Searching Surveillance Video

Arun Hampapur, Lisa Brown, Rogerio Feris, Andrew Senior, Chiao-Fe Shu, Yingli Tian, Yun Zhai, Max Lu
Exploratory Computer Vision Group ,IBM T.J. Watson Research Center, Hawthorne, NY, USA

Abstract

Surveillance video is used in two key modes, watching for known threats in real-time and searching for events of interest after the fact. Typically, real-time alerting is a localized function, e.g. airport security center receives and reacts to a "perimeter breach alert", while investigations often tend to encompass a large number of geographically distributed cameras like the London bombing, or Washington sniper incidents. Enabling effective search of surveillance video for investigation & preemption, involves indexing the video along multiple dimensions. This paper presents a framework for surveillance search which includes, video parsing, indexing and query mechanisms. It explores video parsing techniques which automatically extract index data from video, indexing which stores data in relational tables, retrieval which uses SQL queries to retrieve events of interest and the software architecture that integrates these technologies.

1. Introduction

While applying video analytics to provide real-time alerting based on predetermined event definitions, such as "tripwire," has been explored both in the research literature and in commercial systems, the challenges of searching through surveillance video remains largely unaddressed. While video analysis and pattern recognition technologies are at the core of "intelligent" or "smart surveillance," effective search of surveillance video requires research into searchable meta-data representations for video based features, data models for indexing and correlating diverse types of meta-data, and architectures for integrating technologies into large scale systems.

Searching surveillance video essentially revolves around the following key search criteria:

- Specific search for people and vehicles
- Generic search for objects and events of interest

Search applications require a combination of these criteria to create composite queries and the ability for the search to be applied across multiple cameras distributed over a spatial region.

In this paper we use the IBM Smart Surveillance System (aka S3) as an example system for discussing various aspects of the technology involved in search. Section 2 presents a short overview of related research and commercial efforts around searching surveillance video. Section 3 describes the software architecture of the S3. Sections 4 presents the data model for surveillance events, and 5 presents an overview of the parser. Sections 6 and 7 explore the various aspects of searching for people, events and objects. Sections 8 and 9 present composite queries and the concept of spatio-temporal searching. Section 10 discusses performance metrics for search systems. We conclude the paper with a discussion of the significant research challenges that remain in enabling large scale searching of surveillance video.

2. Previous work

Surveillance video analysis has been extensively studied. However, compared to the vast amount of research in broadcast video search [8,12], very few systems address the search in surveillance video. Lee and Smeaton [9] describe a user interface to retrieve simple surveillance events like presence of person and objects. Stringa and Regazzoni [19] proposed a content-based retrieval system for abandoned objects detected by a subway station surveillance systems. In their system, similar abandoned objects can be retrieved using feature vectors of position, shape, compactness, etc. Berriss et.al. [2] utilized the MPEG-7 dominant color descriptor to establish an efficient retrieval mechanism to search the same person from surveillance systems deployed in retail stores. Meesen et.al. [11] analyzed the instantaneous object properties in surveillance video key-frames, and performed content-based retrieval using a generic dissimilarity measure which incorporates both global and local dissimilarities between the query and target video key-frames. There is significant effort in industrial surveillance systems [1,13,15] targeted toward real-time event detection. Very few of these systems have focused on video search. 3VR[1] does provide capabilities to search for a person based on face recognition. In summary, there is a very limited number of both research and commercial systems focused on searching surveillance video. As a surveillance systems grow in scale and utility, there is an increasingly critical need to provide the corollary search capabilities.

3. Software Architecture

Enabling content based searching of surveillance video involves the following steps:

Step 1: Parsing surveillance video into time intervals corresponding to "events of interest"

Step 2: Extracting meta-data descriptors for these intervals and indexing it into database tables

Step 3: Providing the query interface and result reporting mechanisms for surveillance events

Figure 1 shows the software architecture of S3 (for details refer to [18]) which supports the above three steps with the following software components

Smart Surveillance Engine (SSE): The SSE supports step 1 by parsing the surveillance video. It is designed to process one stream of video in real-time, extracting object meta-data and evaluating user defined alerts. The SSE uploads messages in XML to the central data repository. The SSE provides the software framework for hosting a wide range of video analytics like behaviour analysis, face recognition, license plate recognition etc.

