to overcome otherwise.

While the advantages of a shared system include the vast cov-

NOSSDAV '07 Urbana, Illinois USA

[1] There is a trade-off between spatial and temporal coverage when motion is used. Clearly, mobile phones may not cater to applications where continuous coverage in time is critical.

erage expanse that a single dedicated system may never be able to match, the sharing of physical resources among multiple applications, has certain limitations compared to dedicated and application specific sensor networks. These trade-offs include difficulty in providing performance guarantees, limitations in control of the sensor nodes, more complex privacy and data ownership management, and ensuring protection from malicious users. Such issues affect many open and shared system, and community feedback techniques such as used in Wikipedia and Web 2.0 applications, may help reduce their severity. Our objective is to develop a system that provides best effort service and allows multiple sensing applications to leverage shared resources to the extent available.

1.2 Usage Scenarios

There are many examples of applications that are feasible using sensor networks of shared mobile phones. In many cases, there is an inherent incentive for the user to take samples and share them within a larger user group, such as in the previous example of a runner mapping broken sidewalks for her health enthusiasts e-group, for car drivers in vulnerable river valleys to report flooded roads, for hikers to click pictures of trail blockages after a rainstorm, for shoppers to share window displays from shopping malls, home owners to map noise levels in their community, commuters to map pollution levels across their city, and so on. In other instances, incentives can be provided by the users who need the sensor data. For instance, a surfer who lives some distance from the beach, may place a request for current images showing wave conditions at the beach in a region of his interest (such as, by using a map based interface) and offer a small monetary compensation in return. Phone users currently at the beach who happen to be members of a surfing group, or who have subscribed for data requests with monetary compensation, receive this request as an SMS, and may respond with pictures using very little effort. They may accumulate the small compensations from such contributions to pay part of their phone bill. Similar techniques may be used by news networks to get instant news coverage even before their correspondent reaches the scene, businesses to map customer interest at malls, and other applications that can use the audio-visual feeds to compute metrics of interest.

Many of these applications are based on sensor data that users may share without significant privacy concerns. However, where privacy or data ownership is a concern, the data may be shared within restricted groups only and while the trusted coordination system may archive and index all data, applications may be only able to access restricted portions of it. Such an approach has been used in existing image sharing systems such as the Flickr web service [5] that archives all images but allows applications to access only the data for which they are authorized. Some of the techniques we discuss in this paper for location based abstraction, do not use the raw image data but some features derived from it. For highly sensitive data, the contributors could locally compute the relevant features (which do not reveal the human understandable visual content) and share those. The coordination system could use the features to build the proposed abstraction and data indices. When an application accesses the actual data, the data is served by the data owner, according to their desired control and access policies.

The emerging shift in the Internet to user generated content such as blogs or moblogs, shared images, and amateur produced videos also indicates a trend toward the possibility of shared sensing using mobile devices.

1.3 Key Contributions

Our overall project goals are to provide coordination and networking mechanisms that allow multiple sensing applications to access third party shared resources in an efficient manner. Specifically, the first form of shared resources considered are mobile phones, due to their widespread availability. Building such a sensor network using uncoordinated mobile phones, where each phone is serving a different individual, the system as a whole is serving multiple sensing applications, and the phones move without any application's control, involves many challenging issues. We discuss some of these challenges and present one approach to realizing a coherent shared sensor network based on this volatile swarm of mobile devices, in section 2.

Presenting the data collected by this highly volatile swarm in a usable manner to the sensor network applications makes it critical to obtain location information. Mobility helps increase the spatial coverage significantly, possibly by orders of magnitude, but without location, the samples taken by a mobile device cannot be associated with the corresponding spatial coverage. We discuss how location can be obtained at mobile devices using current technology. We also discuss how the audio-visual nature of the sensed data can be exploited to enhance location accuracy in section 3.

