IBM Smart Surveillance System (S3): Event Based Video Surveillance System with an Open and Extensible Framework

Ying-li Tian, Lisa Brown, Arun Hampapur, Max Lu,
Andrew Senior, and Chiao-fe Shu

IBM T. J. Watson Research Center,
PO Box 704, Yorktown Heights, NY 10598

Abstract

The increasing need for sophisticated surveillance systems and the move to a digital infrastructure has transformed surveillance into a large scale data analysis and management challenge. Smart surveillance systems use automatic image understanding techniques to extract information from the surveillance data. While the majority of the research and commercial systems have focused on the information extraction aspect of the challenge, very few systems have explored the use of extracted information in the search, retrieval, data management and investigation context. The IBM smart surveillance system (S3) is one of the few advanced surveillance systems which provides not only the capability to automatically monitor a scene but also the capability to manage the surveillance data, perform event based retrieval, receive real time event alerts thru standard web infrastructure and extract long term statistical patterns of activity. The IBM S3 is easily customized to fit the requirements of different applications by using an open-standards based architecture for surveillance.

1. Introduction

Smart Video Surveillance is the use of computer vision and pattern recognition technologies to analyze information from situated sensors [1], [2], [3], [4]. The analysis of the sensor data generates events of interest in the environment. For example, an event of interest at a departure drop-off area in an airport is "cars that stop in the loading zone for extended periods of time." As smart surveillance technologies have

matured, they have typically been deployed as isolated applications which provide a particular set of functionalities. However, isolated applications while delivering some degree of value to the users, do not comprehensively address security requirements. In this paper we describe the IBM S3 including the system architecture and its basic functions, technologies, data management, and the flexible framework for different applications.

Video analysis and video surveillance are active areas of research. The key technologies are video-based detection and tracking, video-based person identification, and large-scale surveillance systems. A significant percentage of basic technologies for video-based detection and tracking were developed under a U.S. government-funded program called Video Surveillance and Monitoring (VSAM) [1]. This program looked at several fundamental issues in detection, tracking, auto-calibration, and multi-camera systems [6], [7], [8]. There has also been research on real-world surveillance systems in several leading universities and research labs [9]. The next generation of research in surveillance is addressing not only issues in detection and tracking but also issues of event detection and automatic system calibration [10]. The second key challenge of surveillance—namely, video-based person identification—has also been a subject of intense research. Face recognition has been a leading modality with both ongoing research and industrial systems [11], [12]. A recent U.S. government research program called Human ID at a Distance addressed the challenge of identifying humans at a distance using techniques like face at a distance and gait-based recognition [13]. One of the most advanced systems research efforts in large-scale surveillance systems is the ongoing U.S. government program titled Combat Zones That See [14]. This program explores rapidly deployable smart camera tracking systems that communicate over ad hoc wireless networks, transmitting track information to a central station for the purposes of activity monitoring and long-term movement pattern analysis. There are several technical challenges that need to be addressed to enable the widespread deployment of smart surveillance systems. Of these, we highlight four key challenges.

The multi-scale challenge: One of the key requirements for effective situation awareness is the acquisition of information at multiple scales. A security analyst who is monitoring a parking lot observes not only

where cars are in the space and what they are doing but also pays attention to the license plates for identification. The analyst uses these visual observations in conjunction with other knowledge to make an assessment of threats. While existing research has addressed several issues in the analysis of surveillance video, very little work has been done in the area of better information acquisition based on real-time automatic video analysis, like automatic acquisition of high-resolution face images. Given the ability to acquire information at multiple scales, the challenges of relating this information across scales and interpreting this information become significant. Multi-scale techniques open up a whole new area of research, including camera control, processing video from moving cameras, resource allocation, and task-based camera management in addition to challenges in performance modeling and evaluation.

The contextual event detection challenge: While detecting and tracking objects is a critical capability for smart surveillance, the most critical challenge in video-based surveillance (from the perspective of a human intelligence analyst) is interpreting the automatic the analysis output to detect events of interest and identify trends. Current systems have just begun to look into automatic event detection. The area of context-based interpretation of the events in a monitored space is yet to be explored. Challenges here include: using knowledge of time and deployment conditions to improve video analysis, using geometric models of the environment and other object and activity models to interpret events, and using learning techniques to improve system performance and detect unusual events.

The large system deployment challenge: The basic techniques for interpreting video and extracting information from it have received a significant amount of attention. The next set of challenges deals with how to use these techniques to build large-scale deployable systems. Several challenges of deployment include minimizing the cost of wiring, meeting the need for low-power hardware for battery-operated camera installations, meeting the need for automatic calibration of cameras and automatic fault detection, designing the scalable, extensible open architecture, the database structure for the meta-index, and developing system management tools.

The management of large amount of data challenge: The video surveillance systems which run 24/7 (24 hours a day and seven days a week) create a large amount of data including videos, extracted features, alerts, and etc. How to manage this data and make it easily accessible for query and search are other challenges.

The IBM Smart Surveillance System (S3) is a middleware offering for use in surveillance systems and provides video based behavioral analysis capabilities. It offers not only the capability to automatically monitor a scene but also the capability to manage the surveillance data, perform event based retrieval, receive real time event alerts thru standard web infrastructure and extract long term statistical patterns of activity. An open and extensible framework is designed so that IBM S3 can easily integrate multiple independently developed event analysis technologies in a common framework. This paper is organized as following. Section 2 introduces the architecture of IBM S3. Section 3 describes details of the technologies which are used in the IBM S3 engines. Section 4 introduces the detailed model for IBM S3 data management. The S3 pilot at the IBM T.J. Watson Research Center and its application at a retail store for retail loss prevention are described in Section 5. Section 6 summarizes the advantages of the IBM S3 and discusses future directions.

2. The IBM S3 System

The current version of the IBM S3 includes two components: (1) ***Smart Surveillance Engine (SSE)*** which provides the front end video analysis capabilities; (2) ***Middleware for Large Scale Surveillance (MILS)*** which provides data management and retrieval capabilities. These two components in conjunction with the IBM DB2 and IBM WebSphere Application Server support the following features:

- Local Real-Time Surveillance Event Notification: This set of functions provides real-time alerts to the local application which is running the SSE.

- Web Based Real-Time Surveillance Event Notification: This set of functions provide a web based real-time event notification within 3 seconds of the occurrence of a specified event in the monitored area; for example "Speeding Vehicle."
- Web Based Surveillance Event Retrieval: This set of functions provides the ability to retrieve surveillance events based on various attributes like object type/speed/color.
- Web Based Surveillance Event Statistics: This set of functions provides the ability to compute a variety of statistics on the event data. For example, distribution of people arriving and leaving a building over a day.

