Chapter 1

DISTRIBUTED ACTIVE MULTI-CAMERA NETWORKS

Andrew Senior, Arun Hampapur, Lisa Brown, Ying-Li Tian, Chiao-Fe Shu and Sharath Pankanti
{aws,arunh,lisabr,yltian,sharat,cfshu}@us.ibm.com
IBM T. J. Watson Research Center,
PO Box 704, Yorktown Heights, NY 10598

Abstract We present the IBM Smart Surveillance System that uses a distributed architecture to manage a heterogeneous network of active cameras. This system consists of a distributed network of cameras, each with local processing that interprets video to detect and track moving objects. The system performs multi-camera tracking as objects pass through the fields of view of different cameras, and actively acquires rich, high-resolution data by actively tracking objects of interest with Pan-Tilt-Zoom cameras. The multi-resolution data is stored in a shared index that can be browsed and searched live or post-hoc from a remote location, visualizing very low bandwidth video or activity meta data.

Keywords: Video surveillance, tracking, detection, database, multi-scale, multi-camera.

1. Introduction

Ambient intelligence involves putting information processing devices into the world and delivering intelligently processed information to distributed information consumers — with the ultimate purpose of providing utility to human users. This process will see more and more devices embedded into everyday objects and into the environment (e.g. cars, buildings and street furniture). The aim of ambient intelligence is not to make processing power ambient, for there are economies of scale in putting as much processing as possible into dedicated farms, but to make the sensors and actuators on these devices ambient and to deliver the power of information processing ubiquitously. Efficiencies of communication necessitate the processing power and thus the intelligence itself being local to the sensors and actuators and thus equally ambient.

2. Sensing modalities

As myriads of small, specialized computing devices are deployed in the world, we see them becoming markedly different from the traditional computing paradigm of a processor box with invariable user interface components of a visual display screen, mouse and keyboard. While many specialist sensors will be developed — to detect biological agents, cosmic rays and the like, or perhaps to assist environmental regeneration by controlled release of nutrients or organisms- it is likely that a huge proportion of the devices that will be deployed will be for the purposes of interacting with humans. These sensors will be a mix of active — detecting communication directed at them by “users”— and passive — sensing activity that is taking place within range.

Vision, we argue, is a very important sensing modality for both kinds of sensors. Vision provides a rich source of information about the world- and each sensor is able to receive data from a relatively long range. Vision can tell us about the shape of the world (from stereo and from a wide range of other cues), and is sufficient for identifying people (particularly face recognition, but also gait, ear shape and lip motion [2]). Vision can also capture a wide range of human communication (writing, facial expression, gesture, and body language, even lip-reading [8]). In interacting with people, vision is particularly important because it is the predominant modality through which most humans acquire information, and it is useful and in some cases necessary for machines to “see the world as we see it”.

Sound is an important complementary sensing modality that is very rich source of information when sensing human activity, because of the importance of speech for human communication, and because of its omnidirectionality. As microphone arrays, beamforming and source separation techniques improve, sound signal aquisition is improving in range and quality.

An expanding host of other sensing modalities is available, sensing motion, position, orientation, electric, magnetic and gravitational fields, atmospheric chemicals, and biological agents.

The acquisition of this data provides a challenging fusion problem in creating, and visualizing a rich, multi-modal model of the world. However the complementary problem of filtering this torrent data down to extract the interesting facts that can be usefully acted upon or are worth storing, then delivering the information to mobile users, provides a greater, Ambient Intelligence, challenge

3. Vision for Ambient Intelligence

Because of the importance of vision, as a rich source of passively-available, informative data about the world, as discussed above, we concentrate on vision as a modality for sensing. This modality is particularly useful as a method of building up an understanding of the world for an ambient intelligent system, though it is naturally most powerful when combined with many other sensing modalities. The use of vision is also driven by the growing demand for, and feasibility of, practical systems for visual understanding of the world, particularly in the domain of visual surveillance.

Visual surveillance is an interesting domain inherently suited to ambient intelligence. Conventional surveillance systems have networks of cameras (now beginning to number in the thousands at large installations such as airports) distributed over a wide area. The video from these cameras has traditionally been centralized in a single control room where the video is recorded and observed by a small number of security guards. It is well known that security guards quickly lose alertness when facing banks of monitors displaying empty scenes. The potential for a relentless watcher of every single channel of video with perfect recall has driven the development of intelligent systems tracking objects in surveillance video. Increasingly there is also a demand to acquire such data from mobile platforms (say police cars) and deliver the intelligence so-gathered to a range of heterogenous, dispersed and often mobile devices. The vast quantity of data and the expense of cabling already encourages the distribution of the intelligence, putting the processing near the cameras, and only broadcasting the relatively low-bandwidth information of interest to a central repository.

