Enabling Data Interpretation through User Collaboration in Sensor Networks

Eunjoon Cho, Kevin Wong, Branislav Kusy, and Leonidas Guibas
Stanford University, Stanford, CA, USA
kusy@stanford.edu

Abstract

In order for sensor network deployments to be successful, users need to be able to derive relevant and interesting information from sensor data. Information rich multimedia data can provide high utility to users, but their interpretation is hard in resource constrained domains. We note that humans can derive information from images efficiently and propose a multi-tier camera sensor network that combines simple in-network data interpretation with a lightweight user recommendation system. We explore ways of how to decrease demands on an individual user and to balance the benefits gained from our system with the required work that needs to be put in. One of the advantages of this approach is that its cost-benefit ratio will improve as the system grows in scale, enabling techniques from large social networks, such as data recommendation based on user similarities.

1 Introduction

One of the main objectives of sensor networks technology is the extraction of useful information from the environments. In the past, users were presented with graphs that summarize long-time trends of data measured by limited sensors, such as humidity or temperature sensors. The availability of low-power microphones and CMOS cameras has recently enabled sensors to capture information-rich multimedia data. As a result, advances in assisted living [10, 16], animal species monitoring [5], and traffic monitoring and control [6] have been achieved. However, processing of this typically high bitrate data is challenging in resource constrained domains. In addition to significant energy requirements of transporting multimedia data to users over low power multihop wireless links, computationally intensive algorithms are required to extract useful information from the data.

The fact that humans can extract information from multimedia data efficiently motivated the design of the USer Interpretation CAmera SEnsornet (USICASE) system presented in this paper. The novel aspect of our system is that it allows users to collaboratively extract information from real-time or historical sensor data, focusing specifically on camera sensors. Users attach their interpretations to the images through textual annotations, and the system proactively recommends the interpreted data to other users if the data matches their interests. Some plausible application scenarios are the notification of research group members about a joint lunch in cafeteria or the availability of conference rooms; taking images of an ongoing experiment in a bio-chemistry lab and allowing researchers to discuss its progress using real-time data; or notifying security personnel if an unfamiliar person or object is observed in the office space.

One of the main challenges of our system is to carefully balance the effort that users need to put in and the utility that the system provides. USICASE implements a number of techniques that help to focus data interpretation efforts. Each user defines his region and time of interest as well as a set of keywords expressing his work or hobby related interests. USICASE uses in-network processing techniques to detect changes in the regions of interest and delivers images capturing such changes to the users for interpretation. In return, each user gets a sensor data *Inbox* where he receives annotated sensornet images that match his interests and are constrained to his region and time of interest. Additionally, the system allows each user to retrieve data from available sensors through an intuitive map-based interface.

We have implemented and tested USICASE on a small testbed deployed in our office space. Motivated by the fact that our system reaches its full potential when it grows to large size, we designed its architecture in multiple tiers. The system consists of camera sensor nodes, micro-servers, and a central server. Sensor nodes provide cost-efficient dense sensing, micro-servers enable network to scale, and the central server orchestrates all components by resolving user queries and maintaining information about sensors, micro-servers, users, and annotated images in a central database.

The growing popularity of mobile computing devices makes them an ideal platform for accessing data from sensornet deployments. Since interfaces of these devices are somewhat limited, our main design criterion for the user annotation process was simplicity. We achieved this goal by constructing an ontology tree of data annotations that contains both annotations and relations between them. User is initially presented with a small set of general tags. Depending on the tag selection, the system suggests more specific tags until the user is satisfied, potentially significantly decreasing the time required to annotate images.

We evaluated the scalability of USICASE by measuring the latency of the end-to-end data delivery of data to users and the run-time of various in-network processing algorithms. The system supports multiple parallel data streams within a cluster without the users incurring additional penalty. Both in-network compression and data processing take approximately 1 second per image. We also evaluated the accuracy of image processing in capturing events of interest in the environment. In an 83 minute experiment with a camera pointed at a dynamic environment, our system was able to identify 6 out of 10 major events while successfully filtering out over 97% of motion detection events. Since any meaningful evaluation of the recommendation system would require larger deployment and more users, we leave it for future work.

