Multimodal Observation Systems

Mukesh K. Saini
National Univ. of Singapore
mksaini@comp.nus.edu.sg

Vivek K. Singh
University of California Irvine
singhv@uci.edu

Ramesh C. Jain
University of California Irvine
jain@ics.uci.edu

Mohan S. Kankanhalli
National Univ. of Singapore
mohan@comp.nus.edu.sg

ABSTRACT

In recent years, we have seen a significant research interest in a number of multimodal sensing applications like surveillance, video ethnography, tele-presence, assisted living, life blogging etc. However, these applications are currently evolving as separate silos with no interconnection. Further, the individual application-centric architectures typically tend to focus on specific sensors, specific (hard-wired) queries and deal with specific environments. We present a generic sensing architecture 'Observation System', which allows multiple users to undertake different applications through abstracted interaction with a common set of sensors. The observation system observes behavior of various objects in an environment and keeps a record of important events and activities in an eventbase. In this system, multifarious data collected from disparate sensors and other sources are correlated to understand and gain insights in the environment. The observation system has applications in many areas including but not limited to surveillance, traffic monitoring, ethnography, marketing, and healthcare. In this paper, we present the architecture and functionality of such a system and present details of activity detection using multiple sensor streams in a distributed sensing environment. We also present results of such an approach and potential extensions to the analysis of more complex activities and events.

Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**]: methodology

General Terms

Design, Security, Management, Performance

1. INTRODUCTION

Over the past few years, there has been a growing amount of research conducted in multimodal sensing applications like surveillance, video ethnography, tele-presence, assisted living, life blogging etc. However, these applications are currently being developed as isolated silos with no interconnection. For the same sensed environment, one can find two distinct sets of sensors and differing application-centric system architectures to support two applications, say surveillance and tele-presence. We propose the use of 'Observation System' which is a generic sensing architecture allowing multiple users to undertake different applications with a common set of sensors through a *loosely-coupled* event-based architecture. Thus, observation system (ObSys) is the highly *scalable* and *flexible* super-set of many current media sensing applications. Further, the sensing architecture is generic and can be easily ported over to multiple sensing environments with very little effort.

In order to develop observation systems, we propose two significant deviations from the standard sensing architectures. First, we propose a *loosely-coupled* sensing paradigm wherein the applications do not interact directly with individual sensors but rather with a common observation database which in turn is responsible for collecting information from all available sources.

Secondly we employ an event-based architecture for processing of various queries. This is because *dynamics* of the environment in terms of the *events* are just as important as the (conventionally focused) *statics* of the environment in terms of *objects* present. A combination of dynamic and static elements is handled in our observation system using the 'environment model' and the 'dynamic eventbase'.

To illustrate the salient points of the observation system architecture, we describe two prototype implementations in two different environments each supporting multiple diverse applications. In the first implementation, an observation system working in a university building is used to simultaneously support surveillance and video-ethnography applications. The second implementation is an observation system working in an office environment supporting surveillance, tepresence, and life-blogging views simultaneously.

We highlight the features of generic observation systems in Section 2. In Section 3 we discuss related work. Two prototype implementations of the observation system are presented in Section 4. We present outlook and conclusions in section 5.

2. OBSERVATION SYSTEM: FEATURES

An architectural overview of the ObSys is presented in Fig. 1. In order to make ObSys functional, the first step is to 'set up the environment'. This would involve the placement of sensors at the appropriate positions, registering them with an 'Information Source Registry' and capturing the 3-d physical information of the environment in an 'environment model'.

This is followed by the 'data-acquisition and assimilation' step where the data is captured from multiple sources including sensors and other non-sensory inputs. The information coming from multiple sensors needs to be assimilated to create a macroscopic view. The next stage is 'Data organization and query processing',

MM'08, October 26–31, 2008, Vancouver, British Columbia, Canada.