2. SYSTEM DESIGN

2.1 Assumptions and Requirements

We consider a sensor network of mobile phones which is built as a shared system. Each phone serves its local user's needs first, such as making and receiving voice calls. The user continues to use any software, such as calenders or games, installed on their phone as they need. The sensor networking application only uses the phones as its sensors when available. Thus, the system can only work in best effort mode without any expectation of hard guarantees.

Our system does not control or even know the motion plan of the devices. Sensing requirements must be dynamically mapped to relevant devices in the underlying swarm of sensor devices.

Unlike dedicated and application specific sensor networks where most devices are homogeneous with known configurations, in the shared mobile phone sensor network, devices may be highly heterogeneous, not only in their hardware resources and bandwidth availability but also in terms of human user's willingness to share. We require the sensor networking application to be able to accept the privacy and sharing policies set by the local user of each device. Thus, different mobile phones may participate in the shared sensor network to varying degrees, based on their privacy sensitivity, willingness to share battery energy and bandwidth, device performance and capabilities, local workload, and willingness to provide human assistance for various applications.

2.2 System Architecture

In most current sensor network deployments, sensor nodes are accessed directly using their node ID's or network addresses. While each phone has a unique network address (a phone number, among other unique identifiers) and can be accessed using that identity, this approach leads to difficulties in managing the spatial coverage due to node mobility. Also, this approach requires the application to know the network identities of the mobile phones in advance, which is a significant overhead for the application due to the large number of phones in the shared system.

We propose to use a data based abstraction that does not rely on the node identities of the physical devices. A layer of virtual sensor nodes is superimposed on the physical sensor network. A virtual sensor node at any point in space is based on data samples taken at that point. It is thus static. The stream of data coming from a virtual sensor node corresponds to data samples taken by a single or multiple physical nodes when they visited that location. The stream may be sparse in time if that location is visited infrequently. Figure 1 illustrates this abstraction. The figure shows three key entities in

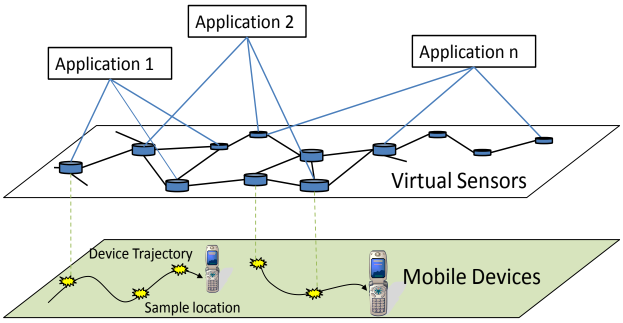

Figure 1: A block diagram of the shared sensor system.

the system. The first is the set of physical *mobile devices* that sense the world. The second is the *virtual sensor layer*. This exists only in the network infrastructure. The third are the *applications* that wish to use the sensor network of mobile phones.

All applications access the virtual sensor layer to obtain sensor data or to program sensing tasks. The physical nodes connect to the network infrastructure as convenient to upload data and may optionally download sensing commands submitted by applications. They may upload data in response to the sensing commands or simply based on what the device user wishes to share. The network infrastructure must populate the virtual nodes and the data streams of the virtual nodes based on the uploaded data. In our current prototype, the network infrastructure is a server hosting a web-service that allows the physical nodes to upload their sensed data (images, video, audio). The virtual nodes are all hosted by this server in the prototype using a database with data samples indexed by location. The location attribute serves as the virtual node address. While our prototype uses a single machine, in a production system, the virtual node hosting workload may be distributed among multiple servers, such as hierarchically organized by geographic regions, sensor modality or other criteria.

