MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Toward Spatial Queries for Spatial Surveillance Tasks

Yuri A. Ivanov, Christopher R. Wren

TR2006-051 May 2006

Abstract

Surveillance systems are largely focused on the movement, storage, and review of video and audio streams. The recent shift from human monitoring toward automated interpretation presages a fundamental change in our relationship with surveillance systems. Despite this shift, the state of the art has so far remained trapped by the notion of a sensor stream. That is, the systems being sold today still largely constrain their analysis tools to operate on a single input stream. Some research systems have tried to present video streams in context: superimposed on a floor plan. Some allow searches for salient people or objects across video streams. We present here a technique for generating queries that are embedded in context. We allow the operator to specify queries that take advantage of the spatial context, by utilizing spatial gestures to assemble the query terms on a map of the site. We show an early prototype system operating on data from a reasearch facility observed by a heterogeneous network of sensors.

Pervasive Workshop on Pervasive Technology Applied Real-World Experiences with RFID and Sensor Networks (PTA)

February 10, 2006: Submitted to the Pervasive 2006 Workshop on Pervasive Technology Applied Real-World Experiences with RFID and Sensor Networks.
May 7, 2006: To appear in the Pervasive 2006 Workshop on Pervasive Technology Applied Real-World Experiences with RFID and Sensor Networks.

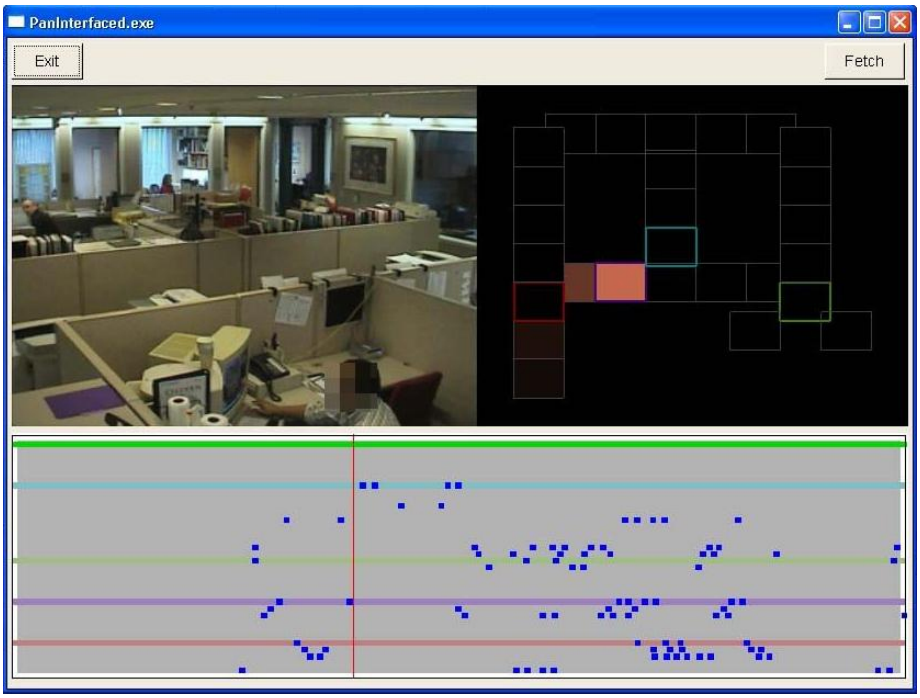

Figure 1: The system interface. Video stream in the upper left. Site map with sensor activation areas in the upper right. The event timeline is on the bottom. Please see the text for details.

1 Introduction

In this paper we will briefly introduce a system for generating queries on multimedia surveillance databases. Queries on these databases differ from queries on typical multimedia databases in that the sensor streams share a spatial and temporal context with one another. We leverage this shared context explicitly in the visualization of the query results as well as in the interface used to author the queries. The queries are generated from spatial gestures that the operator performs on an active map. The map is active in that it is used both for playback and query. This approach treats the surveillance system not as a mundane collection of sensors, but as an integrated sensor network: a unified sensing system. We present early results using data from a heterogeneous sensor network research platform consisting of a camera and a network of motion detectors to provide context for the video stream. We expect the results to be applicable to networks over a wide range of sensor composition: from all-camera, to cameras embedded in a bevy of simple sensors.