2 Related Work

A number of wireless sensor network (WSN) systems with multimedia sensors were deployed in the past few years. Cyclops [14] was perhaps the first mote-class compliant device that enabled image capture in resource constrained domain and identified a number of lightweight image processing techniques successfully used for object detection and recognition. However, both the low image resolution and the low speed of its embedded MCU limited the types of applications that the Cyclops platform could support. Higher end camera sensor nodes such as CMUcam3 [15], or EnalabCam [1] have higher power demands, but provide significantly better image resolution and processing.

Multi-tier architecture is a popular design of camera sensor networks, because it combines benefits of both the dense sampling with low-end cameras and the better data fidelity of high-end cameras. For example, Xie et al. [16] have coupled Cyclops cameras with the higher-end EnalabCam. Helping elderly people in finding lost objects, the system uses low power cameras to detect new objects in the scene and then wakes up the high-end camera node to recognize the object. SensEye [9] used even higher-end pan-tilt-zoom (PTZ) cameras, to provide additional coverage.

Querying languages utilizing in-network processing to extract information from WSNs have been studied extensively. TinyDB [11] views WSN as a large distributed database and offers a simple SQL-extended language for energy- and bandwidth- optimized acquisition and collection of sensor data. Going one step further, Deep Vision system [13] proposed a declarative querying language that encapsulates low-level image processing operations in the language attributes. Users can extract high-level information from the network, such as probability of intruder in a monitored area by specifying how and when should data for the queries be collected. In our work, we proposed similar techniques to detect events of interest in the environment. However, we focused on the simplicity of the user interface and provide map based interface to retrieve data from sensors, instead of a querying language.

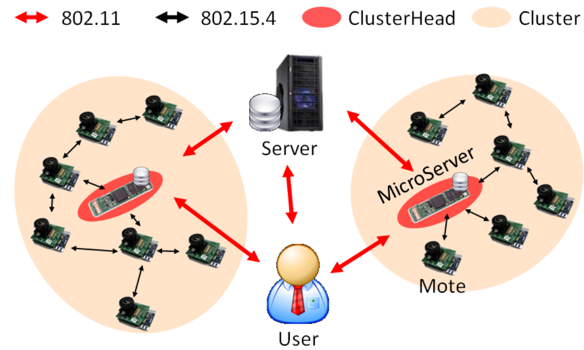

Figure 1. A multi-tier camera sensor network. Motes are organized in ad-hoc clusters. Users access data through both the server and micro-servers.

Substantial body of literature on recommendation systems exists, most of which focused on content-based and collaborative recommendation [4]. Also, ontology trees have been studied [12] to increase the performance of recommendation systems. However, most of these approaches target Internet-based web applications and their applicability to the WSN domain is yet to be studied.

3 USICASE System Overview

We describe USer Interpretation CAmera SEnsor Network (USICASE) system, a sensor network service that allows users to retrieve information rich historical and real-time sensor data related to their interests. The novel aspect of our system is its support for user collaboration in interpreting sensor data.

The utility of USICASE improves with an increasing number of users, as they share the task of data interpretation. Consequently, we designed it as a multi-tier system, consisting of Tier-1 camera sensor nodes, Tier-2 micro-servers and a centralized server (see Fig. 1). Similar configurations were used in camera network deployments in the past [5, 9] due to their numerous advantages: they allow sensornets to run advanced processing algorithms at Tier-2 devices and to scale well due to the higher network capacity of Tier-2 radios. Next, we describe our choice of hardware and software in more detail and describe how the system operates.