Having started to process surveillance video with computers, a whole range of further possibilies soon emerges. Several cameras in a given installation will be able to view the same area or at least the events in the view of one camera will correllate with the events seen by another — at the very least in the form of the same people or vehicles being seen more than once. It is clear that a richer understanding of the goings-on in a surveillance site can be achieved by integrating the data from multiple cameras. With multiple cameras comes a better, three-dimensional understanding of the world. People and vehicles can be tracked not just for short, unrelated timespans, but continuously over the entire time they are in the field of view of the cameras, and activity models can also predict their behavior when out of sight.

Such multi-camera systems increase the need for ambient intelligence as local processing systems need to communicate between themselves, sharing data from their different viewpoints, and further refining their representations before communicating to consumers of their information.

In the remainder of this chapter, we describe a number of the technical challenges involved in building up this rich world view through networks of cameras, as implemented and projected in the IBM Smart Surveillance System. In section 4, we discuss the architectures for communication, followed by single camera object detection and tracking algorithms in Section 5.0. Section 6 describes object normalization for view-invariant reasoning about objects, and Section 7 describes multi-camera strategies for tracking. Section 8 describes work on active camera systems to acquire high-resolution data of objects tracked in static cameras. Finally Section 9 describes the delivery, storage and searching of the tracking data for secure, privacy-protecting, distributed access.

4. Architecture

The IBM Smart Surveillance system is a complete architecture for multi-camera, distributed surveillance video processing, comprising front-end image processing and camera control, local processing to integrate track information from nearby cameras, and a back-end database infrastructure that stores and redistributes the data. Client applications and browse stations access the processed data by issuing queries to the database system.

Figure 1.1 shows an overview of the complete system.

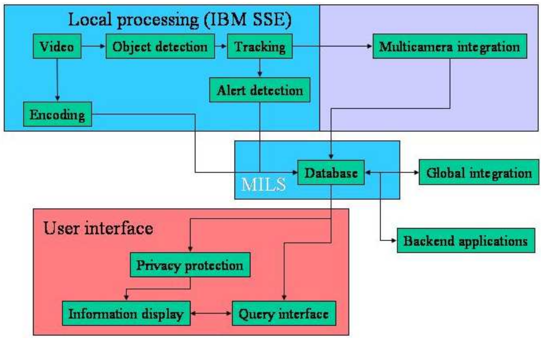


Figure 1.1. Architecture of the PeopleVision system, showing data flows between components.

Real-time video processing in the form of tracking and object detection (Section 5) is carried out close to the camera, on conventional computers or perhaps by embedded processors or specialised boards. Encoding (and encryption) of video is also carried out on, or as close as possible to, the camera, minimizing bandwidth and wiring requirements. Optionally a privacy camera [9] may be used that preprocesses the video and transmits only privacy-protected video.

Integration operations that exploit local knowledge are also carried out close to the cameras. Such operations include track correspondance between multiple cameras, whether overlapping or close, and active camera control that requires a reliable fast feedback loop. Such systems are described in Sections 7 and 8.

Data from all of these processes are communicated to a global data repository in the form of a conventional, DB2 database for concise numerical and string data and IBM ContentManager for rich media objects, in particular the compressed video streams. This back-end system may be centralized or distributed. These commercial data management products handle all the complex data management tasks (such as back-up, security, expiration, distribution, scalability and fast indexing) that are not specific to video surveillance data. Digital data distribution throughout the system can be encrypted, with video using conventional (e.g. MPEG4) video compression and a flexible XML schema for track information and meta-data.

Client applications that may be registered autonomous programs (e.g. elevator controllers, fire alarms etc.) or user-controlled browse stations access the database data through conventional means such as SQL queries. Queries can be formed on database-stored metrics such as colour, size, shape, movement, time, object class. Back-end data monitors observe the data in the database and can be used to add further inferred data, such as associating the recurrence of vehicles or people at different locations or times.

5. Tracking and object detection

The PeopleVision *Smart Surveillance Engine* (SSE) is an automated video surveillance system, constructed around algorithms for object detection and tracking. Using a modular architecture, we have experimented with a number of designs for each component. Object detection and tracking are described in more detail elsewhere [4] but we here give an overview of the principles of operation.

Object detection

The core method for object detection is background subtraction. This compares an automatically acquired reference image with incoming frames of video. Differences between the two images are areas of change in the image which are considered "objects". A sequence of more sophisticated processing attempts to refine the distinction between true objects and other differences in the images, caused by camera-shake, trees blowing in the wind, lighting changes and shadows.