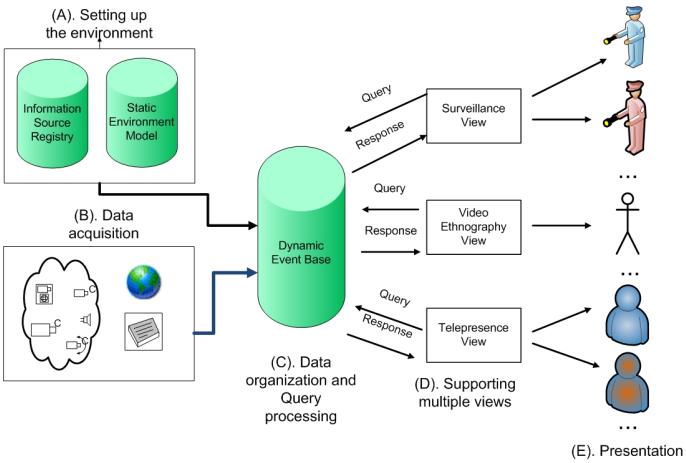

Figure 1: Observation System: Functional architecture

wherein the data indexing, storage etc. is undertaken to make data usable for the querying, retrieval etc. The queries are first defined objectively at an 'elemental level', and later combined to 'domain level' forms to 'support multiple Views/Applications'. The last step is presenting the relevant data to the different clients which may have different security settings and needs in the 'presentation' step.

2.1 Setting up the environment

In this step we place all the required sensors into the environment. Optimal placement and sensor selection issues also need to be handled. On the addition of each sensor, the system must update an ISR (Information Source Registry) which will keep track of the sensor position, type (video, audio etc), orientation etc. The physical description of the environment is also explicitly captured to create an 'Environment Model' (EM). The explicit inclusion of ISR and EM in the architecture, allows the ObSys to work in different physical setups and sensor configurations.

2.2 Data acquisition and assimilation

Data is obtained from multifarious relevant sensors. Non-sensory data sources e.g. time-clock, the web, calendar of events etc. also need to be included wherever relevant. Different confidence and agreement metrics between sensors are used for information assimilation. Further, microscopic views coming from distributed locations in a large scale sensing environment are combined into larger macroscopic views. Note however that this assimilated information should be in an objective form and still be application independent.

2.3 Data organization and Query processing

All raw data should be stored as-is whenever possible. The analysis, interpretations, labels etc. should also be saved separately in the eventbase but not mixed with the data. Irrecoverable losses for compression and abstraction's sake should be delayed as much as possible. We identify 3 different stages where events may occur and hence queries may be formulated.

Data level queries (events) are those which can be formulated using a single sensor/data source e.g. sensor(i).BlobSize and sensor(j).AudioVolume etc., and we expect that such low level detectors would be available as plug-n-play ActiveX components, DLLs or even as part of hardware drivers in near future. These queries would not be directly used by end-users (e.g. security guard or nurse) but rather be used by system administrators to create higher level queries.

Elemental queries would be queries like 'trajectory' detection which would be based on multiple data-level events. Thus they may involve multiple sensors. Note that different administrators may define the same elemental query (e.g. 'trajectory') using different sensors or data level queries. Also, these queries would still be independent of application context. Hence the same elemental level event, 'tracking' of a person, can be translated into 'intruder tracking' for a surveillance application and 'user activity pattern' query for an ethnography application.

Domain/Application level queries are the basic units for end user's interaction with the system. These are created by using one or more elemental level queries in a specific application context. Thus, following the discussion in previous paragraph, 'intruder tracking' and 'user activity pattern' are application level queries.

2.4 Supporting multiple Views/Applications

Explicit mapping of same data level queries into different elemental and application queries for supporting different applications are shown in Fig. 2 (Surveillance and Video ethnography views) and Fig. 5 (Surveillance, Telepresence and Lifebloging views). As can be seen, the loosely-coupled layered architecture allows multiple application views to be created using same sensors/data sources.

The query composition task is handled by an administrator who employs a domain based ontology for creating such queries. Specific user requests for newer queries are periodically handled by the administrator, who can assign new permissions, add sensing resources, compose new application queries using existing data level/ elemental events or deny such requests. The inspiration for this kind of architecture comes from Enterprise Resource Planning systems etc. where a core is developed and specific user-required features can be added/deleted periodically by an administrator.