The virtual sensor layer can solve several problems in the shared mobile phone sensor network design. First, the addresses of the virtual nodes are naturally location based and the applications do not have to discover or manage the identities of the physical sensor nodes. The application may not care which specific phone captured an image as long as the required image is available to it. Of course, the physical device's identity may be stored as an attribute associated with the data for applications that need it, when the device owner allows sharing their identity. Secondly, this removes the complications due to mobility, since the application no longer tracks the motion trajectories of physical mobile devices to determine which nodes to contact for its sensing needs. Thirdly, the physical device may not always be connected to the network, or may not wished to be contacted. For instance, the device may only upload its data at a convenient time, such as when it has spare bandwidth available. The virtual sensor layer makes this disconnected operation transparent to the applications as they always access the virtual nodes, which are available as long as the network infrastructure is available.

There are many other aspects of the system that present challenging and interesting design problems. A mechanism must be provided for the mobile phones to efficiently provide the collected sample data to the virtual sensor layer. The use of resources in communication and storage of data can be optimized by computing the value or relevance of the data for an application. The privacy of the mobile phone users and the sensed individuals should be respected and the data sharing methods must allow for the privacy sensitivity to vary across individuals and with the user's spatio-temporal context. Since, the system is shared, methods are also needed to ensure the integrity of the data provided by unknown contributors. Another issue is the variability in the quality of data introduced by sensing with different phones passing through the same area. The data may need to be carefully calibrated with respect to each other and changes in data quality within the data stream of a single virtual node must be made known to the applications. A further challenge is to allow efficient programming mechanisms for multiple applications run concurrently. Mechanisms are also needed to allow applications to seek human assistance in their data collection processes. In our prototype, a data publishing client has been developed for mobile phones that uses the server's web service to upload its sensed audio-visual data. The client can be configured to collect data automatically or only when the human user explicitly triggers a data sample capture, such as taking a picture. Mechanisms to program the data collection activity in response to application needs is part of our ongoing work. A prototype of our system that addresses some of these challenges appears in [9]. There are many other possible ways to architect the shared mobile phone sensor network and many design variations even within the above architecture, that are not discussed here. The focus of the remainder of this paper is on obtaining location information for making the volatile sensor layer easy to access for applications.

3. LOCATION

To realize the virtual sensor layer proposed above, a key requirement is that the data samples collected by a phone be location stamped. There are many methods to obtain location on a mobile phone:

1. **Cell-tower triangulation:** The cell-phone network typically knows the location of a phone using signal strength measurements at a phone from one or more cellular base stations. This location is accurate to several meters when the phone is in a region with three or more base stations in range. The location accuracy falls to the granularity of a mile or more when only one base station is in range, such as in rural areas. This location information can be accessed using commercial products such as the Mappoint Location Server [11] when the phone is connected to a cellular network that does make this information visible to non-operator owned applications, such as Sprint in the US, TeliaSonera in Europe, or Bell Mobility in Canada.

2. **Phone GPS:** Many recently released and forthcoming mobile phones have built in GPS receivers and these phones can know their location using the GPS system. The location information is accurate to several meters when the phone has good GPS satellite visibility, such as under open sky. The location is typically not available when indoors.

3. **Wireless LAN triangulation:** Many mobile phones, especially newer models, have built in wireless LAN capability. Location can be obtained using WLAN signal strength based triangulation when multiple access points with known location are in range achieving accuracy of under a meter, or sim-

ply by the location of the access point when only one is in range, achieving an accuracy of several meters. When the location of the access point itself is not known, the location may be estimated from the external IP address for the access point. These techniques are accessible on WLAN connected devices through already available services such as the “Locate Me” feature on Local.Live.com.

4. **Human entered tags:** The phone user may key in tags for all or some of the images taken by her. Some of these tags may include address or landmark information that helps infer the location.

3.1 Content Match and Location

A combination of the above techniques may be applied in practice to location-stamp the samples collected by a phone. However, all these methods have limitations in terms of accuracy and availability. One method to help overcome some of these limitations is to exploit the visual data content itself to enhance the location accuracy for sample points and also to obtain location in scenarios where the above methods are unavailable. The underlying assumption in exploiting the visual data is that two or more images taken at the same location are likely to have some common visual component. This enables the following location enhancing alternatives. For images that have no or highly inaccurate location, such as from human entered landmark tags or from the registered home address for a mobile device, matching the content of the image against other images with known locations can help assign a more accurate location stamp to the image. In situations where the location data is somewhat inaccurate, such as indoor locations when GPS is used, or rural locations when cell-tower triangulation is used, content based matching can help place related images together. For instance, for all images taken within a building, such techniques may help differentiate among images taken in different rooms. We explore the use of visual data content for location enhancement in the latter scenario in more detail.