3.1 System Components

Tier-1. Our sensor nodes must provide dense and unobtrusive sensing of the environment and therefore should have low cost and small size. We have selected a higher-end sensor node, the iMote2 [3], to be able to process and compress images at the sensor node level. iMote2 platform offers dynamically scalable voltage (from 13MHz to 416MHz), 256KB of SRAM, 32MB of SDRAM, 32MB of Flash memory and 250kbps CC2420 radio. We equipped iMote2s with EnalabCam [1] which can capture QVGA(320x240) images at 60fps. Fully active, iMote2 consumes about 140mW or 240mW at 13MHz or 104MHz, respectively. Both radio and the camera board consume additional 40mW each. We programmed the iMote2s in TinyOS [8].

Tier-2. The micro-servers run advanced image processing algorithms, maintain database of sensors in a cluster, and provide higher bandwidth communication. Since transferring image data over the low bandwidth Tier-1 radios takes substantial time, clusters of Tier-1 nodes should be relatively small, requiring the micro-servers to be low cost as well. We have used Gumstix Verdex XL6P [2] computers which provide 600MHz CPU, 128MB RAM, and Wi-Fi, USB host, and MicroSD extension cards. Gumstix supports Linux operating system which allowed us to use open source libraries for image processing, database management, and web frameworks. Micro-servers communicate with Tier-1 nodes via a USB-attached gateway sensor node.

Central Server. The central server is a PC class device, running the Linux operating system. The main role of the server is to maintain the central database that holds all of the sensor data collected in the past as well as information about the subscribed users. Users primarily connect to a web server running on the central server via an Internet connected device. Each user has a personalized sensor data *Inbox*, where the system pushes data relevant to his interests as well as new data that needs to be annotated.

3.2 System Operation

The main objective of the USICASE system is to deliver relevant sensor data to users in near real-time. To accomplish this goal, the system needs to 1) obtain information about users, 2) extract contextual information from sensor data, and 3) match and deliver data to the users.

Information about users is collected primarily during user subscription process. Each user creates a profile by selecting *regions of interest (RoI)* and *times of interest (ToI)*. The user also specifies his interests through a set of data annotations which are flickr-style textual tags described in Section 4.2.

Interpretation of sensor data is primarily user-driven. Images are periodically collected at sensors whose *region of coverage (RoC)* intersects with *RoI* of at least one user. To prevent sending users large number of images, USICASE only asks users to interpret and annotate new data if the environment in their *RoI* changes. The system uses a suite of image processing algorithms to detect the environment changes which are described in more detail in Section 4.1.

There are two ways users get information from USICASE. First, the central server determines relevance of each new data item against all user profiles using a data matching procedure described in Section 4.3. If the relevance is above certain threshold, data is pushed to the user's *Inbox*. Second, the system allows users to request data from any available sensor. Instead of providing an extension of SQL language to query data from the network, USICASE provides a map of the deployment area where users can intuitively select the sensor node and the type of data they want to retrieve.

4 Data Interpretation and Matching

We describe the techniques that USICASE uses to interpret and match data to users in this section.

4.1 In-network Image Processing

Due to the limited processing power of sensor nodes, we decided to partition the data processing between the Tier-1 sensor nodes and the Tier-2 cluster heads. In Tier-1, we chose computationally simple image processing algorithms to highlight possible regions of interest in each frame. Once these regions are passed to Tier-2, we performed additional processing by generating descriptors for each such region.

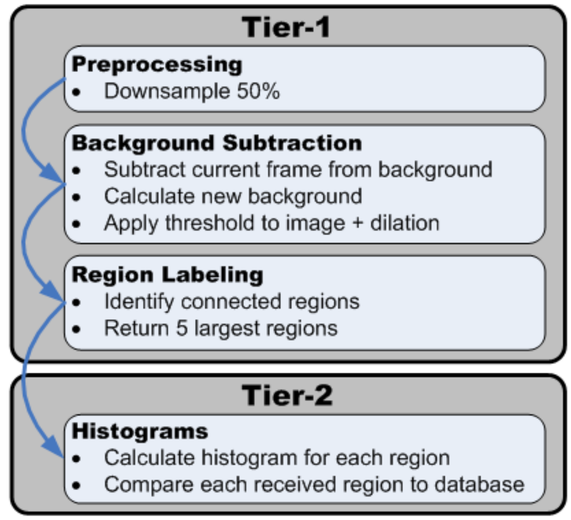

Figure 2. Flowchart of USICASE's image processing algorithm.