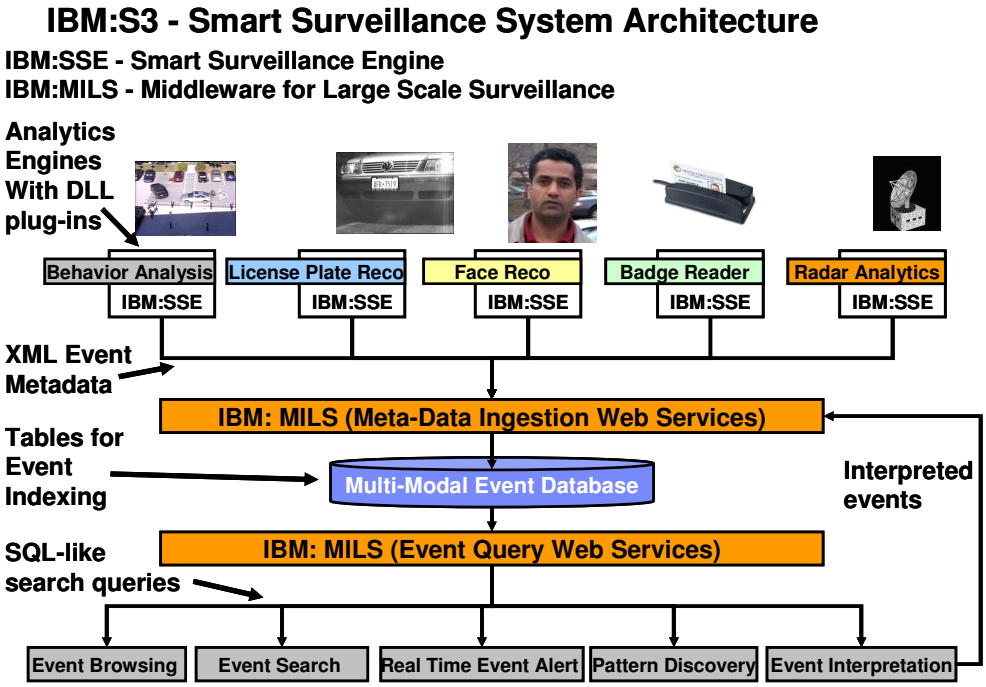


Figure 1: An open and extensible architecture for IBM S3. The smart surveillance engine (SSE) provides a plug and play framework for video analytics. The event meta-data generated by the engines are sent to the database as XML files. Web services API's allow for easy integration and extensibility of the meta-data. Various applications like event browsing and real time alerts can use an SQL-like query language through web services interfaces to access the event meta-data from the data base.

The IBM S3 system architecture presented below is designed to satisfy two key principles.

- Openness: This requires that the system allows integration of both analysis and retrieval software made by third parties and that the system be designed using approved standards and commercial off-the-shelf (COTS) components.
- Extensibility: This requires that the system should have internal structures and interfaces that will allow for the functionality of the system to be extended over a period of time.

Figure 1 shows the S3 system architecture. The architecture enables the use of multiple independently developed event analysis technologies in a common framework. The events from all these technologies are cross indexed into a common repository allowing for correlation across multiple sensors and event types. The example system shown in Figure 1 has the following technologies integrated into a single system:

License Plate Recognition: This technology could be deployed at the entrance to a facility where it catalogs the license plate of each of the arriving and departing vehicles.

Behavior Analysis: This technology detects and tracks moving objects and classifies them into a number of predefined categories. This could be deployed on various cameras overlooking the parking lot and the perimeter and inside of a facility.

Face Detection/ Recognition: This technology can be deployed at entry ways to capture and recognize faces.

Badge Reading: Events from access control technologies can also be integrated into the S3 system.

The events from all the above surveillance technologies are cross indexed into a single repository. In such a repository a simple time range query across the modalities will extract license plate information, vehicle appearance information, badge information and face appearance information, thus allowing an analyst to easily correlate these attributes.

The high level description of data flow in the S3 architecture is summarized as following:

- Sensor data from a variety of sensors is processed in the Smart Surveillance Engines (SSEs). Each SSE can generate real-time alerts and generic event meta-data.

- The meta-data generated by the SSE is represented using XML. The XML documents have some set of fields which are required and common to all engines and others which are specific to the particular type of analysis being performed by the engine.
- The meta-data generated by the SSE's is transferred to the backend MILS system. This is accomplished via the use of web services data ingest API's provided by MILS.
- The XML meta-data is received by MILS and indexed into predefined tables in the IBM DB2 database. This is accomplished using the DB2 XML extender. This allows for fast searching using the primary keys.
- MILS provides a number of query and retrieval services based on the types of meta-data available in the database.

3. The IBM Smart Surveillance Engine (SSE)

3.1 Basic Functionalities of IBM SSE

The IBM Smart Surveillance Engine (SSE) is a C++ based framework for performing real-time event analysis. This engine is capable of supporting a variety of video/image analysis technologies and other types of sensor analysis technologies. It provides the following functionalities.

- Real Time Video Based Alerts: These are alerts which depend solely on the movement properties of objects within the monitored space. Examples include 1) Motion Detection: 2) Directional Motion Detection 3) Abandoned Object Alert 4) Object Removal and 5) Intentional camera movement or blinding:
- Viewable Video Index (VVI): The SSE detects and tracks all moving objects within the cameras field of view. The SSE creates the VVI as a set of XML documents. The VVI encodes all the interesting activities in the video including, 1) Number of objects in a scene 2) Object Class, the current engine classifies objects in Single Person, Group of People and Vehicles, 3) Object appearance properties, including color, texture, shape, size and their changes over time, 4) Object

movement properties, including position, velocity and trajectory, 5) Occlusion parameters when objects are in an occlusion, 6) Background changes due to changes in lighting and stopping of moving objects, 7) Event information: any events that may be flagged by the engine. The VVI index is also ideal for monitoring activities over a mobile wireless device such as a PDA, because of its extreme low bandwidth requirements (~10MB/hour). The index encodes all the activity that occurs in the video in terms of evolving background models, evolving object models and motion trajectories of each of the moving objects. The information contained in the VVI can be used by an application to render the activity in the video independent of the original video stream.

