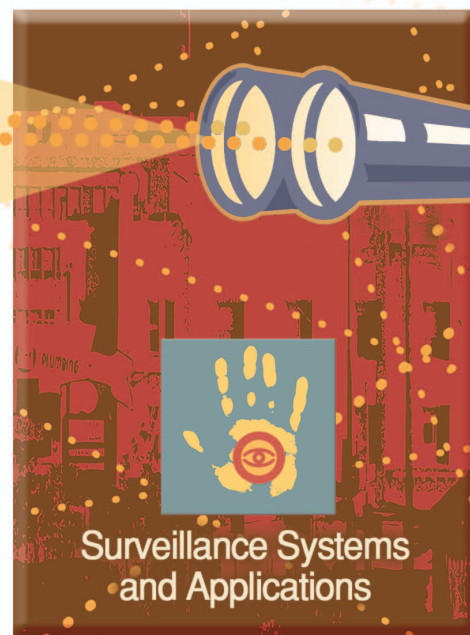


©2008 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE

Gian Luca Foresti, Christian Micheloni,  
Lauro Snidaro, Paolo Remagnino, and Tim Ellis



© EYEWIRE

# Active Video-Based Surveillance System

[The low-level image and video processing techniques needed for implementation]

**T**he importance of video surveillance techniques [3], [9], [25] has increased considerably since the latest terrorist incidents. Safety and security have become critical in many public areas, and there is a specific need to enable human operators to remotely monitor activity across large environments such as: a) transport systems (railway transportation, airports, urban and motorway road networks, and maritime transportation), b) banks, shopping malls, car parks, and public buildings, c) industrial environments, and d) government establishments (military bases, prisons, strategic infrastructures, radar centers, and hospitals).

Modern video-based surveillance systems (see classifications described in [9]) employ real-time image analysis techniques for efficient image transmission, color image analysis, event-based attention focusing, and model-based sequence understanding. Moreover, cheaper and faster computing hardware combined with efficient and versatile sensors create complex system architectures; this is a contributing factor to the increasingly widespread deployment of multicamera systems.

These multicamera systems can provide surveillance coverage across a wide area, ensuring object visibility over a large range of depths. They can also be employed to disambiguate occlusions. Techniques that address handover between cameras (in configurations with shared or disjoint views) are therefore becoming

increasingly more important. Events of interest (identified as moving objects and people) must be then coordinated in the multiview system, and events deemed of special interest must be tracked throughout the scene (see Figure 1). Wherever possible, tracked events should be classified and their dynamics (sometimes called behavior) analyzed to alert an operator or authority of a potential danger. For security awareness based on multiscale spatio-temporal tracking, see Hampampur et al. [10].

In the development of advanced visual-based surveillance systems, a number of key issues critical to successful operation must be addressed. The necessity of working with complex scenes characterized by high variability requires the use of specific and sophisticated algorithms for video acquisition, camera calibration, noise filtering, and motion detection that are able to learn and adapt to changing scene, lighting, and weather conditions. Working with scenes characterized by poor structure requires the use of robust pattern recognition and statistical methods. The use of clusters of fixed cameras, usually grouped in areas of interest but also scattered across the entire scene, requires automatic methods of compensating for chromatic range differences, synchronization of acquired data (for overlapping and nonoverlapping views), estimation of correspondences between and among overlapping views, and registration with local Cartesian reference frames.

This article describes the low-level image and video processing techniques needed to implement a modern visual-based surveillance system. In particular, change detection methods for

both fixed and mobile cameras (pan and tilt) will be introduced and the registration methods for multicamera systems with overlapping and nonoverlapping views will be discussed.

**MULTICAMERA SYSTEMS CAN PROVIDE SURVEILLANCE COVERAGE ACROSS A WIDE AREA, ENSURING OBJECT VISIBILITY OVER A LARGE RANGE OF DEPTHS, AND CAN BE EMPLOYED TO DISAMBIGUATE OCCLUSIONS.**

#### FROM IMAGES TO EVENT REGISTRATION

A visual-surveillance system is comprised of a network of sensors (typically conventional closed circuit (CCTV) cameras), some with overlapping

fields of view, providing continuous (24/7) online operation.