We have experimented with a variety of object detection algorithms and have two principal approaches: adaptive multi-Gaussian and salient motion. The adaptive multi-Gaussian approach models each pixel as a mixture of Gaussians in colour space, following [10], with refinements that examine texture as a way of distinguishing shadows (that change lightness without changing the texture) from objects (that change texture as well as the colour of a pixel). Additional mechanisms are employed for actively "healing" stationary objects into the background. A preprocessor allows additional normalization mechanisms, for instance detecting and correcting for camera vibration, camera automatic gain controls and automatic white balance, as well as pixel noise correction (to remove camera noise and compression artefacts).

An alternative method of object detection employs salient motion detection to distinguish moving objects from motion in the background. Traditional background subtraction fails when there is motion in the "background" region of the image, for instance if there are trees blowing in the wind, flowing water or waves. The multi-Gaussian approach handles some of these situations, but the salient motion approach detects objects moving infront of more severe distracting motion by detecting consistency in the motion over a number of frames. Optic flow is carried out over the whole image, and motion vectors over successive frames are chained together. Regions of consistent motion over time are detected as moving objects.

Tracking

Tracking can be seen as a problem of assigning consistent identities to visible objects. Over time we obtain a number of observations of objects (detections by the background subtraction algorithm) but need to label these so that all observations of a given object are given the same label. When one object passes in front of another, partial or total occlusion takes place, with background subtraction detecting a single moving region. By occlusion handling, we hope to be able to segment this region, labelling each part appropriately, and correctly labelling the detected objects when they separate. In more complex scenes, occlusions between many objects must be dealt with.

When objects are widely separated a simple bounding box tracker is sufficient to associate a track identity with each foreground region. Bounding box tracking works by measuring the distance between each foreground region in the current frame and each object that was tracked in the previous frame, a match being declared if the object overlaps the region or lies very close.

If the foreground regions and tracks form a one-to-one mapping, then the tracking is complete and tracks are extended to include the regions in the new frame using this association. If a foreground region is not matched by any track, then a new track is created, and if a track matches no foreground region, it continues at a constant velocity, but is considered to have left the scene if it fails to match any region for a few frames.

Occasionally, a single track will be associated with two regions. For a few frames this is assumed to be a failure of background subtraction and both regions are associated with the track, but if there are consistently two or more foreground regions, then the track is split into two, to model such cases as when a group of people separate, a person leaves a vehicle, or an object is deposited by a person.

Appearance models

More complex interactions where more than one track is associated to one or more foreground regions are handled by a mechanism that uses an appearance model of each tracked object.

An appearance model consists of an image of the object — a two dimensional array of colour values with a mask indicating which pixels belong to the object. An appearance model is initialized by copying the foreground pixels of a new track. The appearance model can be correllated with detected foreground regions to track the motion of the centroid of an object being tracked by bounding box tracking. At each frame the appearance is updated by copying the current foreground pixels. During an occlusion, the foreground models of all the tracks in

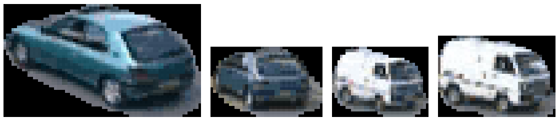


Figure 1.2. Appearance models from a PETS 2001video sequence, showing the appearance of model pixels, as one model recedes (left) and another approaches (right). Pixels not in the model appear black.

the occlusion are used to explain the pixels labelled as foreground by the background subtraction mechanism. We assume a depth ordering among the tracks and try to fit the models front-to-back, building up evidence in an explanation map. The position of each object is predicted with a velocity motion model, then the front-most is localized through correlation. Pixels that fall within the foreground mask of the object are entered into the explanation map as potentially being explained by the track. Subsequent objects are correlated with only those pixels in the foreground region which have no explanation so far, and are entered into the explanation map in their turn.

The explanation map is now used to update the appearance models of objects associated with each of the existing tracks. The depth ordering is recalculated by examining those pixels where two objects overlap. Models which account for these disputed pixels better are considered to lie in front of models which match the colour of the foreground less well. The initial depth ordering at the start of an occlusion is considered to be arbitrary since such occlusions generally occlude only a small fraction of the objects. Each model is only updated in those pixels where the model was the front-most object. Regions of foreground pixels that are not explained by existing tracks are candidates for new tracks.

Track data

With this inductive procedure, track records are created for each object during the period it is visible. Track data is trickled to the database for live visualization and real-time search. A series of post-processing operations can also be carried out to filter out some false alarms and other tracking errors.