The SSE also provides the following support functionalities for the core analysis components

- Standard Plug-in Interfaces: Any event analysis component which complies with the interfaces defined by the SSE can be plugged into the SSE. The definitions include standard ways of passing data into the analysis components and standard ways of getting the results from the analysis components.
- Extensible Meta-Data Interfaces: The SSE provides meta-data extensibility. For example, consider a behavior analysis application which uses detection and tracking technology. Let us assume that the default meta-data generated by this component is object trajectory and size. If the designer now wishes to add, color of the object into the metadata, the SSE enables this by providing a way to extend the creation of the appropriate XML structures for transmission to the backend (MILS) system.
- Real-time Alert Interfaces: The real-time alerts are highly application dependent, while a person loitering may require an alert in one application, the absence of a guard at a specified location may require an alert in a different application. The SSE provides an easy mechanism for developers to plug-in application specific alerts. It provides standard ways of accessing event-meta data in memory and standardized ways of generating and transmitting alerts to the backend (MILS) system.
- Compound Alert Interfaces: In many applications, the users will require the use of multiple basic real-time alerts in a spatio-temporal sequence to compose an event that is relevant in his/her application context. The SSE provides a simple mechanism for composing compound alerts.
- Real-time Actuation Interfaces: In many applications the real-time event meta-data and alerts are used to actuate alarms, visualize positions of objects on an integrated display and control PTZ cameras to get better surveillance data. The SSE provides developers with an easy way to plug-in actuation modules which can be driven from both the basic event meta-data and by user defined alerts.

- Database Communication Interfaces: The SSE also hides the complexity of transmitting information from the analysis engines to the database by providing simple calls to initiate the transfer of information.

3.2 Basic Technologies of IBM SSE

The IBM SSE is based on the following key video analysis technologies [15, 16, 17, 18, 19, 20], some details can be found in paper [3].

- Object Detection: This set of technologies can detect moving objects in a video sequence generated by a static camera. The detection techniques are invariant to changes in natural lighting, reasonable changes in the weather, distracting movements (like trees waving in the wind), and camera shake. Several algorithms are available in IBM S3 including adaptive background subtraction with healing [3, 15] which assumes a stationary background and treats all changes in the scene as objects of interest and salient motion detection [16] which assumes that a scene will have many different types of motion, of which some types are of interest from a surveillance perspective.
- Object Tracking: This set of technologies can track the shape and position of multiple objects as they move around a space that is monitored by a static camera. The techniques are designed to handle significant occlusions as objects interact with one another. Currently, several trackers are implemented in IBM S3 including appearance-based tracker, blob-based tracker, and face tracker. More details about the tracking methods can be found at the publication list at [3, 5, 19, 20].
- Object Classification: These technologies use various properties of an object including shape, size and movement to assign a class label to the objects. Our system classifies objects into vehicles, individuals, and groups of people based on shape features such as compactness, bounding ellipse parameters, and motion features (such as recurrent motion measurements, speed, and direction of motion). From a small set of training examples, we are able to classify

objects in similar scenes using a Fisher linear discriminant to perform feature reduction, followed by a nearest neighbor classifier and temporal consistency information.

- Face Detection: IBM S3 includes frontal and multi-view face detectors based on Haar and optimized wavelet features by using a cascade of boosted classifiers [21].

3.3 High Level Technologies of IBM SSE

3.3.1 High Level Track Post-processing

In order to improve system performance, our system has been augmented with a post track processing module. This is part of the modular design of our system in which background subtraction, tracking and post tracking are each self-contained and can be replaced in a plug-and-play fashion. The post track processing can be used to improve system performance by reducing the number of false positives and temporal and spatial fragmentation errors. This procedure is parameterized so it can adapt to different environments, cameras, or user requirements. The system was developed and tested using a track evaluation method. Based on the quantitative results of the performance evaluation, track false positives and false negatives are identified. Using the feature characteristics of these failures, post processing parameters are designed to improve the system performance. The emphasis of the post track processing is on improving “effective” performance and is based on the practical requirements of the system. Figure 2 shows an example of track stitching performed by the post processing module. Figure 3 shows an instance of a false positive due to specular reflection which is removed by the post processing.

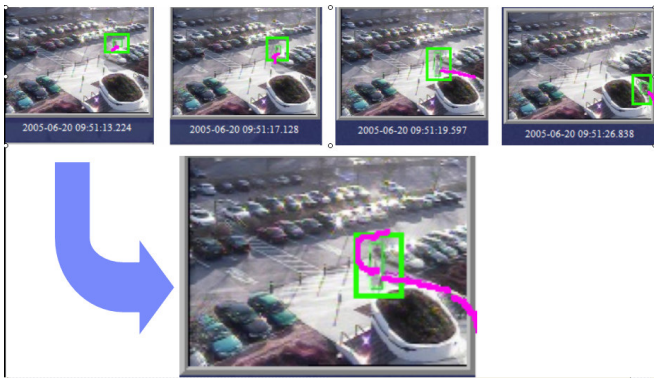


Figure 2. Illustration of track stitching: Top four boxes show four tracks resulting from a person walking slowly toward the building on a very bright day. The bottom image shows the "stitched" track.

Figure 3. Example of false positive due to specular reflection which is removed by track post processing.

3.3.2 Compound Alert Detection

In addition to the basic alarm capability of S3, a novel, multi-camera, spatio-temporal event detection system is available. This system lets users define multiple composite events of high-complexity, and then detects their occurrences in video sequences automatically. Thus, the system not only tracks the objects, but also detects semantically high-level, spatio-temporal events in the sequences. One of the biggest advantages of the system is the fact that the events of interest, are arbitrarily complex, and can be defined and communicated to the system in a generic way, which does not require any technical knowledge or familiarity with programming languages from the end-user. In addition, the events of interest are not pre-defined or hard-coded into the system. In our system, more sophisticated event scenarios can be built by using the primitive events, and combining them by operators. More importantly, the newly defined composite events can be named, saved and used as building blocks to combine them with other composite events. This layered structure makes the definition of events with higher and higher complexity possible. The defined events are written to an XML file, which is then parsed and communicated to the tracking engines running on the videos of the corresponding cameras.

With the proposed system, we have reached to the next level and managed to go from detecting "a person exiting the building" on an outside camera view to detecting "a person coming from the south corridor of the building and then exiting the building" by spanning two different camera views, and using composite event definitions. In the figures (Fig. 4 and 5) below two examples of complex spatio-temporal events are shown: tailgating at a vehicle gate and shoplifting in a retail environment.

Figure 4. Detection of three stages of vehicle tailgating: (left) gate opens (center) two cars detected in region in front of gate (right) gate closes.

Figure 5. Multi-camera composite event detection of shoplifting event at retail environment

4. The IBM Middleware for Large Scale Surveillance (MILS) and Data Management

4.1 The IBM Middleware for Large Scale Surveillance:

The IBM Middleware for Large Scale Surveillance (MILS) provides the data management services needed to build a large scale smart surveillance application. While MILS builds on the extensive capabilities of IBM's Content Manager and DB2 systems, it is essentially independent of these products and can be implemented on top of 3rd party relational databases. It supports the indexing and retrieval of

spatio-temporal event meta. MILS provides analysis engines with the following support functionalities via standard web services interfaces using XML documents.

A: Meta-data Ingestion Services: These are web services calls which allow an engine to ingest events into the MILS system. There are two categories of ingestion services

A.1: Index Ingestion Services: This allows for the ingestion of meta-data that is searchable through SQL like queries. The meta-data ingested thru this service is indexed into tables which allow content based search.