Each visual surveillance network has its own specific architecture. For fixed cameras, the architecture is very much data-driven and its data-to-information flow is bottom-up. As mobile cameras are employed in more sophisticated networks, one might envisage a number of feedback controls to tune camera parameters (for instance, to adapt to weather or illumination conditions) or to track events of interest. Figure 1 illustrates the main processing tasks for a visual system:

- camera calibration with respect to an extrinsic Cartesian reference frame
- scene acquisition
- adaptive modeling of background
- change detection for foreground regions/blobs identification
- multicamera registration.

All of the steps are intertwined: camera calibration and registration can be learned from observation data, and the processes used to achieve this automatically require basic image processing, such as the modeling of background views and the detection of foreground events. *Change detection* and *camera registration* have been chosen as the two basic steps, and they will be described in detail in the next two sections.

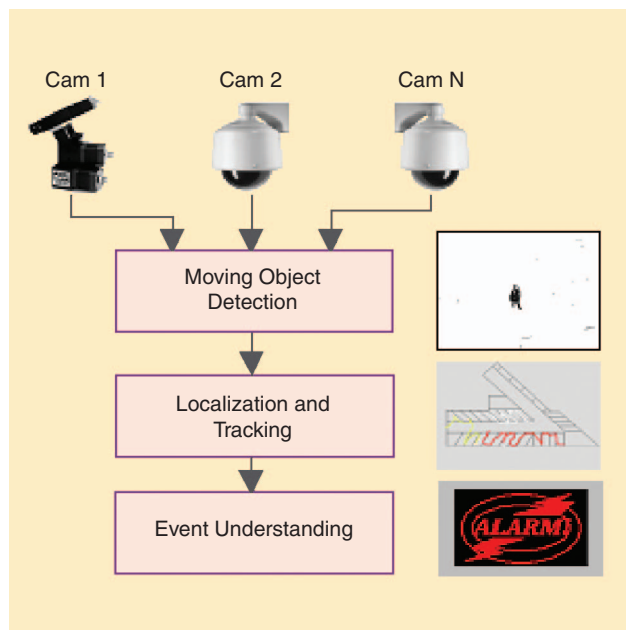
#### CHANGE DETECTION METHODS FOR FIXED CAMERAS

A variety of change detection methods have been developed for fixed cameras (see Figure 2). Exhaustive reviews can be found in [9]. Here, the main methods used in visual-based surveillance systems are detailed.

The simple difference (SD) method (see Figure 3) computes for each time instant  $t$  the absolute difference  $D_t(x, y)$  between the pixel intensities of the input image pair; it then applies a threshold ( $Th$ ) to obtain a binary image  $B(x, y)$ . Threshold selection is a critical task, and the methods proposed by Kapur [14], Otsu [19], Ridler [27], Rosin [29], and Snidaro and Foresti [33] illustrate the variety of approaches that have been employed.

The SD method is the simplest and fastest, but it is very sensitive to noise and illumination changes; this directly affects the gray level recorded in the scene, which could be incorrectly interpreted as structural changes.

To overcome the problem of noise sensitivity, the derivative model (DM) method considers  $n \times n$  pixel regions in the two



**[FIG1]** A general architecture of an advanced visual surveillance system.

input images and computes a likelihood ratio  $L_{ij}$  by using the means and the variances of the two regions  $R_i$  and  $R_j$ .

The output binary image is obtained as

$$B(x, y) = \begin{cases} 0 & \text{if } L_{ij} < L_{Th} \\ 1 & \text{otherwise} \end{cases} \quad \forall (x, y) \in R_i, R_j \quad (1)$$

where  $L_{Th}$  is the threshold.

The shading method (SM) models the intensity at a given point  $I_p(x, y)$  as the product of the illumination  $I_i(x, y)$  and a shading coefficient  $S_p$  which is calculated for each point according to Phong's illumination model. It can be easily proved that, to establish whether a change has taken place in a given region  $R_i$  over two consecutive frames,  $I_{t-1}(x, y)$  and  $I_t(x, y)$ , it is sufficient to calculate the variance  $\sigma_i$  of the intensity ratios  $I_t/I_{t-1}$  in that region. If  $\sigma_i$  is close to zero, no change has taken place.

The DM and SM methods yield similar results. The noise level affecting the output image is lower than that generated by the SD method; however, the accuracy of the detected blobs, in terms of shape, position, and size, is lower. In particular, object contours are significantly altered and the original object shape is partially lost.