Middleware for Large Scale Surveillance (MILS): MILS provides the algorithms needed to take the event meta-data and map it into tables in a relational database. Additionally, MILS provides event search services, meta-data management, system management, user management and application development services. MILS uses off the shelf data management (IBM DB2), web server (IBM Websphere Application Server) and messaging software (IBM MQ) to provide these services.

Solutions: These are mainly web applications (HTML, Java, JSP, applets, Javascript) which use the web services provided by MILS to provide the functionality needed by the user to query the database and view the results.

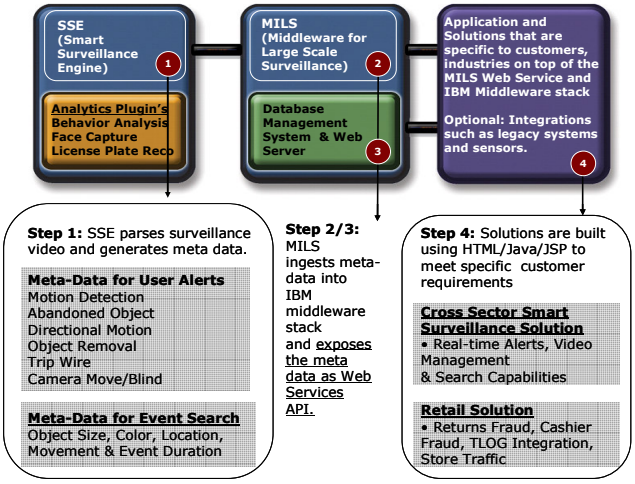

Figure 1: Software architecture of the IBM Smart Surveillance System (S3). This architecture supports both "user defined alerts" and "searching" based on surveillance camera video.

4. Extensible Data model

The most fundamental index into surveillance video is the time of occurrence of an event. The challenge is to automatically derive the time of occurrence of "events of interest" by analyzing the video. Once an event is detected in video, the time interval of the event can be annotated with additional meta-data which captures a more detailed description of the event. Hence, the most basic data model for surveillance events is a time interval. Table 1 below shows the basic data model for two types of surveillance events 1) a car driving thru a parking lot captured on camera 23 and 2) the license plate of a car recognized on camera 35

Example Data Models	
Behavior Meta-data	**License Plate Meta-data**
Camera ID: 23	Camera ID: 35
Unique Event ID: 2379406	Unique Event ID: 4926402
Start: 9/10/06:02:22:15:100	Start: 9/10/06:02:12:15:100
End: 9/10/06:02:22:55:300	End: 9/10/06:02:12:25:453
Keyframe: 23567.jpg	Keyframe: 563783.jpg
Video : //mils/xx/file1.wmv	Video : //mils/xx/file3.wmv
Object Type: Car	License Plate #: 525sds
Additional Fields: (trajectory, color, shape, size, etc)	Additional Fields: (e.g State of Origin)

Table 1. Event time is used as the basis for annotation surveillance events

Each unique event that occurs within a scene is assigned an event identifier which is guaranteed to be unique across all cameras that are being indexed into a single database instance. The event ID is used as the primary key to select from and join across multiple tables in the database. The time of occurrence of the event is used to correlate events across multiple cameras that exist in the system. This data model can easily be extended to accommodate new types of meta-data as new types of video analytics are added to the system. If the meta-data is one of the basic types (INT, CHAR, FLOAT etc.) supported by the database it can be searched using SQL. For special types of meta-data, like color histograms additional user defined search functions have to be developed.

5. Parsing surveillance video

The most basic approach to segmenting surveillance video into events involves detection and tracking. The specific nature of the detection and tracking vary based on the type of video analysis technique used. For example, as a car (person) enters a camera, the SSE would detect the entry of the license plate (or face) into the view and recognize and track it until the car (or person) leaves the camera field of view.