Tier-1 Processing. We primarily focused on detecting moving objects in view of our cameras. We used background subtraction to detect changes from frame to frame and then applied region labeling to both recognize moving objects and to filter these objects by their pixel size. All processing was done using a sub-sampled gray-scale image, converted from the original QVGA RGB image.

We used moving an average filter in our background subtraction algorithm, defined by Eq.(1), with $\alpha = 0.125$.

$$B_{n+1} = \alpha B_n + (1-\alpha) F_n \tag{1}$$

Before calculating the next background image, we subtracted the background from the current frame to obtain a difference image. We then applied a threshold based on the absolute average difference so areas with either small or large changes from the background will be adequately recognized. Next, we applied a binary morphological dilation operation using a 3 by 3 square structuring element to close small holes and connect close regions together. To identify regions of interest, we used a standard 4-neighbor region labeling algorithm to find the largest connected areas of our difference image [7]. We discarded all regions smaller than 150 pixels and returned five largest regions to the cluster-head for further processing and user annotation.

Tier-2 Processing. After the Tier-1 devices forward the region data, we calculated the RGB histograms of the possible regions of interest. We made the assumption that objects that were in motion were distinctive from the background and had unique histograms. We then compared these regions against those in the database of recent images by taking the Euclidian distance between the histogram vectors to determine their similarity. Similarity of histograms of regions allowed us to classify regions as objects that we can track across time and space in our camera network.

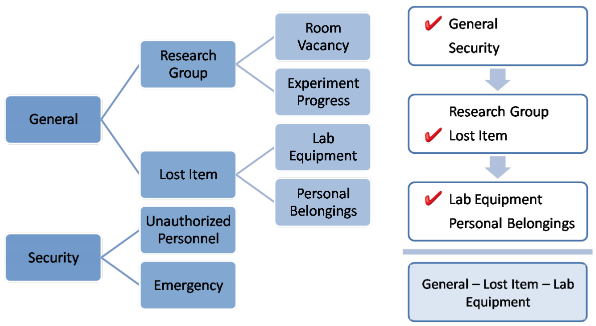

Figure 3. Ontology tree of annotations (left) and the annotation process (right).

Table 1. User profile, used to calculate personal and collaborative recommendation scores.

	Category	Data	
Attributes	**Name**	Eunjoon Cho	
	Password	*******	
	Title	Graduate Student	
	Affiliation	Electrical Engineering	
	Clearance	Low	
Interests	**Category**	**Data**	**Weight(0-5)**
	Region	Room S255	4
	Time	19:00 - 04:00	1
	Tags	Security	5

4.2 User Annotations

As described in the previous section, USICASE allows users to attach interpretations to sensor data through textual tags. Since our system may be used in time and resource constrained environments, the image interpretation user interface needs to be efficient. Therefore, instead of requiring users to type in the interpretations, the system provides users a concise set of predefined data tags. We derive the tags from a predefined ontology tree of annotations with general tags at the root and more specific tags at the leaves. The tree itself is not shown in the annotation process, users are typically shown a few entries from the same level in the tree. For every selection, the process iteratively provides the next set of tags, generally following specialization relation in the tree (see Fig. 3). USICASE also provides an option for the user to directly enter annotations, thus satisfying the requirement of an efficient yet comprehensive annotation system.

4.3 Recommending Sensor Data to Users

We use two sensor data recommendation schemes. *Personal* recommendation delivers data according to the interests i specified by each user in his profile (see Table 1). *Collaborative* recommendation delivers data that matches another user's interest, in case the two users share similar attributes. We measure user similarities by finding correlations in the *Attributes* sections of their user profiles, using information such as job title, affiliation, and clearance.