Figure 2: Left: A pan-tilt-zoom camera system. Right: A wireless, passive, infrared motion detector.

2 Display

Our prototype interface is shown in Figure 1. The upper left pane is a video playback window that could present video streams from any number of cameras. In our research platform there is currently only one camera, but there could be many cameras. The camera is mounted on a pan-tilt platform as shown on the left of Figure 2.

The upper right pane is a site map. The map shows activation zones for the collection of sensors in the database. If a sensor is active, then the zone is rendered in a bright color. This color fades to black several seconds after the sensor activation. The fading helps to provide temporal context to the operator. In our research platform these events are generated by network of 27 passive-infrared (PIR) motion detectors, however they could be generated by any modality: pressure, break-beam, or even virtual detectors built on top of video or audio streams.

The lower panel is the event timeline. The timeline shows the same events as the site map, but in a "piano roll" format. The current time is marked by a vertical line. The event sources are arranged along the vertical axis. The small dark rectangles represent an event (vertical position) being active for a time (horizontal position and extent).

The displays are joined by a common highlighting scheme. Activation zones may be highlighted on the map. These selected sensors are then highlighted on the event timeline by horizontal bars. The highlight colors on the map and the timeline are consistent.

Figure 3 shows the interface with the event timeline zoomed completely out to show the entire data set. The domain includes 150 thousand event records recorded over 12 days. The day and week patterns are clearly visible in the event timeline display as dense vertical bands of activity. Queries on this database return results in 2-4 seconds.

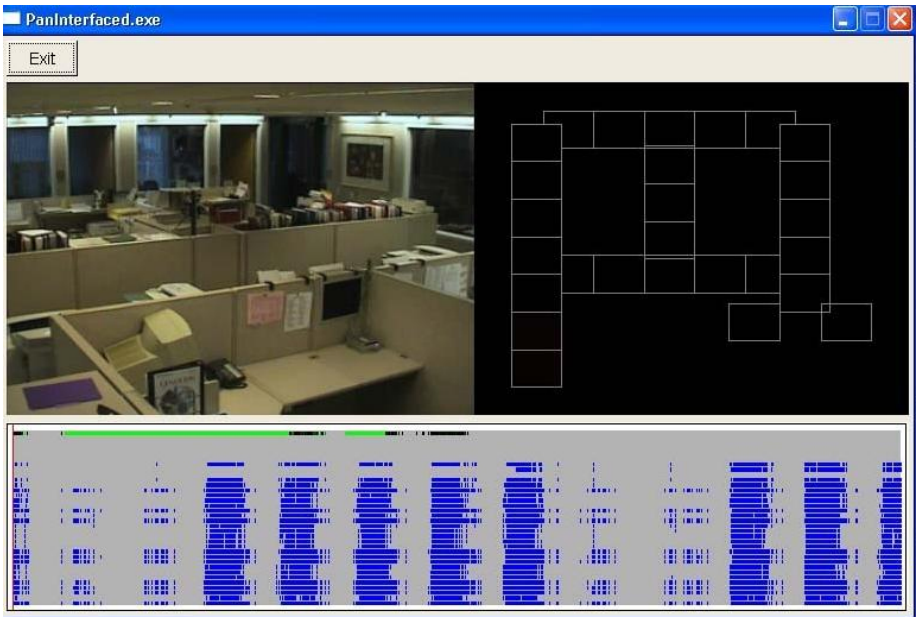

Figure 3: The interface showing the full 12-day data set.

3 Selection and Queries

An initial query might simply request all the segments with motion in the video stream. Perhaps the operator is interested in seeing people who are using the copy machine (at the center of the map). This query will return too much information. Motion in the video may not be related to the copy machine since the camera also views a desk, cubicles, a few walkways and several offices. One could mark out the region of interest on the video frame. Due to perspective effects, objects of interest in the foreground may occupy the same pixels as distraction objects in the background, therefore a simple 2D motion interest region in the video frame will not be sufficient to distinguish objects of interest from distractors.

A better idea is to specify the terms of the query on the map. The system can then draw on other sensors that might view that area directly. It could even utilize perceptual mechanisms make a judgment about the depth of objects in the video frame and generate appropriate events for the database. In any case, the user indicates the region of interest on the map with a familiar pointing gesture. A query is automatically generated that joins the global video motion with the context event and the results are appropriately refined.