The most widely applicable form of surveillance video parsing uses moving object detection and tracking. In common with most video surveillance systems, we use background subtraction [21] to detect changes in a video stream. Background subtraction works by maintaining a statistical model of the observed values of a pixel and modeling the variations to distinguish a change caused by a moving object from changes due to lighting changes or camera vibrations. The detected objects are tracked over their life within a single camera using a tracking system [16]. The tracker associates multiple detections of the same object over time and constructs tracks which represent the movement of a single object (or sometimes the coherent motion of a group of objects). Since it corresponds to a physical object, the track (which designates a time interval) is the fundamental representation in the database. For a given object, we can derive characteristics, such as the object's type,

appearance and identity, which are assumed to be constant over time, although our estimates of these characteristics may be derived from accumulations of multiple observations of the object over time. The following sections discuss how the various attributes of objects can be extracted to enable searching.

6. Specific People & Vehicle Search

6.1. Searching for People

Faces are the key to identifying people. Automatically recognizing people from surveillance cameras still remains a challenging problem for face recognition technologies [6,17]. The first step in achieving automatic face recognition is the indexing of video with a "presence of people index." While face-based people detection is valuable, in most realistic scenes, it isn't sufficient to enable people searching because people:

- could be entering the scene with a pose which limits the visibility of the face from the camera,
- people could be facing away from the camera, in which case face capture / reco will fail

Our approach to creating a "presence of people index" uses a combination of face and people detection to ensure a very low rate of false negatives.

Our face detection method relies on extracting adapted features to encode the local geometric structures of training samples prior to learning. Local feature adaptation is carried out by a non-linear optimization method that determines feature parameters (such as position, orientation and scale) in order to match the geometric structure of each training sample. In a second stage, Adaboost learning is applied to the pool of adaptive features in order to obtain general features, which encode common characteristics of all training face images and thus are suitable for detection. Compared to other techniques (e.g., [14]), our method offers faster learning time and improved detection rates (see [7] for quantitative evaluation on standard datasets).

After detecting a face in the field of view of a surveillance camera, we apply a correlation-based tracking algorithm to track the face in the subsequent video frames. Continuous face detection is used to re-initialize the tracker, using multiple view-based classifiers (frontal and profile) interleaved along the temporal domain in the video sequence.

In addition to detecting and tracking human faces, we also store a keyframe for each captured face image in the database, associated with a timestamp. This allows the user to query the system like ``Show me all people who entered the facility yesterday from 1pm to 5pm." An example of this search is shown at the right of Table 1.

Ideally, for every person passing thru the scene, a face keyframe would be generated and stored in the database. However, due to false negatives in face detection and face pose and person orientation issues, important events might be missed. We address this problem by using a keyframe selection technique that combines a face classifier with a person classifier. If a face is detected and tracked in the video sequence, a face keyframe is stored in the database. Otherwise, a person keyframe is generated if a person is detected and tracked in the video.

We analyzed ten hours of data (from 10 days), with each hour corresponding to the peak hour (i.e., the hour with most people entering the facility) in each day. Table 2 shows our results. Out of 445 people entering the facility (not walking back to the camera), we captured 351 faces, with only 7 false positives. The reason that some faces were missed is that sometimes people enter the door looking down, occluding the face from the camera, which is placed in the ceiling. By running our keyframe selection technique (using face and person detectors), we can capture all remaining 94 persons, as well as 40 persons walking back to the camera, with an additional 19 false positives.

Total # of people approaching camera		445
Total # of people receding from camera		40
Face Detection	Faces Captured	351
	Faces Missed	94
	False Poitives	7
Person Detection	Persons captured	134
	Approaching	94
	Receding	40
	False Positives	19
Overall People False Negatives		0
Overall People False Positives 26/445		5.6%

Table 2: Results obtained from ten hours of surveillance video. Example faces (frontal & profile) captured by our system (blurred to preserve privacy)

6.2. Searching for Vehicles

Searching for vehicles based on license plates is achieved using license plate recognition technology, which is very advanced when compared to the state of face recognition. Unlike human faces, license plates vary widely based on geography. Variations include language, font, background and numbering scheme. Typically, there is no single algorithm or company which can recognize license plates across wide geographies. One approach to handling this variation is to standardize the interfaces to the license plate algorithms and standardize the meta-data representation for the license plate. The software architecture of S3 supports this approach.