A.2: Event Ingestion Services: This allows for the ingestion of events detected in the SSE. For example, a loitering alert that is detected can be transmitted to the backend along with several parameters of the alert. These events can also be retrieved by the user but only by the limited set of attributes provided by the event parameters.

B: Schema Management Services: These are web services which allow a developer to manage their own meta-data schema. A developer can create a new schema or extend the base MILS schema to accommodate the metadata produced by their analytical engine.

C: System Management Services: These services provide a number of facilities needed to manage a surveillance system including

C.1: Camera Management Services: These services include functions for adding or deleting a camera from a MILS system, adding or deleting a map from a MILS system, associating a camera with a specific location on a map, adding or deleting views associated with a camera, assigning a camera to a specific MILS server and a variety of other functionality needed to manage the system.

C.2: Engine Management Services: These services include functions for starting and stopping an engine associated with a camera, configuring an engine associated with a camera, setting alerts on an engine and other associated functionality.

C.3: User Management Services: These services include adding and deleting users to a system, associating selected cameras to a viewer, associating selected search and event viewing capacities to a user and associating video viewing privilege to a user.

C.4: Content Based Search Services: These services allow a user to search through an event archive using the following types of queries.

C.4.1: Search by *Time* retrieves all events that occurred during a specified time interval.

C.4.2: Search by *Object Presence* retrieves the last 100 events from a live system.

C.4.3: Search by *Object Size* retrieves events where the maximum object size matches the specified range.

C.4.4: Search by *Object Type* retrieves all objects of a specified type.

C.4.5: Search by *Object Speed* retrieves all objects moving within a specified velocity range.

C.4.6: Search by *Object Color* retrieves all objects within a specified color range.

C.4.7: Search by *Object Location* retrieves all objects within a specified bounding box in a camera view.

C.4.8: Search by *Activity Duration* retrieves all events with durations within the specified range.

C.4.9: Composite Search combines one or more of the above capabilities.

4.2 Data models in the S3 System

The MILS system has three types of data models, namely, 1) the system data model which captures the specification of a given monitoring system, including details like geographic location of the system, number of cameras, physical layout of the monitored space, etc 2) the user data model which models users, privileges and user functionality 3) the event data model which captures the events that occur in a specific sensor or zone in the monitored space. Each of these data models is briefly described in the following subsections.

A) System Data Model: The system data model has a number of components, listed below.

A.1: Sensor/Camera Data Model: The most fundamental component of this data model is a view. A view is defined as some particular placement and configuration (location, orientation, parameters) of a sensor. In the case of a camera, a view would include the values of the pan, tilt and zoom parameters, any lens and camera settings and position of the camera. A fixed camera can have multiple views for different purposes. The view Id is used as a primary key to distinguish between events being generated by different sensors. A single sensor can have multiple views. Sensors in the same geographical vicinity are clustered into clusters, which are further grouped under a root cluster. There is one root cluster per MILS server.

A.2: Engine Data Models: The engine data model captures the following information about the analytical engines.

- Engine Identifier: A unique identifier assigned to each engine.
- Engine Type: This denotes the type of analytic being performed by the engine, for example face detection, behavior analysis, LPR, etc.
- Engine Configuration: This captures the configuration parameters for a particular engine.

B) User Data Model:

The user data model captures the privileges of a given user. These include

- selective access to camera views
- selective access to camera / engine configuration and system management functionality
- selective access to search and query functions.

C) Event Data Model

This data model represents the events that occur within a space that may be monitored by one or more cameras or other sensors. S3 uses the timeline data model which uses time as a primary synchronization mechanism for events that occur in the real world between sensors. The basic MILS schema allows multiple layers of annotations for a given time span. The following is a description of the schema:

- Event: An event is defined as an interval of time.
- StartTime: Time at which the event starts.
- Duration: This is the duration of the event. Events with zero duration are permitted, for example snapping a picture or swiping a badge through a reader.
- Event ID: This is a unique number which identifies a specific event.
- Event Type: This is a event type identifier.
- Other descriptors: Every analysis engine can generate its own set of tags. If the tags are basic types CHAR, INT, FLOAT, they can be searched using the native search capabilities of the database. However, if the tag is a special type (for example color histogram) the developer needs to supply a mechanism for searching the field.
- Figure 6 shows a fragment of an XML file describing an object track in a camera.

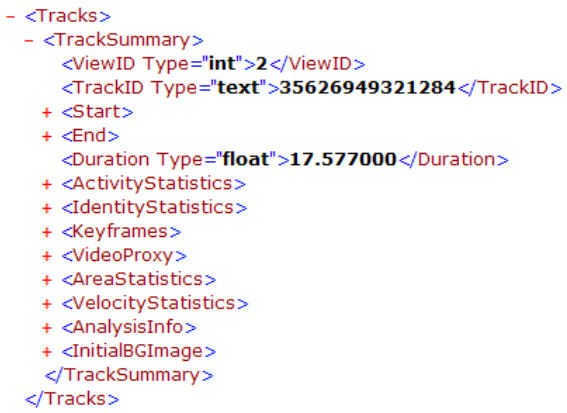


Figure 6 A fragment of object track meta-data represented in XML

5. Example Applications of IBM S3

There are many different applications for video surveillance systems including *homeland security applications* (airports, subways, seaports, critical infrastructure, etc), *retail loss prevention applications*, *retail business intelligence*, *casino gaming applications*, *forensic video investigations*, *manufacturing inspection & safety applications, financial sector applications,* etc. The IBM S3 system [3] is currently being piloted at a number of locations, including a New York area airport, an IBM facility in Asia/Pacific,

an IBM facility in Europe, an IBM facility in Middle East, several IBM research and corporate facilities in the New York area, and a retail store for loss prevention. We will describe some details of the pilot at the IBM Watson Research Center and the application at a retail store for retail loss prevention.

5.1 IBM S3 Pilot Installation at IBM Watson Research Center in Hawthorne, NY

The IBM S3 system currently operational at IBM Hawthorne processes video in real-time (from up to 14 cameras), generating meta-data for ~50,000 objects/events per camera per month. The meta-data is indexed and searchable using COTS (IBM DB2) technology with sub-second retrieval times. Figure 7 below shows a collage of results from the S3 system.