Personal recommendation. Each piece of data is matched with each user's interests and given a value of 1 or 0 to indicate a match m_i. This is weighted with a factor w_i, that specifies how much the user is interested in that specific category. Eq. (2) defines the personal recommendation score for user u as the normalized weighted sum of all matching interests.

$$s_p(u) = \frac{\sum_{i=1}^{N} m_i w_i(u)}{\sum_{i=1}^{N} w_i(u)} \quad (2)$$

Collaborative recommendation. Users with similar attributes are likely to be interested in the same data. This allows our system to recommend data to larger number of users, including new users that have not yet specified any *Interests* in their profiles. Similarity of users u and u', $sim(u,u')$ is simply calculated as the ratio of the tags in *Profiles* of the users that are the same. To determine the collaborative recommendation, we use the weighted sum approach described in [4], utilizing Eq.(3). The personal recommendation score of the data of user u' is weighted with the similarity measure between users u' and u. U is the set of all users whose similarity to user u is greater than 0. The final recommendation is then defined as the sum of the weighted recommendations of other users, normalized by the sum of the similarity measures of all similar users.

$$s_c(u) = \frac{\sum_{u' \in U} sim(u,u') s_p(u')}{\sum_{u' \in U} sim(u,u')} \quad (3)$$

Combined recommendation. The scores from personal and collaborative recommendation are adjusted with a collaborative factor $\alpha(u)$, to indicate how much a user prefers to rely on his personal settings over others. The total score, Eq. (4), is then compared to a threshold to decide whether the data is to be pushed to the user.

$$s(u) = (1-\alpha(u))s_p(u) + \alpha(u)s_c(u) \geq s_{th}(u) \quad (4)$$

5 Evaluation

We evaluated how well the USICASE implementation supports the system's goals, namely delivery of sensor data to users and in-network identification of relevant data.

5.1 USICASE Architecture

To test the performance of our system, we requested images from one, two, and three sensors (in parallel) located in the same cluster. We measured how much time each sensor spent to acquire image from the camera board (IMG_ACQ), execute the motion detection algorithm (IMG_PROC), compress the image (JPEG), and route the data to its cluster-head (ROUTEx). We also measured how long the cluster-head takes to save data in the central database (MYSQL). We repeated each experiment 10 times and present the results in Table 2, denoting different experiments as ROUTEx where x is the number of simultaneous data streams. We requested both gray-scale and color images, with the average compressed size of 11.3kB and 26.5kB, respectively. As we see in Table 2, our transport protocol assures fairness to multiple users with a negligible drop in performance for multiple parallel streams. We also experimented with retrieving images from different clusters and have not observed any significant

Table 2. Statistics of reliable delivery of QVGA images from sensors to a user. We show mean times in seconds and standard deviations in parenthesis.

	Time (GRAY)	Time (RGB)
IMG_ACQ	0.9 (0.0)	0.3 (0.0)
IMG_PROC	n/a	1.4 (0.3)
JPEG	0.3 (0.001)	1.0 (0.003)
ROUTE1	5.8 (0.1)	13.8 (0.1)
ROUTE2	6.0 (0.1)	14.1 (0.2)
ROUTE3	6.0 (0.6)	14.3 (0.4)
MYSQL	1.4 (2.6)	1.6 (3.2)

increase in the delivery times as nodes in different clusters use non-interfering radio channels.

5.2 In-network Processing

To evaluate our in-network processing algorithms, we collected 500 images of our workspace over a period of 83 minutes. We established a ground truth by manually identifying 10 key events, such as people entering and leaving the camera's field of view. Movement outside these key events consisted of people working on computers with minimal movement. Our motion detection algorithm located 1264 events which was further decreased to 35 events by histogram matching, containing 6 of our key test events. This illustrates the trade-offs between the detection accuracy and the amount of work that users need to expend.

Our image algorithm was able to correctly identify events where people left the area, but it was not as accurate in determining when a person returns. For example, if a person returns between frames and resumes working, the histogram matching would treat this as an already classified object and ignore the event. Other spurious events included small movements within the image, which a human typically would not consider as important, or large changes in lighting, which histograms can not capture. Our pruning processing was simple by contemporary standards, and could be improved by using descriptors which are more robust against object illumination changes, rotation, or scaling.