Note that while content matching helps assign location to visual data alone, correlating the location with the time-stamp of the image also tells us the location of the mobile device at the time the image was taken. Thus, all other data samples from the same device, such as audio data or samples from other sensors connected to the phone, taken with nearby time-stamps, can be assigned a location-stamp as well.

Suppose a large number of images is contributed by several mobile phones participating in the shared sensor network. Suppose next that these images have been separated into multiple virtual nodes based on the location stamps, where images with location stamps within a small threshold distance δ of each other are assigned the same virtual node.

Consider a virtual node corresponding to several images taken within a building. Suppose that mobile phones that used GPS location assigned the last known GPS location, obtained just before GPS satellite visibility was lost - the location of the building entrance. Then, all these images corresponding to different floors and rooms within the building would be assigned to the data stream of this single virtual node. Also included in the same stream would be images that had location stamps within δ of this virtual node’s location but in fact correspond to other locations, such as an image taken a mile or more away but assigned the same location due to cell-tower localization error. Our goal is to show that content matching can be a useful technique to cluster together images that belong to the same location (such as a room) within the building and separate out images assigned to this virtual node in error.

3.2 Algorithms

We use the term zones to refer to the finer granularity regions within a common geographic vicinity. For instance, different shops in a mall, or different aisles in a store, may be termed as different zones. The problem of refining location based on content for a set of images assigned the same locations by the applicable location technology may be broken up into three parts.

First, for each new image that is assigned to this geographic vicinity, we need to determine which zone within that vicinity it belongs to. If the zones are known, then, a suitable matching technique is needed to select the most closely matching zone for the image, and to reject false matches. Each zone may contain multiple existing images, and the new image may yield matches with multiple images in different zones.

Second, the zones themselves may not be known a priori, and may be required to be determined from the image data itself. This is a hard problem since neither the number of zones, nor a distribution of images among those zones may be known.

Third, the zones may have to be associated with geographic location. Matching the images by content and separating them into matching zones does not itself yield information about which zone is located in which physical area of the geographic vicinity.

There are several possible methods to compare the content of image data, such as color histograms, texture matching, and key feature matching. However, none of the content matching techniques is perfect and yield a non-negligible number of false matches. The false match problem is acute in our problem setting since all images being compared do belong to the same geographic vicinity. For instance, images taken in different rooms of a building are likely to contain some similar content. We select key feature matching [10] due to its robustness to lighting changes, image size variation, and imaging device changes. Key features correspond to selected objects in a scene such as corners or peculiar textures that are expected to be preserved across images taken from different points of view and in different lighting. Key features in two images that correspond to the same physical object are likely to match. Fig 2 shows two images with matching key features. It may be seen that while some matches correspond to the same physical objects, there are false matches also.

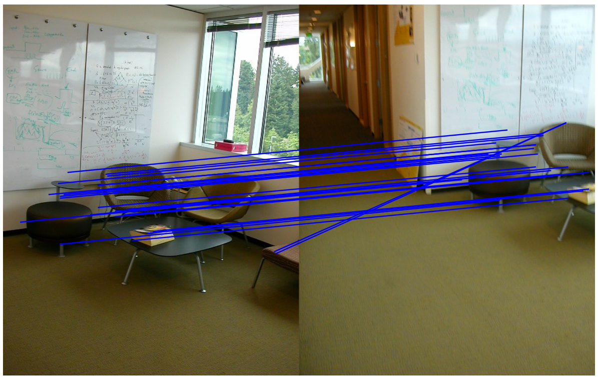

Figure 2: Key featres for content matching.