7. Generic Search Criteria

Generic search includes search for objects and behaviors of objects over time. This search can be qualified by one or more of the following: object color, object class, object size, object shape, object location, object movement, time of event of occurrence and event duration.

7.1 Search by Object Color: Object color is determined by (1) converting RGB object colors to a 6 color Hue/Saturation/Intensity (HSI) space, (2) periodically

updating and normalizing the 6 color HSI cumulative histogram over the life of the object and (3) determining the three dominant colors and their percentages. For vehicle color estimation, the final primary color is determined based on hue if sufficient (regardless of the amount of achromatic pixels), and the relative amount of black and white. Table 3 shows the results of color classification for vehicles entering and exiting our facilities for a total of 8 hours (4 hours for two days). The overall correct color classification is 80%. Over half the misclassified vehicles are black or white vehicles misclassified as white or black respectively. Although this may be improved with parameter tuning taking into consideration the variations in lighting conditions, the most significant issue here is due to the variable amount of shadows included in the object segmentation and the percent of the true black components for each vehicle (i.e. windshield size, tires, accessories etc.) Figure 2 and 3 shows illustrative examples of vehicles classified correctly and incorrectly.

7.2 Search by object class: Object classification is performed using a view invariant classifier[3]. An object can be classified as either a person or a vehicle based on shape features such as compactness and principal axis ratio, and motion features such as speed and degree of recurrent motion. Table 45 shows results for vehicles and people entering the front of our laboratory for 4 hours one morning. (May 16 2007, Camera #2, between 8AM and 12PM). Overall 307/334 or 92% of the vehicle/person object tracks were correctly classified.

7.3 Search by Object Size: Object size is often useful to determine object class for objects moving orthogonally to the camera viewpoint. Object size was used to distinguish pedestrians from vehicles and to distinguish standard vehicles (cars, SUVs, minivans) from mid-size vehicles (delivery trucks, large pickups) and large trucks (such as garbage trucks and tractor trailers) for our camera looking orthogonally to the entry road. Table 4 shows the results of a size search used for object classification.

7.4 Search by Object Shape: Currently our system does not support explicit search by shape. However as described in the object classification section, shape of objects is used to determine the class.

7.5 Search by Object Movement:

Object movement can be qualified by several parameters such as speed, acceleration, direction, and extended properties of the objects trajectory (like find all people walking in a zig-zag manner thru the parking lot). The SSE computes several of these parameters for use in evaluating user specified events like directional motion of the object. At this time, our search interface only provides the ability to search based on the speed of the object.

7.6 Search by object location: This is achieved by storing the entire trajectory of the object into the database. The tracker (described in section 5) generates a trajectory for each moving object in the scene in image coordinates. When the user selects a region of interest (ROI) within an image (yellow box), this is used to generate a SQL query which retrieves all the objects whose trajectories intersect with the ROI. Figure 4 (left) shows the results of events recovered when the user selects the yellow region outlined in the image.

	COLOR SEARCH → GROUND TRUTH						
COLOR SEARCH→ RESULTS	BL	WH	RE	YE	BU	GR	
BL	119	36	20	0	1	0	165
WH	3	102	1	0	1	0	107
RE	0	0	18	0	0	0	18
YE	0	0	0	1	0	0	1
BU	0	0	0	0	7	0	7
GR	0	0	0	0	0	2	2
	122	138	39	1	9	2	