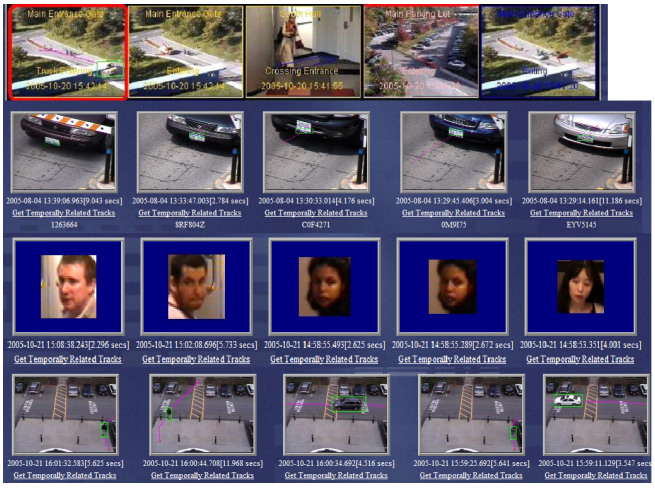


Figure 7: Results from Operational S3 System at IBM Research in Hawthorne, NY. Row 1: Real-time alert applet, icon pops up on screen with audio alert within 5 secs of the events occurrence. Row 2, 3, 4: *Combined results from multiple searches:* Row 2: License Plate Search, showing the cars entering the facility between 13:29 & 13:39 on Aug 8th 2005 below each car is the license plate number. Row 3: Face Capture Search, showing people entering the facility between 14:58 and 15:08 on Oct 21st 2005. Row 4: All activity in the parking lot (people & vehicles) between 15:59 and 16:01 on Oct 21st 2005.

Figure 8 – 13 show some interfaces of IBM S3 for the list of camera views, query results of car, query results of person, face capture, license plate recognition, and specific alert definition for the pilot in Hawthorne NY.

Figure 8: An Interface showing the various camera views currently available in the system.

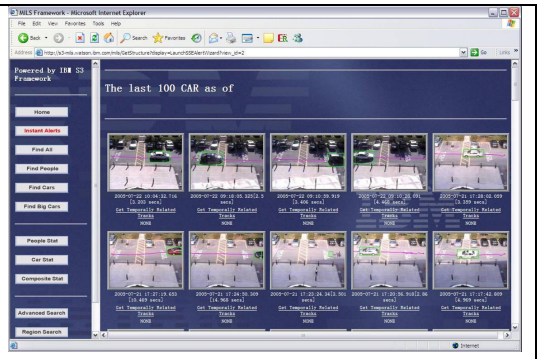


Figure 9: An Interface showing the Results from a "Find Cars" Query.

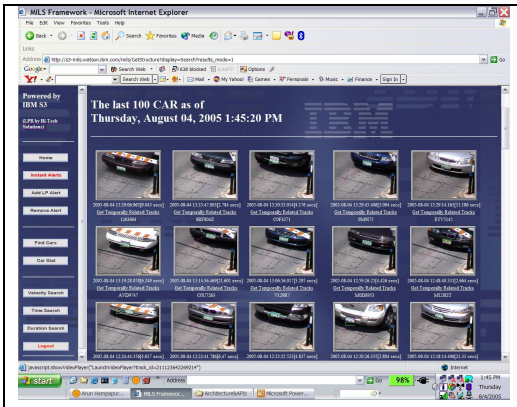


Figure 10: An Interface showing the Results of "License Plate Recognition."

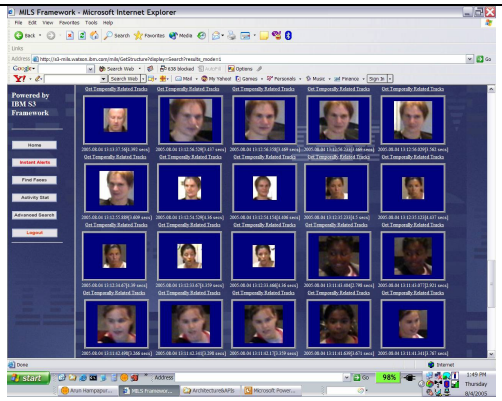


Figure 11: An Interface showing the Results of "Find Faces."

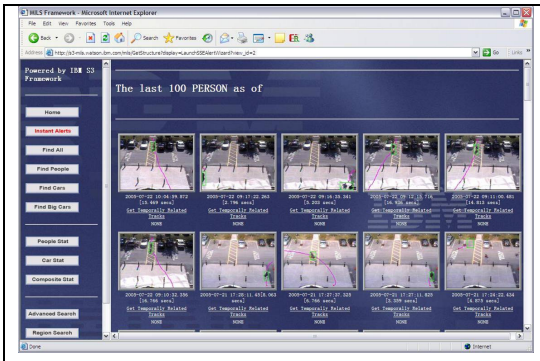


Figure 12: An Interface showing the Results from a "Find Person" Query.

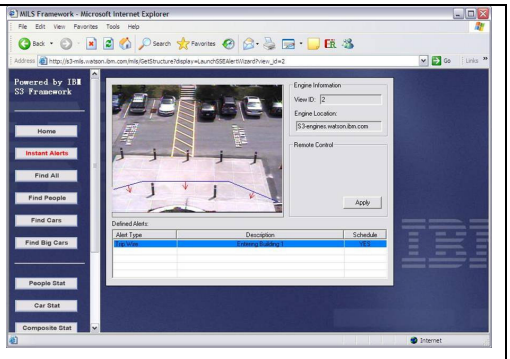
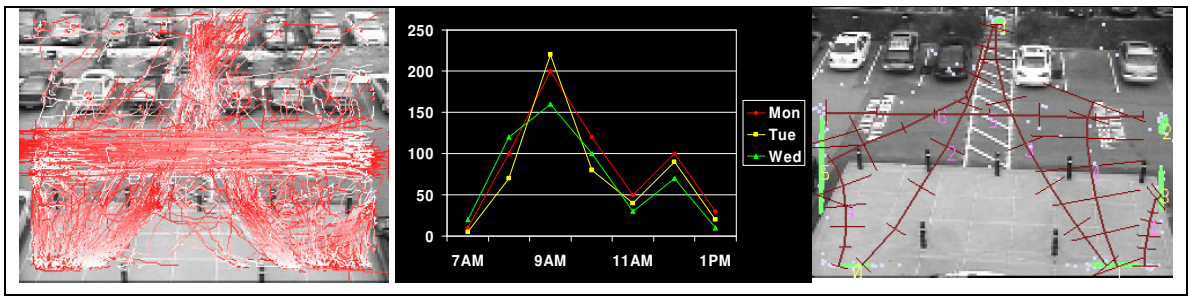


Figure 13: An Interface for defining specific alerts at a camera.

In IBM S3, we have a preliminary structure for detecting trajectory anomalies. This system shown in Figure 14 analyzes the paths of tracked objects, learns a set of repeated patterns that occur frequently, and detects when an object moves in a way inconsistent with these normal patterns.