We focus on the problem of zone discovery. Several methods are already available to match a given image against multiple images but the problem of automatically discovering the zones themselves is sparsely addressed.

We use the following procedures to exploit key feature based content matching for location assignment. Suppose n_i represents the number of key features in an image i and n_{ij} represents the number of matching key features across two images i and j. We define a matching metric $m(i,j)$ between two images based on their key features as:

$$m(i,j) = \frac{n_{ij}}{n_i + n_j} \quad if \quad i \neq j \tag{1}$$

and $m(i,i) = 0, \forall i$.

Zone Discovery Algorithm: Suppose N images have been assigned to a geographic vicinity and we wish to separate these into K zones where K is unknown.

Consider a graph $\mathcal{G}$ with N vertices each corresponding to an image and weighted edges between them, where the weight of the edge between vertices v_i and v_j is m_{ij}. Define a binary relation R, that relates each node to its neighbor connected with the maximum weighted edge:

$$R(v_i, v_j) = \left\{ v_j | \max_{j \in \{1,\ldots,N\}} m_{ij} \right\} \tag{2}$$

subject to m_{ij} being higher than a threshold, m_0. We cluster together images that belong to the same zone by computing the transitive closure, R^* of this relationship for all vertices. The algorithm starts computing a path from a vertex v_i to its neighbor with the maximum weighted connecting edge, until, another node in the same path or in a previously assigned sub-graph (path generated from a previously processed node) is reached. This separates $\mathcal{G}$ into multiple sub-graphs, $g(k)$, each corresponding to a zone. This is a conservative procedure in the sense that images assigned the same zone are likely to belong to the same physical zone, but multiple zones may have been generated for a single physical zone, such as when there is insufficient overlap among images from two different corners of a room. Other possibilities such as based on unsupervised learning, using small training sets, or when K can be determined using alternate means are also of interest, and are part of our ongoing work.

As sensors feed in more data samples, each new image must be assigned to one of the previously determined zones. While it is possible to re-compute the zones each time a new image is obtained, that has a high computational overhead. Instead, the new image may be assigned to one of the existing zones based on its content match. Prior work has considered this problem [17, 7] when an image data-set with known locations is available or constructed from a dense corpus of overlapping images, and test images are compared against this set to determine the location of the test image. Further scalability concerns have been addressed in [12, 15] for matching against large number of images. These techniques are applicable to our problem scenario. However, the problem is simplified in our scenario since rather than comparing a test image against the entire corpus of previously known images, we only need to compare it within a small number of zones of a geographic vicinity and the number of images representing each zone is likely to be small.

As an illustration, consider using key feature based content match. Suppose a small number of images n_k has been assigned to a zone k, where $k = \{1, ..., K\}$. We use the following metric to match a new image i against multiple images in zone k:

$$M_k(i) = \frac{1}{n_k} \sum_{j=1}^{n_k} m(k,j) \tag{3}$$

where $m(k,j)$ is the normalized key feature based match score as defined in (1). The zone assigned to the image is simply the k that yields maximum $M_k(i)$.

The zone assignment process can also be iteratively used to refine the zones determined previously, such as when many images with two or more competing zone matches are found, it may lead to merging the separate competing zones into a single zone. It is worth noting that the binary relation defined in (2) may be asymmetric and this can indeed lead to such merging depending on the order in which the images arrive.

Each zone also needs to be associated with a physical location. This could be achieved based on some of the images for which fine grained location is known, such as through human supervision, opportunistic GPS signal received near windows, or when the number of images is sufficiently large, using geometric 3D reconstruction from the images themselves.