2.5 Presentation system

Information must be presented to a user based on nature of data to be presented, client characteristics, and access rules. Thus sophisticated multi-display devices as well as PDAs can be used for presenting information coming from an ObSys.

3. RELATED WORK

We notice that there is significant research happening in each of the individual sensing applications. Surveillance in particular is a fairly well-developed area, however, most of the works still involve a very tight coupling between the specific surveillance task required and the sensing strategy employed [11]. A few works have started to recognize the need for a generic architecture which could be employed for multiple surveillance scenarios. For example, a preliminary effort to move away from traditional transactional surveillance model to the observational model is shown in the IBM:S3 system[4].

We found many interesting works in other media sensing applications like telepresence [6], assisted living [7], video ethnography [8], video life blogging [9] etc. However each of them focuses on only their specific application and no effort at a generic framework has been made.

There have been some efforts like [10, 3] to create frameworks for handling generic query formulation across different variations *within* the surveillance domain. While this idea is useful, we want to build a generic system *across* multiple media applications. Looking from a sensor network perspective, there have been attempts at looking at the entire sensor networks as a Database and then posing queries to them. Works like [1] and [2] describe how networking components can be abstracted (and separated) from the sensing component. While again, the basic ideas are interesting, clearly the

focus of their works is very different as we do not focus on networking and resource constraint issues and rather focus on event based architecture for rich media streams.

As mentioned earlier, we are leveraging on the event based architecture for creating generic systems. Event based queries have been explored earlier in our previous works like [12]. We have also argued for handling heterogeneous databases for answering rich query sets in [5]. In this work however, we extend the ideas to utilize a heterogeneous event based architecture for supporting multiple application views at the same time.

4. PROTOTYPE IMPLEMENTATIONS

In this section, we describe our efforts at creating an observation system for one outdoor and one indoor scenario. For the outdoor implementation, we create an observation system for a multi-floored building (Donald Bren Hall) at the University of California, Irvine using 8 hand picked cameras as information sources. We develop an 'environment model' of the building which has 3-D model of the building with important objects and their attributes. All sensors and important points of interest are registered in the model. In this implementation we use two data level event detectors viz. blob (size/position detector) and motion-vector detector. These were combined with current-time non-sensory data for the detection of various elemental and application level events. A summary of these translations is shown in Fig. 2.

In case of emergency evacuation event, most cameras capture multiple people running toward the exits (Fig. 3). Similarly abnormal event is detected if the motion trajectory of the person differs from others above some threshold and intrusion is detected by combining contextual information (time of the day and restricted areas) with sensory information i.e. presence of the person in the restricted areas. For Video Ethnography application, spatial activity patterns and specific area action patterns are extracted to detect different events listed in Fig. 2. A sample result for floor 1 is given in Fig. 4.

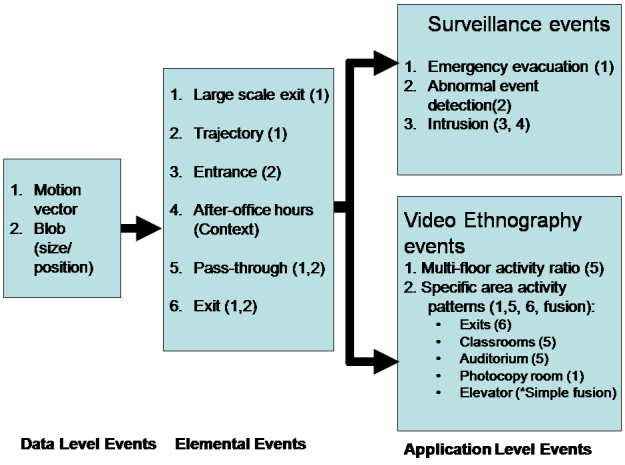

Figure 2: Event classification and their interrelation for outdoor environment experiments. The numbers in parentheses denote the immediate lower level event used to detect the event.

For indoor implementation, we create an office environment in the lab in National University of Singapore. The office has one desk, one white board, one shelf, one visitor chair, and one telephone. The mock office has only one entrance. The designated hours are assumed to be from 10am to 8pm. For this setup we

Figure 3: Evacuation event detection. Dark rectangle at the upper left corner in each image represents *evacuation* status.