The system begins by detecting object entrance and exit locations (referred to as sources and sinks.) Here the start and end points of tracks are clustered to find regions where tracks often end or begin. These points will tend to be where paths or roads reach the edge of the camera's field of view. Having clustered these locations, we have a simple classification for trajectories by labeling a track with its start and end location (or as an anomaly when it starts or ends in an unusual location such as a person walking through the bushes). We then apply a secondary clustering scheme to further detect anomalous behavior. The trajectories of all tracks with a given start/end location labeling are resampled and clustered together. This gives an average or "prototypical track" together with standard deviations, as shown in Figure 14. Thus most tracks from a given entry location to a given exit will lie close to the prototypical track, with typical normal variation indicated by the length of the crossbars. Tracks that wander outside this normal area can be labeled as anomalous and may warrant further investigation. Principal components of the cluster can also indicate typical modes of variation or "eigentracks" giving a more accurate model of normal vs. abnormal.



(a) (b) (c)

Figure 14 (a): Summary view showing the retrieval of trajectories all events that occurred in the parking lot over a 24 hour period. Trajectory color coding, start white and end is red. (b): Activity distribution over extended time period, X-Axis is time, Y-Axis is the number of people in the area. Each day of the week is shown with a different line. (c): Unsupervised behavior analysis. Object entrance/departure zones (green ellipses) and prototypical tracks (brown curves) with typical variation (crossbars).

5.2 IBM S3 Pilot Installation at Retail Stores for Retail Loss Prevention

"Shrinkage" is a catch-all term to describe a shortfall in the accounts of retail stores. Stores in developed countries may have a shrinkage of 1–2% [22, 23, 24] of sales, as indicated by comparing stock levels with actual sales, but the causes of this shrinkage are generally unknown. Shrinkage is unnecessary loss which businesses are keen to reduce, but reducing shrinkage is only possible after identifying the causes of shrinkage in a particular retail sector, chain or store. The main types of shrinkage are 1) Clerical error (miscounting stock, accounting errors); 2) Misplaced or "lost" stock; 3) Shoplifting; 4) Employee theft; 5) Theft by supplier; 6) Returns fraud; 7) Tag switching (putting a low-price tag on an expensive item); 8) Sweethearting (employee-customer collusion to obtain discounts or merchandise).

Video surveillance can play a part in reducing all of these sources of shrinkage. IBM S3 pilot at retail stores provides the following functionalities:

- Detecting customer service events
- Detecting entrance events
- Associating entrance events, customer service events, and transaction log information
- Indexing by appearance such as color, size, etc.
- Counting customers

Here, we describe the IBM S3 that was piloted specifically to detect occurrences of returns fraud. We focus on the type of return fraud which involves a person buying an item and taking it away, then returning to the store with the receipt, taking another of the items from the shelf and taking it to customer service and asking to return it for a refund. In some stores a liberal refund policy means that a receipt is not even necessary. Figure 15 shows a store layout (not to scale) with camera placements to detect customers at entrances and the customer service desk. Two cameras at the customer service counter record activity there, including capturing the appearance of customers returning items. A separate set of cameras point at the doors and capture all activity of people entering and leaving the store. Figure 16 shows the field of view of some of these cameras. Our approach to returns fraud is to segment events in each of these cameras, to

filter them and then provide an interface to allow the association of returns events with a corresponding door entrance event showing when the person comes into the store. Figure 17 shows one view of the interface. Fundamental to the application are the events detected through the use of visual tracking algorithms and displayed in the interface panes at bottom left and bottom right.

Detecting Customer Service Events: The bottom left pane in Figure 17 shows customer service events- the detection of customers carrying out transactions at the customer service desk. Customers are detected and tracked using an efficient hierarchical face tracking algorithm that forms part of the IBM Smart Surveillance System. In our face tracking, the frontal faces are first detected by a face detection plug-in with a skin tone filter to filter out the false faces. For continuously detected faces, a simple and efficient blob tracking method is employed to track the face based on size and location and update the track model, track history, and track state. The track model includes the track image, mask, and size. The track history includes the track length, area, and position. The track state indicates the track is an incipient track or a stable track. An incipient track will become stable if it continually exists for N frames. In our system, we set N = 2. To keep tracking the customer when the face detector failed or the customer turns away, a mean shift tracker is activated only when the track of the face region is stable. When the mean-shift tracker is running, only the track history and track state are updated. Combining the simple blob-based face tracker and mean-shift tracker brings the following advantages: a) it is less error prone compared to using mean shift all the way through, as in the long run mean shift can be distracted by similar colored background objects; b) the model distribution are updated more reliably which is harder to perform if mean shift is used all the time; c) it is much faster and more efficient. In the experiments, it was observed that this hierarchical structure performs 5 to 6 times faster compared to using mean shift all the way.

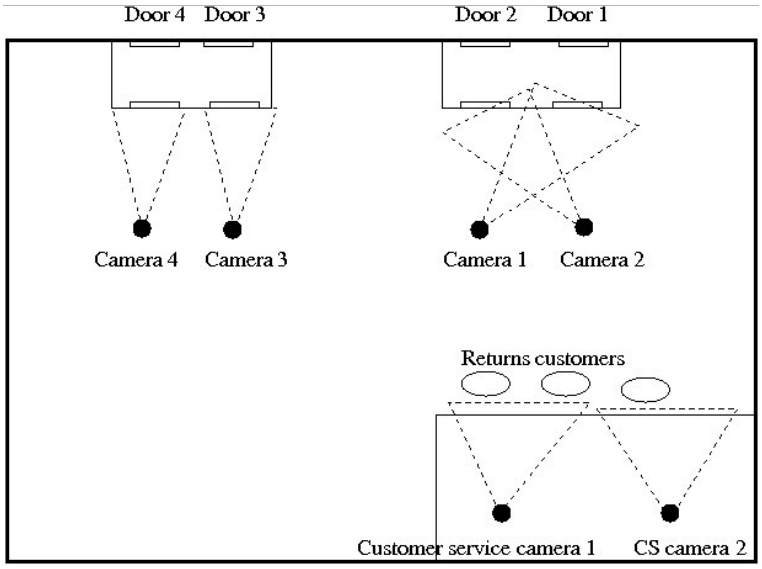


Figure 15: A sample store layout (not to scale) showing camera placements to detect customers at entrances and the customer service desk.