In addition to object location and appearance, we also use a classification system to decide the type of object (e.g. person or vehicle) which is also stored in the database. Because of the variability in appearance of objects, classification relies on normalization as described in the next section.

6. Normalization

Normalization of image data is an important process in order to infer physical properties of objects in the scene measured in invariant units, such as meters or miles per hour. Even if only relative quanitities are required, physical properties of objects, such as their height or size, should be invariant to their location in the image. Measurements from image data must take into account the perspective distortion due to the projection of the world onto the image plane and other distortions such as lens distortion. In particular, for typical surveillance video with a far field view (i.e., the camera has its viewing direction nearly parallel to the ground plane), the farther aways an object lies, the smaller its projected image size will be. On the other hand, for an overhead camera looking down at a scene, a person standing more directly underneath the camera will appear foreshortened (Figure 6).

Investigators in digital video surveillance have recently begun to address this issue. Traditionally this has been done by semi-automatic calibration (relying on an expert) or rich geometric primitives in the image (such as parallel or orthogonal lines in the image). But realistic digital surveillance, which can be generally deployed, needs an automated solution.

Lv *et al.* [7] were the first to pioneer an effort to perform self-calibration of a camera from the tracking data obtained of a walking human. This method computes a 7 parameter transformation from 3D points in the world to 2D points in the image. The parameters are the focal length, the principal point (u_0, v_0), three rotation parameters, and the height of the camera from the ground plane. These are computed by finding three orthogonal vanishing points of the imaging system using every pair of observations of a human's projected height and location. With sufficiently high quality data, this method can be used to perform a full intrinsic and extrinsic calibration but in practice is somewhat unstable with realistic tracking data.

More recently, Bose and Grimson [3] proposed a method to perform ground plane rectification based on tracked objects moving at a constant velocity. The rectification can be either affine (making parallel world lines appear parallel in rectified image) or metric (making angles in the world plane equal to angles in the rectified image). This method assumes the ground is planar and it is possible to acquire tracks of objects moving at a constant velocity. In practice, these assumptions cannot always be satisfied. The ground is often not planar and it is difficult observe tracks of objects moving at a constant velocity, in particular for views of pedestrians only or complex intersections/roadways.

Stauffer *et al.* [11] present a method in which projected properties P_j of a particular track j, are modeled by a simple planar system such that the value of the property varies linearly with the distance from the horizon line:

$$P_j(t) = s_j(ax_j(t) + by_j(t) + c), \tag{1.1}$$

where t represents an instance in time or the frame of the measurement. For each track j, an individual scale factor parameter s_j and three global parameters of the planar model (a, b, c) are found as the best fit to the observations (x_j, y_j) for all j. This method is applied to all tracks regardless of the object type (vehicle, pedestrian, animal etc.) The limitation of this approach is that object properties such as height and width depend heavily on the viewpoint direction, particularly for vehicles whose length and width vary greatly. Although in theory, the change in the projected property should vary nearly linearly with distance, this also assumes a planar ground surface, no occlusion, and only perspective distortion.

We propose a method which does not rely on a planar ground surface, is not limited to certain camera viewpoint directions (far field), is not linear/planar, nor does it require objects moving at a constant velocity. Our system relies either on pedestrian data obtained from our classifier or input into the system. In the former case, the classifier is run over a few days, to obtain several sequences in which pedestrians traverse the space. The classifier determines if the track is a person, a vehicle or a group of people. In each case, a confidence measure is assigned to the classification result. Over several days, sequences classified as humans, whose confidence measures are relatively high are selected as input data to the normalization system. This typically finds sequences of pedestrian data without shadows, from decent imaging conditions (no precipitation or wind) and simple pedestrian shape and motions (not carrying objects, wearing hats, holding umbrellas, or performing odd behaviors.)

For each frame j, in the sequence, the position (x_j, y_j) of the foot of the pedestrian (based on the location of the bottom of the major axis of an ellipse which is fit to the data), the length, H, and orientation, θ, of the major axis are used. Normalization is performed by a least squares fitting of a second order polynomial to this data. For each property $p \in (H, \theta)$, we minimize the sum of squares:

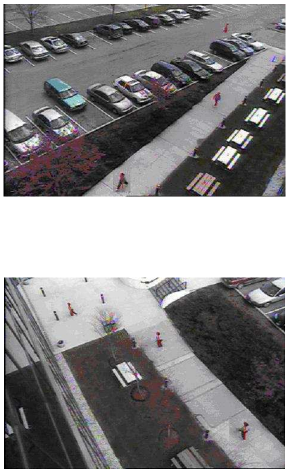
$$\min_{a_1,\ldots,a_6} \sum_j \left[p_j - p(x_j, y_j, a_1, \ldots, a_6)\right]^2 \tag{1.2}$$

where $a_1, \ldots, a_6$ are the coefficients of the polynomial. For each position in the image, we can predict the height and orientation of the projected image of a person (Figure 6).