Table 3 Color Results: BL-Black, WH-White, RE-Red, YE- Yellow, BU-Blue, GR-Green

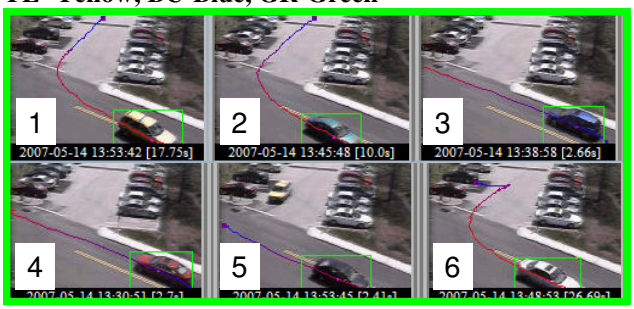

Figure 2 Retrieved keyframes (cross indexed to video by time) (1) yellow, (2) green, (3) blue, (4) red, (5) black and (6) white vehicles. (Trajectory color indicates direction of movement, blue is track start, red is track end).

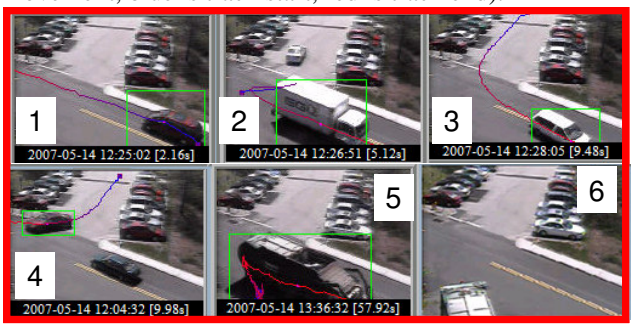

Figure 3 Keyframes 1,2,3 show errors from searching for black objects. Keyframes 4,5,6 are results of searching for white objects. For 1,2,3 notice the dark shadows and color of windows etc which lead to misclassification. The garbage truck appears to be black in the keyframe (5) but in playing , it appears that the truck including the lower body is white.

7.7 Search by time of occurrence of events:

Every event indexed into the database is required to have an event start timestamp and event end timestamp (see section on data model). These time stamps are used to retrieve events within the user specified time of interest. Currently we only support retrieval of events that occurred a) before a user specified time b) after a user specified time c) during a user specified interval.

OBJECT CLASS RESULTS	Object Class Ground Truth			
	V	P	O	
V	77	0	1	
P	9	230	19	
O	1	18	53	
	86	248	73	

SIZE SEARCH → RESULTS	SIZE SEARCH → GROUND TRUTH					
	P	C	MS-T	L-T	O	
P	17	2				
C	3	39			20	
MS-T		3	1			
L-T				1		
O					11	
Total	20	43	1	1	31	

Table 4 Left: Object Classification Result: V→ Vehicles, P→ Person, O →Other Right: Search using object size. P→Person, C→Car, MS-T → Medium Sized Truck, L-T→ Large Truck, O → Other.

7.8 Search by duration of event: Every event recorded by the system has an associated time duration. The duration of an event can be used for multiple purposes. Below are sample events from a query for events of duration longer than 50 seconds. These sample events demonstrate how loitering can be detected by using the event duration query (figure 4 right).

8. Composite Search

All the criteria discussed above can be combined into a single query to search for events of much higher degree of complexity. Consider the following scenario. Employee's at a facility have registered a complaint that one of the drivers from an express mail company is driving very fast in the parking lot. Knowing that, the delivery truck is yellow, we can use the composite query as follows:

FIND ALL, Object Type = "VEHICLE", Object Size > X1 & Object Size <X2, Object Color = Yellow, Object Speed > S1

Applying such a query to events over a month would help establish a pattern of speeding violations committed by the delivery truck, thus narrowing down the specific driver.

9. Spatio-temporal Search

In a number of applications, the events of interest are a combination of basic events over space (cameras) and time. In an earlier work [20], we propose a spatio-temporal event detection system which lets the users to specify multiple composite events of high-complexity, and then detects their occurrence automatically. Events can be defined on a single camera view or across multiple camera views. Semantically higher level event scenarios can be built by using the building blocks, which we call the primitive events, and combining them by operators. More importantly, the newly defined composite events can be combined with each other. This layered structure makes the definition of events with higher and higher complexity possible. The event definitions 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 the next level and managed to go from detecting "a person exiting the building" to detecting "a person coming from the south corridor of the building and then exiting the building."

