Exploiting Semantics for Event Detection Systems

Ronen Vaisenberg[1]
School of Information and Computer Sciences University of California, Irvine
Email: ronen@uci.edu

I. Introduction

This dissertation is building towards a flexible and scalable middleware for sentient spaces wherein sensors are used to observe the state of a physical environment in real time to create awareness which, in turn, is used to build applications that bring new functionalities and/or new efficiencies to the environment. Examples of such sentient spaces could be as varied as smart buildings that use video sensors for surveillance or for building pervasive applications such as person locator to dynamically instrumented crisis sites wherein sensors carried by first responders and/or brought to crisis sites are used to monitor the crisis site and the progress of response activities. The key goal of this work is towards developing a robust and scalable middleware technology that can handle challenges related to heterogeneity of sensors, limited resources (e.g., network bandwidth), and the need for rapid deployment (e.g., self-calibration of sensing), and programmability (e.g., hiding complexity of sensor and sensor programming from the application writers).

These research objectives are deeply related to two areas that have had significant research interests in the past: stream processing systems, and sensor networks. However, none of the existing research in these areas meets the challenges posed by sentient spaces.

II. Addressed Challenges

The background for this dissertation are two problems published in MMCN 2008[1] and 2009[2].

The first problem studied was that of **Recalibration in State Monitoring Sensors**. Sensors are deployed usually in an unsupervised environment where physical perturbations might lead to incorrect output generated by the sensor. To support the automatic recovery of sensors from such situations, a general purpose framework was developed that exploits the sensor's observed system semantics and performs an automated recalibration of the detection parameters. The main idea behind the approach is to model the monitored system as a finite state machine and learn the semantics of transitions between states. Once the sensor deviates from the learnt model, the algorithm attempts to find a new set of parameters that maximize the consistency with the learnt model.

We observe that the task of low level event detection is to detect the state of an observed system, based on sensor readings, *reliably*. Thus sensor readings are translated to a finite set of possible system states, which represent the observed system's state. For example, the state of the coffee machine in our office kitchen can be represented by four states of interest: "Empty", "Half-Full", "Full" and "Coffee-Pot Off", illustrated in Fig. 1. The semantic characteristics of the monitored system (which in this case represent **the temporal characteristics of coffee drinking**) are captured by a temporal state transition model: $p_{semantic}(S_i|S_1,..,S_{i-1})$ which is the probability of the system being at state S_i at time t_i, given it was at states $S_1,..,S_{i-1}$ at times $t_1,..,t_{i-1}$.

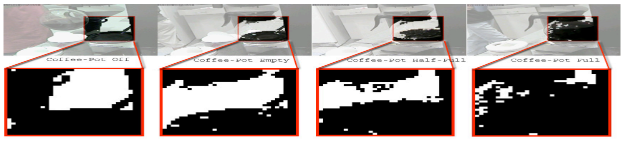

Fig. 1: The 4 different states in our example. The black/white cells represent dark/bright average pixel color.

The Recalibration Process is initiated when the set of detected states deviates significantly from the system's semantic model, assumed to occur due to physical perturbations, e.g., change in the field of view.
The input for the recalibration process are the system semantics and a stream of past sensor observations:

$$O = \{o_1 =< t_1, f_1 >, o_2 =< t_2, f_2 >, .., o_n =< t_n, t_n >\}$$

where f_i is a vector of extracted features, (e.g., color or texture features for a video sensor) and t_i is its corresponding timestamp. As a first step, the feature level processing takes place: the observations are clustered into k clusters (k is the number of monitored system's states), based on feature similarity. Assuming that changes in the feature values represent changes in the system's state, these clusters represent the different states of the monitored system. At the second step the algorithm takes into account the system semantics and determines which system state is represented by which cluster. Each possible assignment of states to clusters ($k!$ possible assignments) translates to a temporal state transition model, which is evaluated against the system semantics. The algorithm efficiently finds the assignment of states to clusters which maximizes the consistency with the semantic model. The new states assigned to the sensor observations O are used to tune the parameters of the detection algorithm after the physical perturbations. In our case, the detection algorithm labels new observations by finding the nearest neighbor among k different representative features, each represents one system state (see Fig. 1).

The k centroids of the labeled clusters, generated by the recalibration algorithm, represent the new set of parameters

[1] Advised by Prof. Sharad Mehrotra

tuned by the recalibration process.

The second problem studied is that of **Scheduling Under Resource Constraints**. In particular, consider a real-time tracking system which is responsible of monitoring human activity as observed by a large number of camera sensors. When considering systems of relatively large scale, constraints arise from network bandwidth restrictions, I/O and disk usage from writing images, and CPU usage needed to extract features from the images. Assume that, due to resource constraints, only a subset of sensors can be probed at any given time unit. The "best" subset of sensors to probe under a user-specified objective (e.g., detecting as much motion as possible, maximizing probability of detecting "suspecious" events). With this objective, we would like to probe a camera when we expect motion, but would not like to waste resources on a non-active camera.