From this information, we can also normalize any of a range of metrics used by the surveillance system. Normalized metrics include area, length, major/minor axis length, major axis angle, and velocity magnitude. For subsystems which rely on frame to frame alignment of the detected object, such as appearance-based tracking or recurrent motion estimation for the classifier, normalized metrics alleviate the need to scale to an initial segmentation and to estimate a re-scaling on a frame-to-frame basis. It is also now possible to distinguish if a projected view is getting larger for other reasons, such as change in 3D position.

Many subsystems of an automated digital surveillance system can benefit from this normalization. The frame-to-frame tracker can predict frame-to-frame size and orientation changes for appearance-based alignment, or even ignore objects which are smaller for their location than objects of interests for that location should be.

An object classifier can better distinguish physical objects based on properties which are invariant to image location such height, width or speed. For a multi-camera system with either overlapping or non-overlapping views, normalization information can be used to improve matching of object tracks. Lastly, forensic retrieval systems can now search based on absolute sizes and ignore the complex variations due to perspective distortion and the range of viewpoints across different cameras.



Size (pixels)	Orientation (degrees)
11	85
18	80
21	95

Size (pixels)	Orientation (degrees)
24	65
20	81
18	90

Figure 1.3. The top picture shows a typical surveillance far field view of a scene. A person appears smaller the farther away they are. The table to the right shows the change in size and orientation of a human at each of the three locations. The bottom picture is an overhead camera looking down at a scene. A person appears smallest when they are more directly underneath the camera. The table at right shows the change in size and orientation of the corresponding person in the scene. These size/orientation values are predicted for the given position based on prior data and can be used to normalize the live data at each position and across camera views.

7. Multi-camera coordination

Ambient intelligence becomes particularly valuable when a system includes multiple cameras. Integrating information between image processing systems that interpret the video locally requires local communication between intelligent devices. The representational power of such systems increases far beyond the abilities of a system that replicates independent video interpretation systems.

We can distinguish three kinds of intercamera relations according to the proximity of the cameras fields of view:

- overlapping,
- close,
- distant.

Any or all of these relations may be found in a multi-camera system. When cameras overlap, tracking systems can synchronously combine information to disambiguate tracking and derive richer models of objects seen from multiple viewpoints. As described in the next section, resolutions of the two cameras might be radically different, enabling significantly enhanced representations of tracked objects. Overlapping cameras also allow unambiguous continuous tracking of objects over extended regions far beyond the field of view of a single camera. [1]. Overlapping fields of view can be explicitly coded into a system with calibration, or can be learnt by watching the behaviour of tracks over some extended training period, and detecting correllations [11].

When two cameras have non-overlapping but close fields of view, similar continuous tracking can also be carried out, but this requires a degree of inference and a concomitant uncertainty. Training allows a system to learn the interconnection between cameras fields of view [5, 6] — learning when and where objects leaving one camera's field-of-view are likely to be detected in another's.

So far, in the peoplevision system we have only examined the case of cameras with overlapping fields of view. The cameras are calibrated with respect to one another using a homography (a linear transform from image coordinates in one view to image coordinates in another view, that applies to points lying on the ground plane). Objects are tracked using our conventional 2D tracking algorithms in each view, but using the homography, we can know when and where tracks in one view should be visible in another's. Matching tracks are then associated and given a common label, allowing continuous tracking over extended regions.

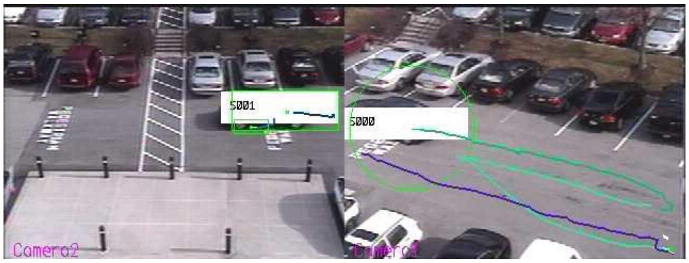


Figure 1.4. Overlapping views from two cameras. When the object in camera 2 (right) enters the field of view of camera 1 (left), the two tracks are registered and tagged as relating to the same object.