4. EXPERIMENTS

In this section we illustrate some the algorithms proposed above on real image data. We collected 150 images within a building, with a majority corresponding to two distinct zones: a conference room and a kitchen lounge, and the rest arbitrarily distributed across building corridors and an individual office on the same floor. All these images represent image samples contributed by mobile devices that use GPS location and as the GPS signal is unavailable indoors, all these images are assigned the same GPS location - that of the building entrance. Our goal is to explore content matching methods to separate these images into distinct zones within the building.

We first computed the key features and the inter-image match as defined in (1) for all image pairs. Then, we applied algorithm 1 to separate the images into distinct zones. Also, to enable us to measure the algorithm's error, we manually recorded the true zone for each image. The error is defined as follows: for each assigned zone, we check whether all the images belong to the same true physical zone. The true physical zone of an assigned zone is taken to to be the physical zone corresponding to the majority of the images for that assigned zone. Images which do not belong to the corresponding physical zone are counted as erroneous images and dividing this count by the total number of images assigned to the zone yields a normalized error for each assigned zone.

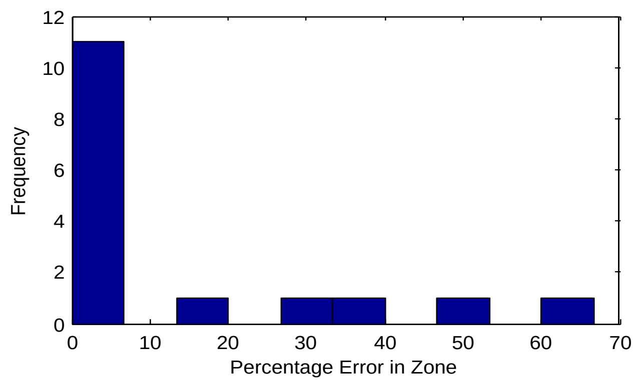

Figure 3: Distribution of zone errors.

We obtained an aggregate error of 12.52% averaged over all zones. However, the error has a high variation, with some zones showing zero error and others showing a very high error. A histogram of the error across 16 zones is shown in Figure 3. It may be seen that while most zones have a very low error, there are some zones with very high error. Manually examining the data corresponding to those zones revealed that these are zones which correspond to physical zones with small number of images. This is understandable as key feature matching is likely to yield fewer true matches in these cases making it more likely for the maximum weight neighbor to be a false match.

Another parameter determined by the algorithm is the number of zones. As mentioned, the algorithm is conservative in combining images into the same zone, causing it to overestimate the number of zones. Multiple zones are generated for the same physical zone when there is insufficient overlap among images. In our dataset, we obtained 16 zones, as opposed to five true zones.

The second aspect illustrated experimentally is the assignment of zones to images when the zones themselves are known. In our data-set, the first 100 images correspond to the two zones with majority of the images. We used 45 of these as a base set which has been divided into their true zones. The remaining images were assigned to the two zones using the metric in (3). Comparing the results with the ground truth, yields 20.01% error, where an error implies that the image was assigned to the wrong zone. More conservative zone assignment methods may be used to reduce the error, at the possible expense of leaving some images unassigned. As the number of images grows, the error will decrease since more overlap and matching regions will exist among images. The results also indicate the limitations of the specific key features used in this implementation for computing image matches, and suggest the need for exploring other content matching techniques to realize our data centric abstraction.

5. RELATED WORK

Several projects have considered the use of mobile devices for building sensor networks. Specifically, the use of mobile phones has been proposed in [16, 13]. Large scale sensor networks using mobile devices dedicated to sensing, and carried by people or vehicles, have also been proposed [1, 8, 4, 3, 14]. Our techniques of using location based virtual sensor nodes rather than application level tracking of the highly mobile physical devices is relevant to all these projects. The use of visual data content for location enhancement is applicable as well, when the mobile sensors have a camera as part of their sensor suites.

The use of similarity among multiple data samples, specifically text documents and images, has also been considered in [6] for data collected by a single individual. Our goal is to leverage data similarities in sensor data from multiple individuals to provide a location based abstraction. Sharing of sensor data is already available at `earthcam.com` for sharing network camera feeds, on `weatherunderground.com` for posting weather sensor data, and `sensorbase.org` for some other sensors. These applications demonstrate the viability of sharing sensor data and also motivate the need for methods that facilitate sharing, such as the ones presented in this paper.