The LIG method [22] is based on the assumption that pixels at locations having a high gray-level gradient form a part of an object and that nearly all pixels with similar gray levels will be also part of the same object. The intensity gradient is computed as

$$G(x, y) = \min \{I(x, y) - I(x \pm 1, y \pm 1)\}. \quad (2)$$

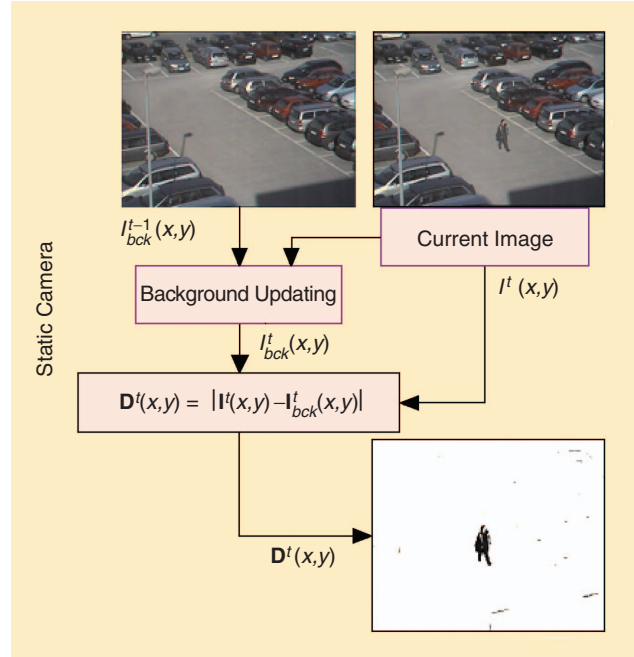
Thereafter, the  $G(x, y)$  image is divided into  $m \times m$  subimages in order to limit the effects of illumination change on the computation of local means and deviations. The regional means and deviations are first smoothed using the neighboring regions and then interpolated to refill a  $m \times m$  region. Finally, a threshold procedure is applied to isolate object pixels from the background. The LIG method gives satisfactory results, even if is not sufficient to completely discriminate the object from the background.

### BACKGROUND UPDATING

To minimize errors in the background change detection process due to noise effects, illumination, and/or environmental changes, advanced visual-based surveillance systems apply background updating procedures [8]. A classical approach is based on the Kalman filter (see Figure 4). Background updating procedures attempt to determine significant changes using an estimate of the background scene (see an example in Figure 5).

In Figure 4, the filter is applied to each pixel  $(x, y)$  of the input image to adaptively predict an estimate of the related background pixel at each time instant. Figure 3 illustrates the data flow diagram for a Kalman-based background updating module [8].

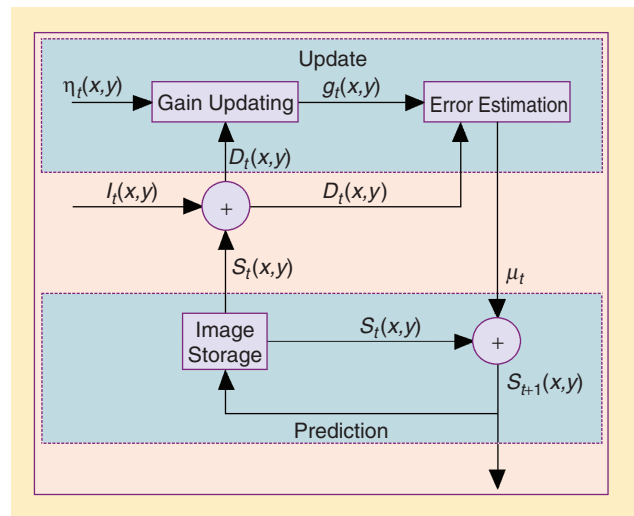
The dynamic system model is represented by



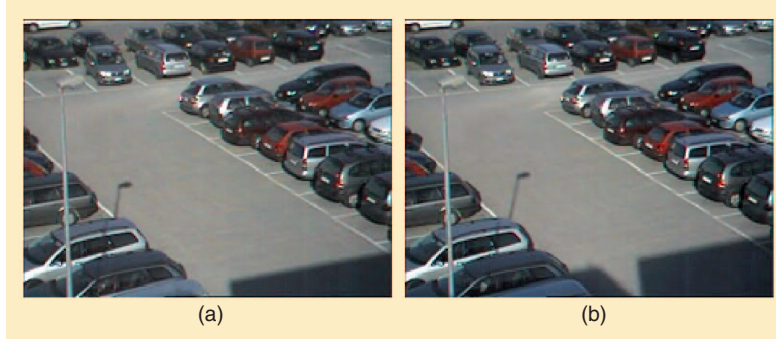
[FIG2] A flow chart showing the main tasks involved in the computation of the image of differences in context of static camera.