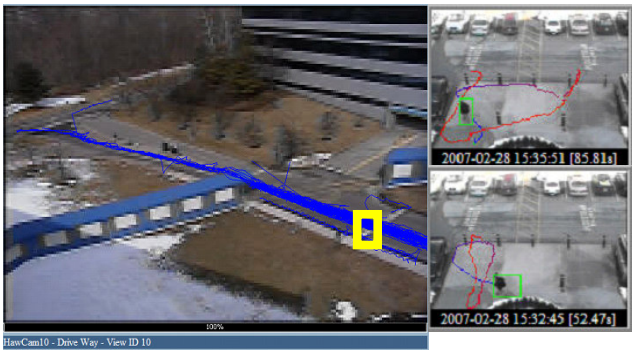

Figure 4. (Left) Results of spatial search, showing the trajectories of all objects that passed thru the user selected yellow region. User can click on the trajectory to index into the video clip**. (Right)** Loitering events (note the long person trajectories) retrieved by using the event duration query.

10.Search Performance

The performance of a search system can be characterized along two dimensions.

Precision & Recall: This is a measure of how well the system is meeting the requirements of the user's query. The precision and recall of the overall system is a function of the precision and recall of each of the individual video parsing mechanisms (face, license plate, etc). The previous sections have presented the results for people detection. The precision and recall of all of the other retrieval techniques such as color, size, location, event time, and event duration are dependent on the precision and recall of the underlying event parsing system (object detection and tracking). A detailed evaluation of event parsing (detection and tracking) can be found in [4][5].

Table 5 Test Set Description

Test Data Set Description with ground truth	
Data volume	4 cameras, 10 sequences, 36 minutes (2267 frames)
Total # of hand marked objects	2964 objects in 2267 frames
Total # of objects tracks FOV	90 objects

As shown in table 5, we used a test set of videos which was hand marked by a person for objects in each frame and tracks over the sequence. The results of running our base object detection and tracking algorithms are shown in tables 6 and 7. At the selected operating point (set of thresholds of object size, sensitivity of detection thresholds, track match thresholds, etc) the base performance of the detection and tracking algorithms is good on objects that are of significant size (above 169 sq pixels). The false positives when measured at a track level tend to be very short lived trajectories (77 frames, less than 3 secs). Typically, events that occur in the real world are of significantly longer duration. While improvement in the base event parsing is always desirable, the current level of performance is more that adequate for a search and real-time alert in retail & city surveillance

Table 6 Object Detection Performance Summary

Object Detection or Background Subtraction Results	
False Positives	0.03 objects per frame
False Negatives (missed object)	628 of 2964 = 21.2%
Avg size of missed object	226 sq-pixels

Table 7 Object Tracking Performance Summary

Object Tracking Results	
False Positives (spurious tracks)	25 with average length of 77 frames
False Negatives (missed tracks)	24/90 = 26.6%
Avg size of missed tracks	169 sq-pixels

Retrieval Time: This is the time between the user launching a query and the system responding with results. This time varies widely based on the type of query, with location searches being the most expensive and searches based on native SQL types falling into a different bucket. Below is a sample performance result for color retrieval, which is a native SQL query.

Table 8: Retrieval time result summary

Database Server, Dual Xeon, 3.8Ghz with 4GB Ram running IBM DB2	
Total number of events on Main Parking Lot Camera	From Apr 30, 07 to May 14, 07 10997 events over 15 days
Red car search	219 events retrieved in under 5secs

11. Conclusions

Enabling effective search of surveillance video is a challenging problem, as it involves not only the challenges of understanding events and activities in video, but also the challenges of generating searchable meta-data and indexing into a database, providing search and visualization mechanisms. The current activities in research and industry have barely begun to scratch the surface of the challenges involved in search video. This paper presented a framework for addressing surveillance search and demonstrated several aspects of searching surveillance video. Huge challenges and research opportunities remain open in the space of surveillance video search, from dealing with the challenges of searching for colors under varying lighting, to searching for people, to indexing large amounts of video and making it searchable, to providing intuitive interfaces for enabling search and interaction, with the holy grail being to reduce the time to investigate situations like the London bombings.