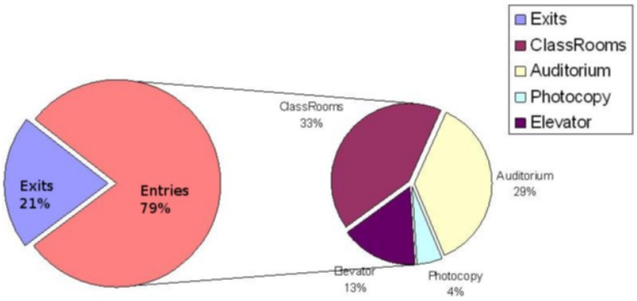

Figure 4: Different activities undertaken by users on floor 1

use 6 cameras and 1 microphone to observe the office. Following same philosophy as in outdoor case, 'Environment Model' is created using 3-d model of the office and important point of interest are registered. Both audio and video sensors are registered in the model. Additional non sensory information used is current time. In the indoor implementation, we use three data level events viz. blob detection, face detection and sound energy detection. These events are then used to detect elemental level events. Elemental events are used to make higher level inferences to detect application level. It can be noticed that same set of data and elemental level events is used to make different interpretations for Surveillance, Telepresence, and Life Blogging. The details of the three types of the events and their interrelations are shown in the Fig. 5.

We can select the application view at run-time using the tabs given on GUI (Fig. 7). In surveillance and telepresence views, the application level events are used to satisfy live queries, whereas in the lifeblogging view, we stored the application level events in the form of a blog. This aspect establishes our claim of supporting diverse media sensing applications with same set of sensors and the loosely-coupled ObSys sensing architecture.The aplication level event statistics for all three applications are shown in Fig.6.

While not all aspects of a generic observation system are functional yet, we have made significant progress in the aspects of 'Data organization and query processing' and 'Supporting multiple views and applications' which have allowed users to have multiple (Surveillance, Telepresence, Lifeblogging, and Ethnography) views across two different environments. In the longer term, we would like to create a fully working observation system which can be deployed in different environments, ranging from small rooms to large-scale outdoor areas with little to no set-up costs.

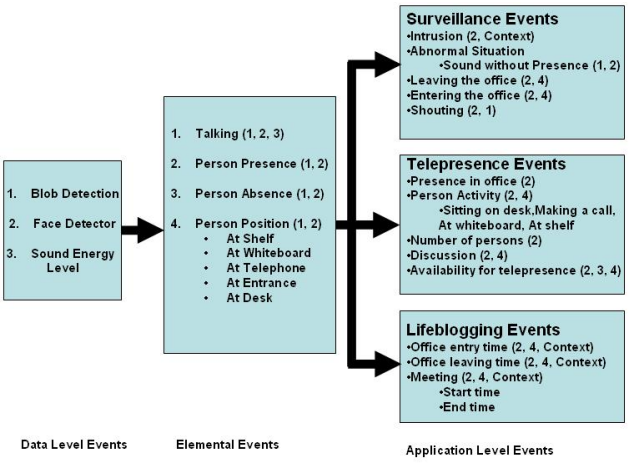

Figure 5: Event classification and their interrelation for indoor environment experiments. The numbers in parentheses denote the immediate lower level event used to detect the event.

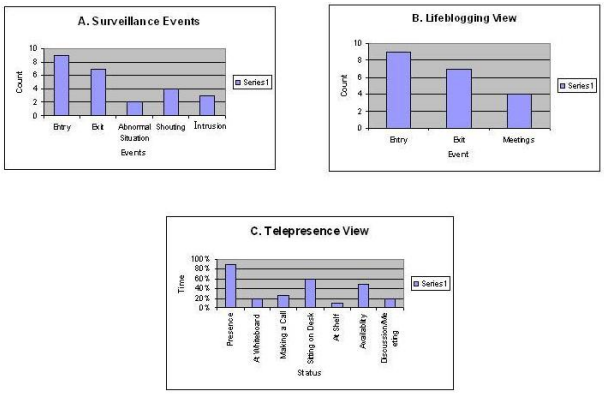

Figure 6: Event statistics for indoor implementation

5. OUTLOOK AND CONCLUSIONS

To conclude, in this paper we have described our notion of 'Observation System', which is a generic superset of media applications like surveillance, video ethnography, tele-presence, assisted living etc. We have described how an abstracted sensing architecture can allow multiple application views to be generated from the same set of multi-modal data. In order to create such generic scalable systems, we need to employ a loosely-coupled sensing architecture and an event based query processing architecture. We have also described two prototype implementations and the obtained results are encouraging.