6. CONCLUSIONS

We proposed to use the large number of mobile devices present in our environment as a sensor network. Among many interesting challenges in realizing this vision, we focused on using location information to make this highly volatile and mobile swarm of sensor devices usable by sensor networking application without having to track the device motion trajectories. In addition to using many of the available location technologies, we showed that for audio-visual sensor data, the data content itself can be exploited to refine location granularity. Even when the physical location cannot be determined, clustering together images which correspond to the same zone within a larger geographic vicinity can be advantageous for application interested in sensing a specific zone. While the content based techniques are directly applicable to visual data, correlating the time-stamps and device identities across samples can allow the network infrastructure to organize samples from other sensing modalities into location based virtual nodes as well. We have presented some of our very initial experience in organizing a mobile device based sensor network. This domain presents several interesting research problems, and the audio-visual nature of the sensed data can be exploited in solving some of them. Future work includes exploring sophisticated content matching, such as based on combination of multiple image features, the use of motion trajectories learned from motion patterns, and correlation of other sensor data in inferring user location.

7. REFERENCES

[1] T. Abdelzaher, Y. Anokwa, P. Boda, J. Burke, D. Estrin, L. Guibas, A. Kansal, S. Madden, and J. Reich. Mobiscopes for human spaces. *IEEE Pervasive COmputing*, 6(2):20–29, 2007.

[2] Y. Benkler. Coase's penguin, or, linux and the nature of the firm. *Yale Law Journal*, 112, 2002.

[3] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. Srivastava. Participatory sensing. In *ACM Sensys World Sensor Web Workshop*, Boulder, Colorado, USA, October 2006.

[4] S. Eisenman, N. Lane, E. Miluzzoand, R. Peterson, G. Ahn, and A. Campbell. Metrosense project: People-centric sensing at scale. In *ACM Sensys World Sensor Web Workshop*, Boulder, Colorado, USA, October 2006.

[5] Flickr API. http://www.flickr.com/services/api/.

[6] J. Gemmell, R. Lueder, and G. Bell. The mylifebits lifetime store. In *ACM SIGMM 2003 Workshop on Experiential Telepresence (ETP 2003)*, November 2003.

[7] J. Hong, X. Tan, B. Pinette, R. Weiss, and E. Riseman. Image-based homing. *IEEE Control Systems Magazine*, 12(1):38–45, Feb 1992.

[8] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden. CarTel: A Distributed Mobile Sensor Computing System. In *4th ACM SenSys*, Boulder, CO, November 2006.

[9] A. Kansal, M. Goraczko, and F. Zhao. Demo: Building a sensor network of mobile phones. In *ACM International Symposium on Information Processing in Sensor Networks*, April.

[10] D. G. Lowe. Object recognition from local scaleinvariant features. In *Proceedings of international conference on computer vision*, page 11501157, Greece, September 1999.

[11] Microsoft mappoint location server. http://www.microsoft.com/mappoint/products/locationserver/.

[12] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, 2006.

[13] C. Ratti, A. Sevtsuk, S. Huang, and R. Pailer. Mobile landscapes: Graz in real time. http://senseable.mit.edu/graz/.

[14] O. Riva and C. Borcea. The urbanet revolution: Sensor power to the people! *IEEE Pervasive Computing*, 6(2):41–49, Apr-Jun 2007.

[15] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *CVPR '07: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[16] Nokia sensorplanet. http://www.sensorplanet.org/.

[17] C. Silpa-Anan and R. Hartley. Localisation using an image-map. In *Proceedings of the 2004 Australasian Conference on Robotics and Automation*, Canberra, Australia, December 2004.

[18] SlamXR. http://www.msslam.com/slamxr/slamxr.htm.