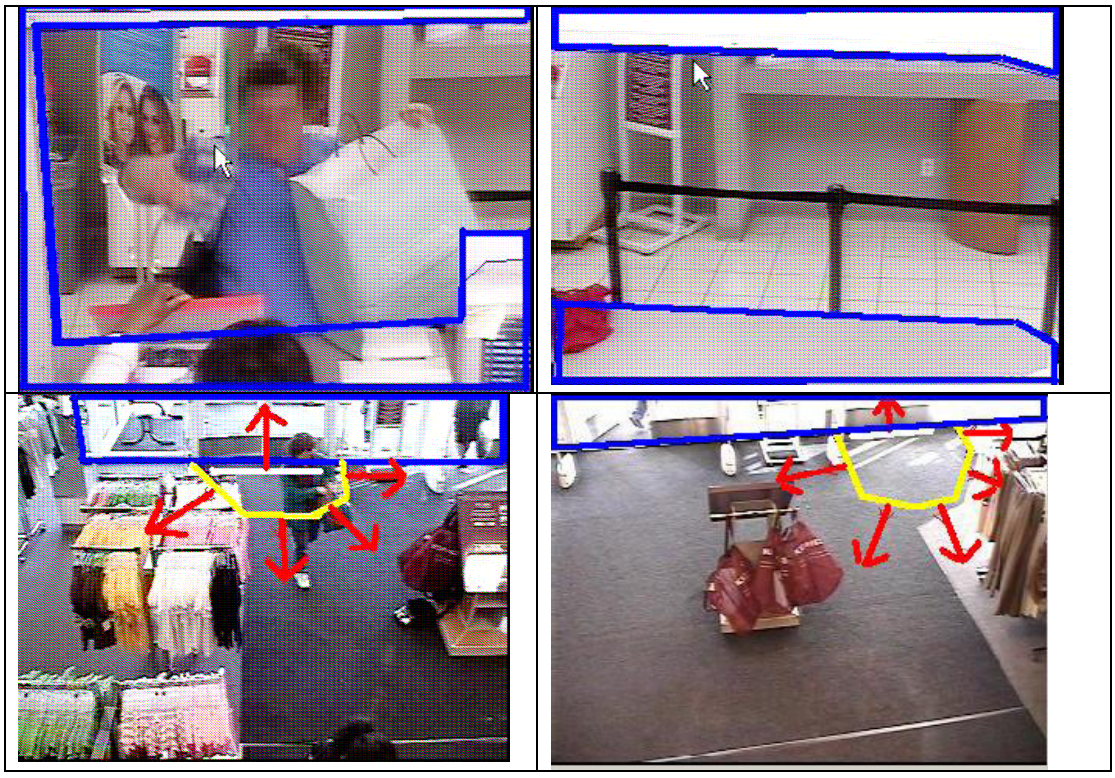


Figure 16: Views from four of the six cameras — two at the customer service desk (top) and two of the four doors (bottom). The "region of uninterested" is shown in blue for all frames and the entrance tripwires (enter in yellow, leave in white) are drawn on the views.

Figure 17: The user interface is divided into a controls panel (top), customer service events (left) and entrance events (right). These example results are restricted automatically to match the selected "red" search criterion—all matches are wearing red or pink clothing. All faces in this paper are pixilated for privacy.

Detecting Entrance Events: The lower right hand pane of the interface (Figure 17) shows entrance events — keyframes of every person detected entering the store. The entrance events are also detected using a tracker from the IBM S3. A separate tracker runs independently on each of four cameras — one for each customer entrance to the store. Detecting entrance events from the store doors is a challenging task, because of lighting, geometry and the presence of distracting (particularly door) motion. Here the resolution obtained is barely enough for face detection, and the angle obtained from ceiling-mounted cameras decreased the performance of face detection. While the cameras can be directed at the glass doors to frame completely customers entering, bright back lighting during the day, and dark night time scenes led us to point the cameras more steeply down into the more constantly illuminated carpet. Since this pilot used dedicated cameras, we were free to position and steer both customer service and entrance cameras, but we had no influence on the store environment such as layout, lighting, backgrounds etc.

Doors present an additional complexity in that their movement generates large scene changes that are not of interest in our application, but are not possible to model with background subtraction. By angling the cameras down, most of the door area was out of the cameras' fields of view, but the remaining visible door area was marked as a "region of uninterested" that is eliminated from background subtraction calculations. On these scenes (example frames are shown in Figure 16) we applied an adaptive background subtraction algorithm [15] which is a fast multiple-Gaussian algorithm that provides robustness, among other things to changes of lighting. This algorithm produces a foreground mask indicating moving objects that are not explained by the background model.

These foreground regions are then tracked using our "Color Field" probabilistic appearance model tracking algorithm (earlier versions of which have been described in [19, 20]). This models the shape and appearance of objects and allows pixel-wise resolution of occlusions of multiple objects, with continuous identity maintenance of objects in visual occlusions.

All detected activity is tracked and stored in the IBM S3. Much of the scene activity is not relevant for the Returns Fraud Prevention task, and is not presented to users of the RFP interface, but is available to other users carrying out other search tasks. The selective presentation of relevant material is carried out by using the tripwire alerts feature of the system to filter out only those tracks that correspond to a person entering the door. A directional tripwire is drawn in front of the door, and tracks crossing the tripwire are flagged as "entrance" events. Since the door region itself is marked as "uninteresting", detection of objects only takes place in front of the door region, so the tripwire is bowed out in front of the threshold, as shown in Figure 16.

Keyframe Generation: The tracking systems outlined above partition the video stream into a set of discrete events of interest. These events can be reasoned with (counting, looking at object appearance, trajectory etc.) and can be seen as a quantization of the video which allows more concise summarization and representation to the user. In the current user interface, each event is represented by two keyframes. The first uses the default keyframing policy of the tracking system, which is to present the full frame view

of the video for the frame when the tracked object had the largest visible area. (This correlates to it being closest to the camera and fully entered into the frame and thus with most recognizable details). Onto this frame, are drawn the bounding box (to distinguish which object the track represents if several are moving in the scene at once) and the trajectory of the object with direction indicated by color gradient and a "track start" icon.

The second keyframe is a "zoomed-in" view which shows a higher resolution view of the tracked object. In the case of the face tracker, this is just the detected face region (in a frame when the face was detected, rather than tracked, and with the largest area). For the entrance tracker, a head detection algorithm is used to try to extract image regions that correspond to the head-and-shoulders of the tracked person. This process uses two strategies for head detection, assigning a score to the "quality" of the region extracted. A history of such regions is maintained and only the best four for the track are stored in the database.

Indexing by Appearance: Once the computer carries out segmentation, detection and tracking as well as visualization, salient color of the customer is detected to match the entrance events and the customer service events. Salient color detection works by calculating a color histogram of the tracked objects at the entrance and storing the dominant peaks in the database. The histogram is computed in the cylindrical Hue/Saturation/Intensity color space. The cumulative histogram is computed only when the track is 2n frames old (n an integer >0) to minimize computational costs. White and black are defined as the high intensity/high saturation and low intensity/low saturation conic portion of the HSI cylinder respectively. The rest of the cylinder is divided uniformly by hue into 6 colors (red, magenta, blue, cyan, green, yellow). The dominant color is the peak in the 8-color cumulative histogram for the tracked object. At search time, the user can limit the displayed results to only those matching a particular dominant color worn by the customer at the customer service desk, using a pull-down menu of color names. Figure 17 shows the results when "Red" is selected.