We intend to extend our work into larger number of sensors with multiple modalities and larger physical distribution in near future. We are also looking at implementing the completely working observation system wherein all described components e.g. setting up the environment are fully automated and efficient.

6. REFERENCES

[1] P. Bonnet, J. Gehrke, and P. Seshadri. Towards sensor database systems. *Lecture Notes in Computer Science*, pages 3–14, 2001.

[2] O. Gnawali, R. Govindan, and J. S. Heidemann:. Implementing a sensor database system using a generic data dissemination mechanism. *IEEE Data Enginering Bulletin*, 28(1):70–75, 2005.

[3] F. Golshani and N. Dimitrova. A language for content-based video retrieval. *Multimedia Tools and Applications*, 6(3):289–312, 1998.

[4] A. Hampapur, S. Borger, L. Brown, C. Carlson, J. Connell, M. Lu, A. Senior, V. Reddy, C. Shu, and Y. Tian. S3: The ibm smart surveillance system: From transactional systems to observational systems. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, 2007.

[5] R. Jain. Out-of-the-box data engineering - events in heterogeneous data environments. In *International Conference on Data Engineering*, pages 8–21, 2003.

[6] R. Jain, P. Kim, and Z. Li. Experiential meeting systems. In *ACM Workshop on Experiential TelePresence*, pages 1–12, 2003.

[7] S. Moncrieff, S. Venkatesh, G. West, and S. Greenhill. Multi-modal emotive computing in a smart house environment. *Pervasive and Mobile Computing*, 3(2):74–94, 2003.

[8] E. Ochs, A. Graesch, A. Mittmann, T. Bradbury, and R. Repetti. Video ethnography and ethnoarchaeological tracking. *The Work and Family Handbook: Multi-Disciplinary Perspective, Methods, and Approaches*, pages 387–409, 2006.

[9] C. Parker and S. Pfeiffer. Video blogging: content to the max. *IEEE Multimedia*, 12(2):4–8, 2005.

[10] E. Saykol, U. Gl̈udl̈ukbay, and O. Ulusoy. A database model for querying visual surveillance by integrating semantic and low-level features. In *International Workshop on Multimedia Information Systems*, pages 163–176, 2005.

[11] M. Valera and S. Velastin. Intelligent distributed surveillance systems: a review. In *IEE Proceedings on Vision, Image and Signal Processing*, pages 192–204, 2005.

[12] U. Westermann and R. Jain. E - a generic event model for event-centric multimedia data management in echronicle applications. In *International Conference on Data Engineering Workshops*, 2006.

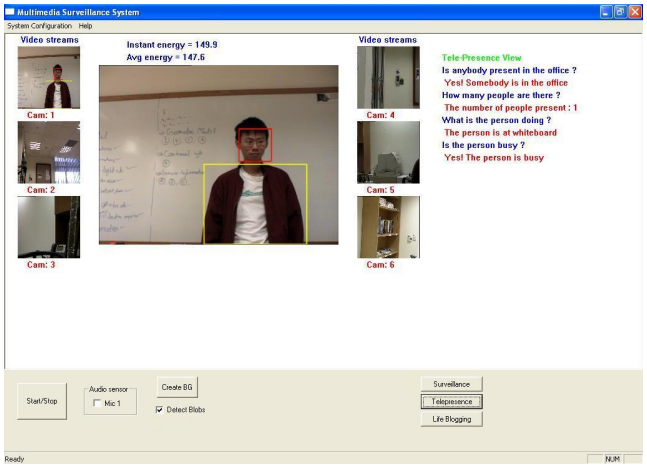

Figure 7: Telepresence View