8. Multi-scale image aquisition

In many camera installations, resolution limits the capabilities of the system. Currently deployed surveillance system almost exclusively use analog video, and are often monochrome and stored on poor-quality, time-multiplexed analog video tapes. The blurred grey images of surveillance footage are familiar from television, seemingly never clear enough to identify a criminal. While quality is improving, and digital technologies promise higher quality and higher resolution, resolution will be limited for years to come. Supposing that a face recognition system requires 100 pixels across a face to perform recognition — to identify any face in a 100m wide space would require a 40 gigapixel static camera- far beyond current projections and data-handling capabilities. In practice, surveillance systems requiring high resolution images use the foveation principle seen in human vision — of directing a high-resolution sensor to areas of particular interest. Security guards steer pan-tilt-zoom (PTZ) cameras with a joystick, or use preset zoom positions to quickly examine points of regular interest.

As with all video surveillance though, the guards are fallible. They must be alert to detect an incident in the first place and then require great skill to track when a person must be tracked continuously across multiple cameras. (For instance, a continuous video record may be required to obtain a conviction for shoplifting.) There is the further problem that it becomes impossible to track more than one target at once. Faced with these problems, we have developed several multi-scale video acquisition systems that use automatically-controlled PTZ cameras to acquire close-up images of tracked objects.

Active Head Tracking and Face Catalogging

One such multi-scale acquisition system is the Active Head Tracker. This uses three or more calibrated cameras to acquire high resolution images of heads and faces. The system is conceived of as an image acquisition or preprocessing system for a number of human understanding systems, from face recognition to focus of attention determination and audio-visual speech recognition.

The system, shown in Figure 1.5 is based upon independent two-dimensional tracking in each of two static cameras. Typically the cameras are wide angle to achieve joint coverage of a wide operational area. Applying our tracking algorithms gives us the position of each independently moving object in each image. 3-dimensional calibration of the cameras allows triangulation of the two sets of object tracks to obtain both a correspondance between the objects in the two sets, and a 3D position for each object. To triangulate on a distinctive position, we triangulate the 2D centroid of the head in each view, which approximates to the projection of a 3D central point. The head is found by an algorithm that analyses the contour of the segmented object looking for an object part whose position and shape is consistent with being a head.

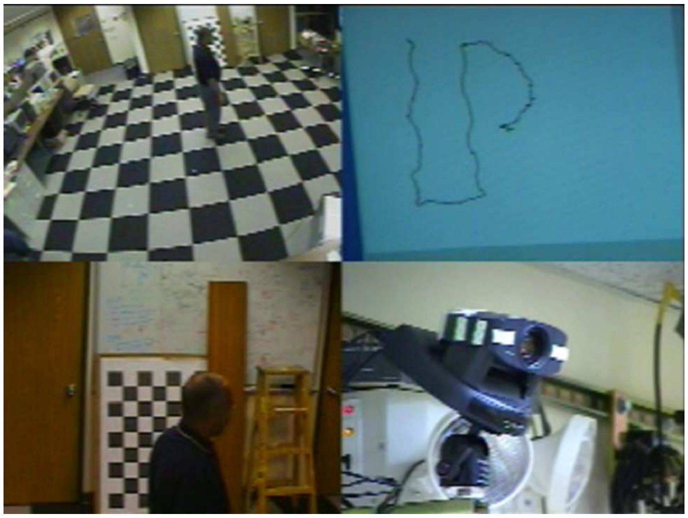


Figure 1.5. The active head tracker. Top left: a person seen by one of the fixed, wide-angle cameras. Bottom left: a close-up view captured by the PTZ camera (seen bottom right). Top right: the track of the person's head projected into the horizontal plane.

The 3D wide-baseline stereo triangulation gives a 3D position for absolute spatial indexing of object tracks and the disambiguation of occlusions that might present difficulties to a single camera, or even narrow-baseline stereo system. It also provides a target point for the active image acquisition system. A calibrated Pan-Tilt-Zoom camera is directed towards the target point and zoomed appropriately to capture the whole head area, taking into account both positioning errors from triangulation and the object speed — a wider zoom being necessary to ensure that faster objects are kept in the frame. The active head tracker provides a significant magnification compared to the static cameras.