There still may be too much data returned. If the operator happens to know that they are interested in a person who was likely to approach from a certain direction, then the above scenario is easily extended to a sweeping gesture, or a series of pointing gestures that specify a path. The system automatically joins

ID	sensor-id	start-time	end-time	avg-magnitude	time-processed

Figure 4: Structure of the table storing the data for the motion sensors in our system.

the context constraints with the video and the results are again appropriately refined.

As an example, consider the `motionsensor` table in the database that stores information about the sensor activations. We use the following table structure, including record id, sensor id, activation interval and average activation strength:
Imagine that the user selects sensors with IDs 265, 293 and 283 in the given order. Currently we assume that at the normal walking speed a person would activate these sensors such that one fires within 1-3 seconds after the other. This, the following temporal query is generated[1]:

```
 SELECT t0.start_time, t2.end_time
   FROM motionsensors t0, motionsensors t1, motionsensors t2
   WHERE t0.sensor_id = 265 AND t1.sensor_id = 293
      AND t2.sensor_id = 283
      AND (t1.start_time - t0.start_time) > 1000
      AND (t1.start_time - t0.start_time) < 3000
      AND (t2.start_time - t1.start_time) > 1000
      AND (t2.start_time - t1.start_time) < 3000
ORDER BY t0.start_time
```

Since the system has access to the database of events, it can automatically mine the data for statistics, such as inter-arrival times. A table of inter-arrival times could allow the system to build better queries by automatically picking only the most likely temporal offsets for a specified sequence of events. In a future version of this system we imagine that the query space will be rendered in a way very similar to the event timeline. This query display would allow the user to graphically tune the timing of the components to specify overlap, joins, or different lags.

4 Reviewing Results

Figure 5 shows the result of a query very similar to that described above. The operator selected a sequence of four context events: a path leading from the lower-right corner to the to the top end of the central hallway. The selected events are highlighted in order: red, blue, cyan, green.

On the event timeline the horizontal highlight bars indicate the sensors that were involved in the query. The temporal extent of each query result is shown

[1]This query is simplified for the sake of clarity. For a large database such query can get computationally intensive and may to be optimized.

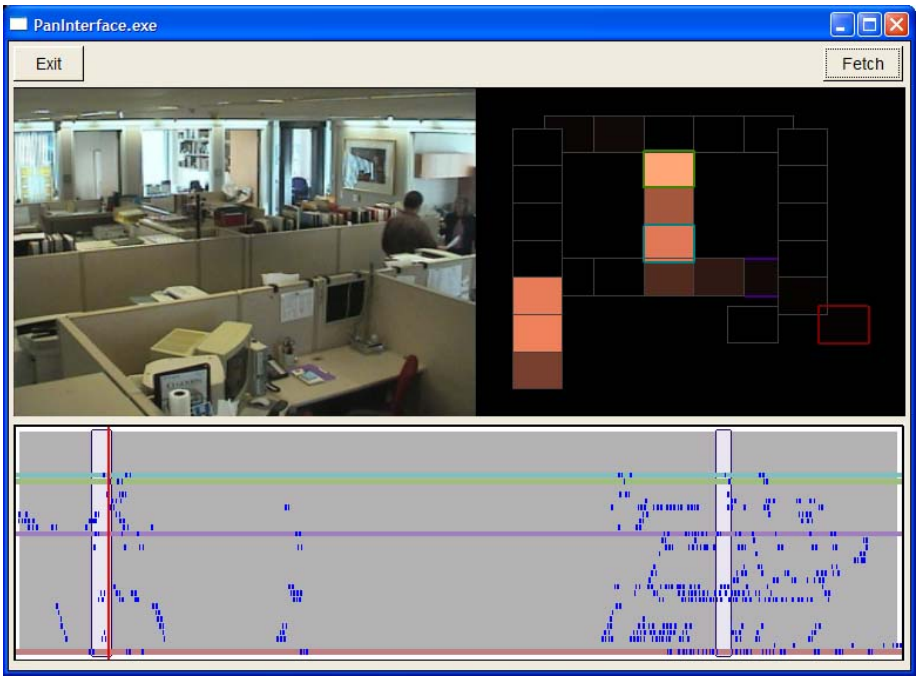

Figure 5: Query results. highlighted in the event timeline with the playback above showing the very end of the first result.

as a white rectangle on the gray background of the event timeline. It is possible to see that each result contains dark boxes indicating event detection ordered red (bottom), blue (middle), cyan (top), and green (second from top).