6 Discussion and Future Work

We have designed, implemented, and deployed a camera sensor network service that allows users to work together to extract information from images collected by wireless camera sensors. Since the users are the primary information providers, we have proposed a combination of in-network processing techniques and an efficient data annotation interface to decrease the load on an individual user. We have designed our system to be scalable in size, observing that the cost-benefit ratio of our system will improve as more users join in and share the data interpretation load.

Our work can be extended in a number of exiting ways. We would like to allow users to rate how well our system estimates relevance of the data and to rate the data itself. Through correlations of ratings of different users, we would be able to improve the collaborative recommendation capabilities of our system. Using high level image processing techniques techniques such as object recognition and tracking, we could match objects and automatically transfer user annotations between images containing identical objects. Object recognition would also allow users to retrieve all images with similar objects, track a mobile object both spatially and temporarily across multiple cameras, or specify a higher level query that could return objects that exhibit a specific mobility pattern between rooms or cameras. Finally, we would like to further improve the annotation process by customizing the ontology tree to a particular user or data item and by using relations other than specialization for annotation suggestions.

Acknowledgement

This work was supported in part by Army AHPCRC grant W911NF-07-2-0027-1 and NSF grant CNS-0626151. The authors would also like to thank HyungJune Lee and Martin Yoon-Wicke for their valuable comments.

7 References

[1] Enalab camera. `http://enaweb.eng.yale.edu/drupal/`.

[2] Gumstix. `http://gumstix.com/`.

[3] iMote2. `http://embedded.seattle.intel-research.net`.

[4] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, June 2005.

[5] S. Ahmadian, T. Ko, S. Coe, M. Hamilton, M. Rhimiand, S. Soatto, and D. Estrin. Heartbeat of a nest: using imagers as biological sensors. *Center for Embedded Network Sensing, Tech. Report, Paper 1*, 2007.

[6] J. Campbell, P. B. Gibbons, S. Nath, P. Pillai, S. Seshan, and R. Sukthankar. Irisnet: an internet-scale architecture for multimedia sensors. In *MULTIMEDIA '05*, pages 81–88, New York, NY, 2005.

[7] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision, Volume I.* Addison-Wesley, 1992.

[8] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister. System Architecture Directions for Networked Sensors. In *Architectural Support for Programming Languages and Operating Systems*, pages 93–104, 2000.

[9] P. Kulkarni, D. Ganesan, P. Shenoy, and Q. Lu. Senseye: a multi-tier camera sensor network. In *MULTIMEDIA '05*, pages 229–238, New York, NY, USA, 2005. ACM.

[10] D. Lymberopoulos, T. Teixeira, and A. Savvides. Detecting patterns for assisted living using sensor networks: A case study. *SensorComm '07.*, pages 590–596, Oct. 2007.

[11] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TinyDB: An acquisitional query processing system for sensor networks. *Transactions on Database Systems (TODS)*, 2005.

[12] S. E. Middleton, N. R. Shadbolt, and D. C. D. Roure. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems*, 22:54–88, 2004.

[13] M. Rahimi, S. Ahmadian, D. Zats, J. Garcia, M. Srivastava, and D. Estrin. Deep vision: Experiments in exploiting vision in wireless sensor networks. Submitted for publication., 2006.

[14] M. Rahimi, R. Baer, O. I. Iroezi, J. C. Garcia, J. Warrior, D. Estrin, and M. Srivastava. Cyclops: in situ image sensing and interpretation in wireless sensor networks. In *SenSys '05*, pages 192–204, 2005.

[15] A. Rowe, A. Goode, D. Goel, and I. Nourbakhsh. Cmucam3: An open programmable embedded vision sensor. Technical Report RI-TR-07-13, Carnegie Mellon Robotics Institute, Pittsburgh, PA, 2007.

[16] D. Xie, T. Yan, D. Ganesan, and A. Hanson. Design and implementation of a dual-camera wireless sensor network for object retrieval. *IPSN '08*, pages 469–480, April 2008.