References

[1]. 3VR, http://www.3vr.com/Products/#smartsearch.

[2]. W.P. Berriss. W.G. Price and M.Z. Bober, "Real-Time Visual Analysis and Search Algorithms for Intelligent Video Surveillance",International Conference on Visual Information Engineering, 2003.

[3]. L.M. Brown, "View Independent Vehicle/Person Classification" ACM 2nd Int'l Workshop on Video Surveillance & Sensor Networks, 2004.

[4]. Brown, L.M.; Lu, M.; Chiao-Fe Shu; Ying-li Tian; Hampapur, A.; Improving performance via post track analysis IEEE VS-PETS 2005.

[5]. Lisa M. Brown, Andrew W. Senior, Ying-li Tian, Jonathan Connell, Arun Hampapur, Chiao-fe Shu, Hans Merkl, and Max Lu, " Performance Evaluation of Surveillance Systems Under Varying Conditions," IEEE Int'l Workshop on Performance Evaluation of Tracking and Surveillance, Colorado, Jan., 2005 .

[6]. Face Recognition Vendor Test (FRVT), http://www.frvt.org/FRVT2006/

[7]. Rogerio Feris, Ying-li Tian and Arun Hampapur, Capturing People in Surveillance Video, The Seventh International Workshop on Visual Surveillance at IEEE CVPR 2007, Minneapolis, MN

[8]. A. Hauptmann, "Lessons for the Future from a Decade of Informedia Video Analysis Research", International Conference on Image and Video Retrieval, 2006.

[9]. H Lee, A Smeaton, N O'Connor, N Murphy, User Interface for CCTV Search System, Imaging for Crime Detection and Prevention, 2005. ICDP 2005.

[10]. L. Marcenaro, F. Oberti, G.L. Foresti and C.S. Regazzoni, "Distributed Architectures and Logical-Task Decomposition in Multimedia Surveillance Systems", Proceedings of IEEE, Vol.89, No.10, 2001.

[11]. J. Meessen, M. Coulanges, X. Desurmont and J.F. Delaigle, "Content-Based Retrieval of Video Surveillance Scenes," Multimedia Content Representation, Classification and Security, 2006.

[12]. M. Naphade, et al. "On the Detection of Semantic Concepts at TRECVID", ACM International Conference on Multimedia, 2004.

[13]. ObjectVideo, http://www.objectvideo.com/products/vew/.

[14]. P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features", IEEE CVPR'01.

[15]. PyramidVision, http://www.pyramidvision.com/.

[16]. A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, R. Bolle, Appearance Models for Occlusion Handling, in Journal of Image and Vision Computing , November 2006

[17]. A.W.Senior, L. Brown, C.-F.Shu, Y.-L.Tian, M. Lu, Y. Zhai, A. Hampapur, Visual Person Searches for Retail Loss Detection: Application and Evaluation, International Conference on Vision Systems 2007, Bielefeld, Germany

[18]. Chiao-Fe Shu, et al, IBM smart surveillance system (S3): a open and extensible framework for event based surveillance, IEEE AVSS 2005 Page(s):318 - 323

[19]. E. Stringa and C.S. Regazzoni, "Content-Based Retrieval and Real Time Detection from Video Sequences Acquired by Surveillance Systems", IEEE ICIP, 1998.

[20]. Velipasalar, S.; Brown, L.M.; Hampapur, A, Specifying, Interpreting and Detecting High-level, Spatio-Temporal Composite Events in Single and Multi-Camera Systems, CVPR Worskop SLAM, 2005.

[21]. Y Tian, M.Lu, Hampapur, Robust and efficient foreground analysis for real-time video surveillance, CVPR 2005