That vertical bar in Figure 5 indicates that the current playback time is at the end of the first query results. We can see that there is a target at the right end of the middle hallway in the video (right in the video corresponds to up on the map). On the map we also see a bright activation on the final sensor zone, and a fading trail leading back down along the selected path and then right. Note that there is a second activity trace on the lwer-left corner that is concurrent with the target trace, but is otherwise unrelated to the current query. We can tell at a glance that: this result satisfies our query, we have a visual of the target, and we know that he approached the hallway from the lower-right.

Figure 6 illustrates a query crafted specifically to recover a person performing the (admittedly odd) behavior of walking loops around one of the circular hallways. A loop of activations is marked on the map and is clearly reflected in the structure of the query results in the event timeline.

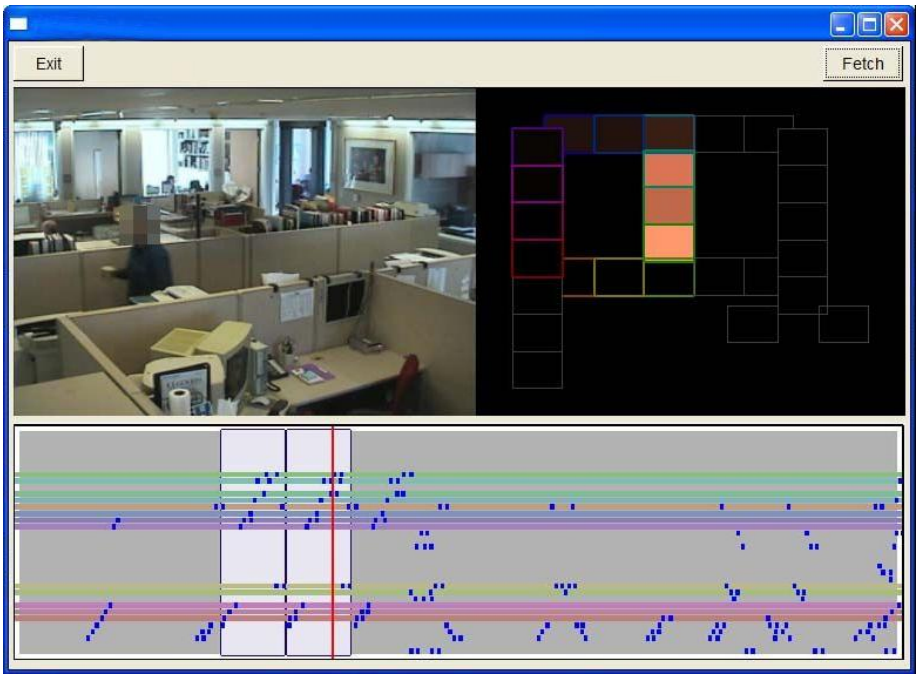

Figure 6: Result for the "loop" query. (Face obscured to protect the innocent.)

5 Conclusion

We have presented a gestural interface to surveillance video queries. The system automatically generates database queries in response to intuitive, contextually-embedded gestures on a 150K event database within seconds.

References

[1] 3vr security, inc. - intelligent video management systems for the mainstream security industry. www.3vr.com.

[2] R. A. Bolt. 'put-that-there': Voice and gesture at the graphics interface. *Computer Graphics Proceedings, SIGGRAPH 1980,*, 14(3):262–70, July 1980.

[3] Christoph Maggioni. Gesturecomputer—new ways of operating a computer. SIEMENS AG Central Research and Development, 1994.

[4] David McNeill. *Hand and Mind: What Gestures Reveal about Thought.* The University of Chicago Press, 1992.

[5] H. S. Sawhney, A. Arpa, R. Kumar, S. Samarasekera, M. Aggarwal, S. Hsu, D. Nister, and K. Hanna. Video flashlights: real time rendering of multiple videos for immersive model visualization. In *Proceedings of the 13th Eurographics workshop on Rendering*, pages 157–168, 2002.

[6] Vistascape security systems - automated wide-area surveillance. www.vistascape.com.