Using the above affordances to browse through the events, if a match is found, the user can examine the keyframes and video to attempt to determine if fraud has taken place. An archive button allows the user to save the matched events for rapid future access, and preserves these events despite data expiration that may be enabled on the database.

Transaction Log Integration: Since the focus of the application is returns fraud, the forensic mode of investigation is helped considerably by the addition of a Transaction Log (TLOG) browser. The Transaction Log comes from the store database and consists of one record for every transaction carried out on any register in the store. The TLOG data is ingested into MILS using the same mechanism as is used for video events, and is browsed through another page of the web interface. A preliminary page allows the user to browse events by time, seeing histograms of TLOG event frequency over time, and then, within a given time period, to view the events of a particular type, including register number, transaction amount. Clicking on a particular TLOG event takes the user to a page showing visually detected returns at the same time, for the register in question, and the investigation can continue as previously described, but this time with direct, rapid access to only the events of the desired type (in most cases returns).

Customer counting: The counting of customers was achieved as a side-effect of the tripwire alerts used for filtering customer entrance events. An alarm statistics page that is part of the IBM S3 provides browsing and time slicing of alarm counts (viewing numbers of entrance events by hour, day or week across different periods). A second alarm on each door detects exit events which provide a corroborative count (which is in fact more accurate because, being on the true threshold, it does not pick up false alarms from internal traffic). The two can be used in conjunction to estimate the average time a customer spends in the store. While many “customer counting” solutions are available, using techniques such as beam-breakers and pressure pads. This application shows the flexibility of IBM S3 — the person counting was essentially available “for free” once the returns fraud solution was in place, and it provides richer, more useful data than a dedicated people counting solution would offer.

6. Conclusions and Future Directions

The IBM S3 system is designed to be an open framework for event based surveillance. One of the primary advantages of S3 is its ability to make the process of integrating technologies into IBM S3 easier. The use of a database to index events opens up a new area of research in context based exploitation of smart surveillance technologies. This will be one of the key future directions for our research. Additionally, the S3 system will be deployed in a variety of application environments including homeland security, retail, casinos, manufacturing, mobile platform security etc. Each new environment brings with it challenges both in the core video analysis domain and in the indexing and event interpretations domains. In the core video analysis domain, we will not only keep improving the system robustness for different environments, but also improve the system scalability. The current system can processes 4 video streams by one 2.8GHz PC, our goal is to process 8-16 video streams per computer. In the indexing and event interpretations domains, more complex event interpretations and long-term pattern analysis need to be explored.

7. References

[1] R. Collins, et al. `A system for video surveillance and monitoring', VSAM Final Report, Technical Report, CMURI-TR-00-12, May 2000

[2] M. Greiffenhagen, D. Comaniciu, H. Niemann, V. Ramesh, `Design,analysis and engineering of video monitoring systems: an approach and case study', The Proceedings of the IEEE, vol. 89, no. 10, pp. 1498-1517, October

[3] Arun Hampapur, Lisa Brown, Jonathan Connell, Ahmet Ekin, Norman Haas, Max Lu, Hans Merkl, Sharath Pankanti, Andrew Senior, Chiao-Fe Shu, and Ying Li Tian, Smart Video Surveillance, Exploring the concept of multiscale spatiotemporal tracking, IEEE Signal Processing Magazine, March 2005.

[4] Alan J. Lipton, Craig H. Heartwell, Dr Niels Haering, and Donald Madden, Critical Asset Protection, Perimeter Monitoring, and Threat Detection Using Automated Video Surveillance, white paper, ObjectVideo.

[5] IBM research, *PeopleVision Project Home Page* http://www.research.ibm.com/peoplevision/

[6] G. Stauffer, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 747–757, Aug. 2000.

[7] Haritaoglu, "Harwood and Davis, W4: Real time surveillance of people and their activities," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[8] T. Horprasert, D. Harwood, and L. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," in *Proc. IEEE Frame-Rate Workshop*, Kerkyra, Greece, 1999.

[9] Remagnino, Jones, Paragios, and Regazzoni, *Video Based Surveillance Systems Computer Vision and Distributed Processing*. Norwell, MA: Kluwer , 2002.

[10] VACE: Video Analysis and Content Exploitation [Online]. Available: http://www.ic-arda.org/InfoExploit/vace/

[11] Blanz and Vetter, "Face recognition based on fitting 3D morphable model," *IEEE PAMI*, vol. 25, no. 9, pp. 1063–1074, Sept. 2003.

[12] J. Phillips, P. Grother, R. Micheals, D.M. Blackburn, E. Tabassi, and M. Bone, "Face recognition vendor test 2002 P," in *Proc. IEEE Int. Workshop Analysis and Modeling of Faces and Gestures* (AMFG'03).

[13] Human ID at a Distance, U.S Government, DARPA Project.

[14] Combat Zones That See, U.S. Government DARPA Project.

[15] Ying-li Tian, Max Lu, and Arun Hampapur, "Robust and Efficient Foreground Analysis for Real-time Video Surveillance," IEEE CVPR, San Diego, June, 2005.

[16] Ying-li Tian and Arun Hampapur, "Robust Salient Motion Detection with Complex Background for Real-time Video Surveillance," IEEE Computer Society Workshop on Motion and Video Computing, Breckenridge, Colorado, January 5 and 6, 2005

[17] L.M. Brown, "View Independent Vehicle/Person Classification," ACM 2nd Int'l Workshop on Video Surveillance & Sensor Networks, Columbia University, New York City, NY, October 15-16, 2004.

[18] J. Connell, A.W. Senior, A. Hampapur, Y-L Tian, L. Brown, and S. Pankanti, 'Detection and Tracking in the IBM PeopleVision System," IEEE ICME, June 2004

[19] A.Senior, A.Hampapur, Y-L Tian, L. Brown, S. Pankanti, R. Bolle, "Appearance Models for Occlusion Handling," in proceedings of Second International workshop on Performance Evaluation of Tracking and Surveillance systems in conjunction with CVPR'01 December 2001.

[20] A. W. Senior, "Tracking with Probabilistic Appearance Models,' ECCV workshop on Performance Evaluation of Tracking and Surveillance Systems 1 June 2002 pp 48--55.

[21] Unpublished IBM Technical report, "Multi-view face detection by Harr and optimized wavelets features," 2006.

[22] Centre For Retail Research. The European retail theft barometer. Technical report, Centre For Retail Research, 2005. www.retailresearch.org.

[23] R. Hollinger. National retail security survey final report. Technical report, University of Florida, 2003.

[24] J. Guthrie. New zealand survey of retail theft and security report. Technical report, University of Otago, 2003.