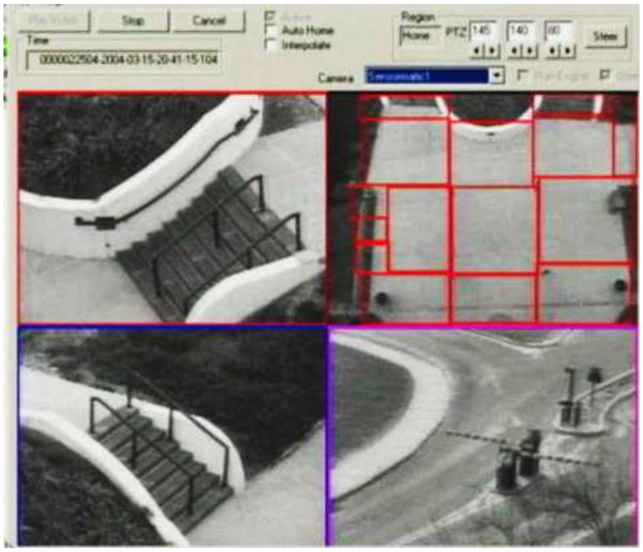
In a refinement of this open-loop active head tracking system, a face detection system is applied to the PTZ camera output. Should the imaged head be facing the camera, a face is detected and the camera is servoed in (with a negative feedback control loop now making the system independent of any calibration, segmentation and triangulation errors) until the face fills the video image. A still or video clip is recorded and stored in a database, for human or machine face recognition.

A further extension to the system is the development of sophisticated camera scheduling policies that control the assignment of multiple cameras among the multiple people being tracked by the system. The choice of camera policy is application-dependent. The addition of a multi-camera head pose estimation system [12] that operates at head resolutions as low as 8 × 8 pixels and thus can be applied in the wide-angle views, allow us to determine, in an absolute, world coordinate system the head orientation of the subjects allows the system to direct at each subject the camera most likely to see the subject's face.

Uncalibrated, multi-scale data acquisition

An alternative path that we have followed for the acquisition of multi-scale data is to use a single, uncalibrated camera to trigger foveation by one or more active cameras. In this system, a single fixed camera observes an overview, and an operator selects in the image a number of regions of interest for which high-resolution images are desired (Figure 8.0). For each of these regions, a separate PTZ camera is steered to zoom in on the area of interest, and the steering parameters are recorded and associated with the region of interest. For each available PTZ camera, a separate set of regions and zoom coordinates can be chosen.

After this simple training operation, the system runs autonomously, tracking targets in the 2D view of the static camera (as in Section 5 and steering the corresponding PTZ camera to the associated PTZ coordinates when any object enters one of the regions of interest. Again camera assignment becomes an important, but application dependent



Time: 0000022504-2004-03-15-20-41-15-104

Region: Home PTZ 145 140 60

Camera: Sensormatic1

Figure 1.6. Multi-Scale image acquisition. The top right pane shows the static, "master", camera's view point. The other panes show the views of the steerable "slave" cameras. Boxes in the master's view show regions of interest for which steering parameters of the other cameras have been set up.

aspect of the problem. A typical assignment policy might follow these rules:

- For a single object steer all cameras at the target all the time.
- Assign one camera to each of multiple objects according to which has the best view (for instance "best" might be interpreted as the camera with target area whose centre the object is closest to, or the camera which can be steered to point at the target in the shortest time). Assign additional cameras evenly to the targets of which they have the best views.
- After following an object for a given period of time, permute the camera assignments so that alternate views of a target are acquired (e.g. left and right sides).
- Steer all available cameras to certain designated targets, e.g. shoplifters.

Once a camera is in place looking at the intended target zone, captured images are taken immediately and periodically, and are sent with tracking information to the central database. The track browsing and query program allows the user to select a track and see all the zoomed-in images associated with that track.

Extensions

A refinement of the system allows it to operate with a single camera acting as both the master (fixed) and the slave (PTZ) camera. As above, the object detection and tracking are run on the video from the camera with a wide-angle view. Regions of interest are drawn in this field of view and associated with close-up PTZ parameters for the same camera examining the area of interest. When a tracked object enters a region of interest, tracking is suspended, the camera steers to the close-up viewpoint, acquires images for some fixed period (or until some condition is met, such as there being no motion in the close-up view), and then zooms out and recommences tracking. In this way, we have designed a system that autonomously captures licence plate images from every vehicle driving up to an entry barrier, while maintaining an overview of a wide area when no vehicle is at the barrier.

More complex scenarios can also be handled with the system, for instance having multiple master cameras that can each be slaved to acquire close-ups in each other's fields of view when there is no activity in their own.

Extensions to the system are being developed, including continuous control of the PTZ camera based on tracking in the fixed camera; and tracking within the moving PTZ camera view.

9. Indexing Surveillance Data

The vision and active camera control technologies described above process torrents of video data to extract streams of useful information. The information can be used to trigger real time alerts of events requiring human attention, but also provides a rich data stream for delivery to other devices, and for storage and post-hoc searching.

In our Middleware for Large Scale Surveillance (MILS) architecture, individual smart surveillance engines, or groups of collaborating smart surveillance engines that share information about common tracks, communicate tracking information to a database that may be centrally controlled or decentralized. In our implementation SSEs trickle live track data in XML data chunks to a DB2 database. The database aggregates track information and summarizes that information in track summaries that allow rapid searches over large quantities of surveillance data — from long periods of time and multiple cameras.