$$S_{t+1}(x, y) = S_t(x, y) + \mu_t \quad (3)$$

where  $S_t(x, y)$  represents the gray level of the background image point  $(x, y)$  at the instant time  $t$ ;  $S_{t+1}(x, y)$  represents an estimate of the same quantity at  $t + 1$  and  $\mu_t$  is an estimate of the system model error. Such an error takes into account the system model approximation and is formed by two components (i.e.,  $\mu_t = \beta_t + v_t$ , where  $\beta_t$  represents a slow variation (with non-zero mean and a temporal range comparable to the filter response time) and  $v_t$  represents a white noise with zero mean). The model for  $\beta_t$  is a random walk model,



[FIG3] General scheme of the Kalman-based background updating module.



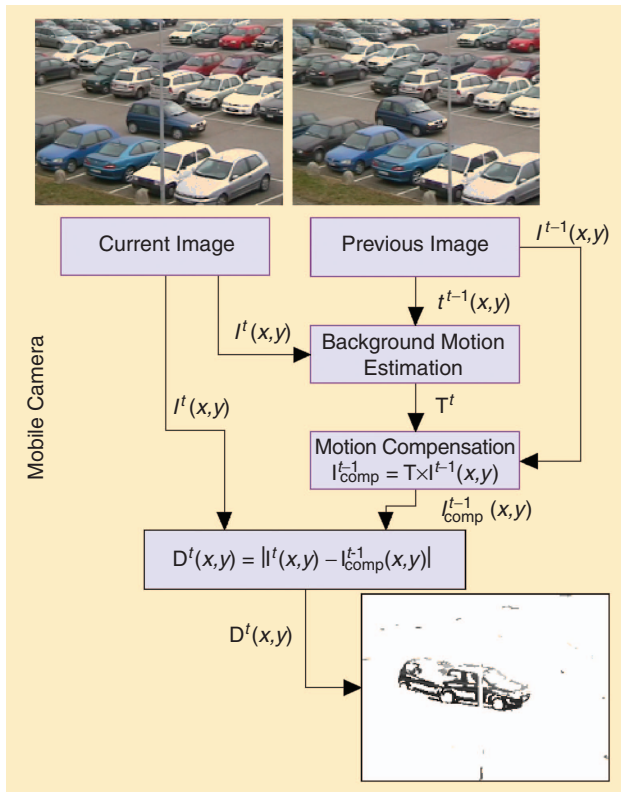
**[FIG4]** Example of background updating. (a) has been acquired at start time, while (b) represents the background image after two hours. It is worth noting how the building shadow has been associated with the background.

$\beta_{t+1} = \beta_t + \varepsilon_t$  with initial condition  $\beta_0 = 0$ , where  $\varepsilon_t$  is white noise with zero mean.

The measurement model is represented by

$$I_t(x, y) = S_t(x, y) + \eta_t(x, y) \quad (4)$$

where  $I_t(x, y)$  represents the gray-level of the current image point  $(x, y)$  and  $\eta_t(x, y)$  is the measurement noise (an esti-



**[FIG5]** A flow chart with the main tasks involved in the computation of the image of differences in the context of a mobile camera. It is worth noting how the motion compensation is employed to negate the camera motion.

mate of the noise affecting the input image). This noise is assumed to be Gaussian with zero mean and variance  $\sigma^2$ . The recursive equations of the Kalman filter are applied to obtain the optimum linear estimate of the background image  $S_t(x, y)$  based on the observations (system measures)  $\{I_t(x, y) : 0 \leq t \leq k\}$  and initial conditions  $S_0(x, y) = I_0(x, y)$ .

The updating module uses the gray-level  $D_t(x, y)$  of the difference image point  $(x, y)$  and the estimate  $\eta_t(x, y)$  of the noise affecting the input image to update the filter gain  $g_t(x, y)$ . Then, it computes an estimate of the system model error  $\varepsilon_t$  on the basis of the filter gain  $g_t(x, y)$  and the  $D_t(x, y)$  value (i.e.,  $\mu_t = D_t(x, y) \cdot g_t(x, y)$ ). The prediction module computes an estimate of the gray-level  $S_t(x, y)$  of the background image point  $(x, y)$  at the next frame. The image storage block simulates the delay between two successive image acquisitions.