The main idea behind our approach is the use of sensor semantics to guide the scheduling of resources. We learn a dynamic probabilistic model of motion correlations between cameras, and use the model to guide resource allocation for our sensor network.

Formally, we define a plan for N cameras to be a binary vector of length N that specifies which cameras will be probed in the next time instant. $Plan = \{C_i | 1 \leq i \leq N\}$, where $C_i \in \{0, 1\}$. The cameras were selected to optimize an application-dependant **benefit function (BF)**. For example, a particular application may want all image frames for which there is motion (all motion events are equally important), while another application may define that two images of two different individuals are more important than two of the same person. Another consideration is the **cost** of a plan, in terms of network resources, referred to as **cost function (CF)**. Different plans may not cost the same in terms of network resources since it may be less expensive to probe the same sensor at the next time instant. In a fully general model, one might also place the number of sensor probes K into the cost function.

Learning the Habits of People's Whereabouts: Monitoring a real-time activity can benefit from accurate predictions, given that these predictions arrive early enough, for the real-time process to take action and the process is fast enough to act on it.

We collected motion from a dozen camera sensors spread across two floors in our CS building. The semantics of interest included: 1. **A-priori** knowledge of where motion is likely to be. In the case of the building, it is likely that the camera at the front door will see more motion than other cameras. 2. **Self correlation** of camera stream over time. Given that a camera observes an event and given the camera's field of view (FOV), one could predict the probability that the event will continue. For instance, a camera focussing on a long corridor will have a person in view for a longer period of time compared to a camera that is focused on an exit door.
3. **Cross-Correlations** between cameras. Clearly a person who exits a FOV of one camera will be captured by another depending upon the trajectory of the individual and the placement of the cameras.

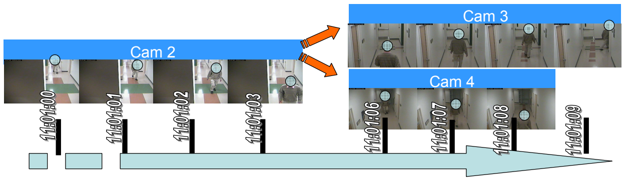

Fig. 2: Illustration of a Conditional Correlation of Motion Between Cameras.

The Real-Time Scheduling of Data Collection: takes place after every probe of the sensors by the system. The monitoring system is overwhelmed by the number of sensors, and can only monitor a small subset of all sensors at any given instant of time. Based on the partial information of people's whereabouts collected by the system's probes and the learnt semantics, the algorithm efficiently computes the best set of sensors to probe in the next time instant, to maximize the monitoring system's informative probes (for example, probes that contain motion).

III. Future Work

While the dissertation work has to date focused on exploiting semantics for calibration and scheduling in multimedia streams, it is headed towards both semantic event stream management and a high level query language support to build applications on top of the multimedia sensor infrastructure. In this context, we are working on a prototype system, SATViewer which visualizes streams collected from multiple sensors.

The key challenge in supporting a higher level query language over sensor captured streams is that of *uncertainty* in the semantic meaning of the captured information. For example, consider a query for streams containing entity e_i. Arguably, streams with a clear front picture will be returned with high recall and precision. However, there might be many other important streams which contain entity e_i which will probably not be returned as an answer, simply because the underlying processing fails to recognize e_i in it. We plan to address the uncertainty problem by learning entity semantics to reduce the uncertainty about an entity when only partial identifying information is available incorporating techniques for semantic entity resolution.

The key challenge in designing a visualization tool for a pervasive system is that of *information overload* - limitations in user perception and in available display sizes prevent easy assimilation of information from massive databases of stored sensor data. For instance, in our setting, we have over 200 camera sensors deployed at two buildings; even a very simple query interested in monitoring these buildings will have to visualize 400 streams (audio/video) for any given time. In SATViewer,we plan to address the information overload problem using two key strategies – (a) ranking and prioritization of relevant sensor streams and (b) summarization of selected sensor streams. Both will exploit semantics of different modalities.

References

[1] R. Vaisenberg, S. Ji, B. Hore, S. Mehrotra, and N. Venkatasubramanian. Exploiting Semantics for Sensor Re-Calibration in Event Detection Systems. In *Proceedings of SPIE*, volume 6818, page 68180P. SPIE, 2008.

[2] R. Vaisenberg, M. Sharad, and R. Deva. Exploiting Semantics For Scheduling Data Collection From Sensors On Real-Time To Maximize Event Detection. In *Proceedings of SPIE*. SPIE, 2009.