Searches can be on any of the data stored in the database, from track motion (position and speed at any time, or aggregated motion), to model appearance (size, colour, type) to aggregate queries that describe multiple attributes for a single track or combination queries involving multiple tracks. Larger scale queries can be statistical in nature or involve aggregation of data over long periods of time. Some examples of the queries that could be answered by such a database query engine are as follows:

- Show all blue cars travelling north-to-south this morning.
- Show the fastest vehicle track in a given time period.
- Show who abandoned this luggage and where they are now.
- Show the average speeed of people at a location.
- Show all tracks that came close to a particular person (handoff detection).

Visualization

Since the database contains the background appearance, and the appearance and motion of all objects, a highly compressed summary video can be rendered from the database contents, allowing rapid visualization of all incidents meeting search criteria, even over low-bandwidth networks. In parallel to tracking and database storage, a typical system will also encode and index a high fidelity digital video record that can be played back to visualize query results. We have created browsing applications that can deliver the summary information (e.g. to a hand held device) or the full video in response queries or browsing.

10. Privacy

Since video surveillance is a powerful tool with considerable privacy implications, we have also been investigating ways to protect privacy in a video surveillance system. The techniques we have developed centre on the idea of re-rendering video information according to the object oriented representation extracted by our video understanding system. Our ideas on video privacy are more fully explained in a separate paper [9].

The tracking, detection and classification of objects results in a separation of the video into independent streams for the background and each tracked object. Given this information, we can rerender the video manipulating each of these streams independently, for instance replacing each object by a solid rectangle that conveys the location, size and motion of an object without carrying any information about appearance and thus race, age, gender etc. Such rerendering can be tuned to the application in question and governed by access control lists and privacy policies that, for instance, allow security guards to override the obscuration, and permit law-enforcement officers access to raw, unchanged data.

Such a capability has been added into our MILS infrastructure, with tracked regions being blurred out in video playback for users with restricted priviledges.

11. Conclusions

In this chapter we have presented the IBM Smart Surveillance System that is a distributed system for the understanding of visual input from a network of cameras. The system is an exploration of a particular kind of ambient intelligent system with distributed sensors, local processing and delivery of resulting data to mobile users over wireless networks. The system extracts rich useful information that can drive real-time alarms or be searched after-the-fact as an index to stored video.

References

[1] J. Black and T. Ellis. Multi camera image tracking. In *International Workshop on Performance Evaluation of Tracking and Surveillance*, 2001.

[2] R.M. Bolle, J.H. Connell, S. Pankanti, N.K. Ratha, and A.W. Senior. *Guide to Biometrics: Selection and Use*. Springer-Verlag, New York, 2003.

[3] B. Bose and E. Grimson. Ground plane rectification by tracking moving objects. In *Joint IEEE Int'l Workshop on VS-PETS*, pages 9–16, October 2003.

[4] J. Connell, A.W. Senior, A. Hampapur, Y.-L. Tian, L. Brown, and S. Pankanti. Detection and tracking in the ibm peoplevision system. In *IEEE International Conference and Multimedia Expo*, 2004.

[5] T.J. Ellis, D. Makris, and J.K. Black. Learning a multi-camera topology. In J. Ferryman, editor, *PETS/Visual Surveillance*, pages 165–171. IEEE, October 2003.

[6] S. Khan and M. Shah. Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(10), October 2003.

[7] F. Lv, T. Zhao, and R. Nevatia. Self-calibration of a camera from video of a walking human. In *Proc. International Conference on Pattern Recognition*, pages 562–7, August 2002.

[8] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A.W. Senior. Recent advances in the automatic recognition of audiovisual speech. *Proceedings of the IEEE*, 2003.

[9] A.W. Senior, S. Pankanti, A. Hampapur, L. Brown, Y.-L. Tian, and A. Ekin. Blinkering surveillance: Enabling video privacy through computer vision. *IEEE Security and Privacy*, 2004.

[10] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Fort Collins, CO, June 23-25*, pages 246–252, 1999.

[11] C. Stauffer, K. Tieu, and L. Lee. Robust automated planar normalization of tracking data. In *Joint IEEE Int'l Workshop on VS-PETS*, October 2003.

[12] Y.-L. Tian, L. Brown, J. Connell, S. Pankanti, A. Hampapur, A. Senior, and R. Bolle. Absolute head pose estimation from overhead wide-angle cameras. In *Intl. Workshop on Analysis ad Modelling of Face and Gesture*, October 2003.