A multilayered background model has been used in the CMU surveillance system [5], which employs a combination of temporal differences and template matching to perform object tracking. Each time an object enters the scene and remains stationary for a predefined amount of time, it is considered a new layer and is added to the background (i.e., a parked car). This adopted multilayer solution was effective in solving typical problems of background-based change detection approaches, such as “holes” or “ghosts” created by objects in the background that began to move (such as a car leaving the area).

None of the methods described above address the problem of multimodal backgrounds that are typical of outdoor scenes, where the pixels can switch state (e.g., due to trees waving in the wind). For these conditions, a single Gaussian model per pixel cannot be assumed. Stauffer and Grimson [34] modeled the recent history of each pixel,  $\{x_1, \dots, x_t\}$ , as a mixture of  $k$  Gaussian distributions. The probability of observing the current pixel value is given by

$$P(x_t) = \sum_{i=1}^k \omega_{i,t} * \eta \left( x_t, \mu_{i,t}, \sum_{i,t} \right) \quad (5)$$

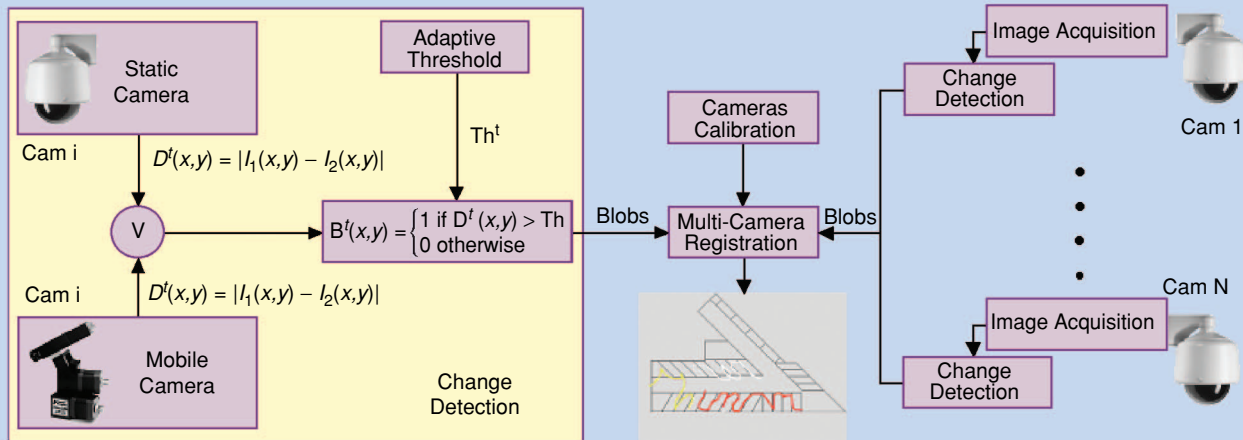
where  $k$  is the number of distributions,  $\omega_{i,t}$ ,  $\mu_{i,t}$ ,  $\sum_{i,t}$  are an estimate of the weight, the mean, and the covariance matrix of the  $i$ th Gaussian of the mixture at time  $t$ , and  $*$  is the convolution operator.  $\eta$  is a Gaussian probability density function:

$$\eta \left( x_t, \mu, \sum \right) = \frac{1}{(2\pi)^{n/2} |\sum|^{n/2}} e^{-1/2(x_t - \mu)^T \sum^{-1} (x_t - \mu)}. \quad (6)$$

The parameter  $k$  is, in part, determined by the available memory and computational power; a value in the range of 3–5 is suggested by Stauffer and Grimson [34].

## CHANGE DETECTION METHODS

Camera registration techniques require information about the position of objects in the image plane and the calibration of the sensor to correspond image pixels to points on a 2-D ground-plane map. The registration process can benefit from the use of multicamera techniques. In this context, the detection of mobile objects (based on the computation of the position on the image plane) is referred to as change detection (CD) and can be performed by using image differencing techniques (see the figure below).



Low-level part of the general architecture of a visual surveillance system. On the left the change detection module is deeply described for a generic  $i$ th camera which could be either static or mobile.

CD is commonly implemented at either the pixel [28], edge, or more complex feature level (such as lines and corners) [32]. Real-time image and video processing methods require low-computational cost; therefore CD algorithms using complex features are not generally employed. Frame-differencing CD methods that have been applied in the context of visual-based surveillance systems can be categorized into two principal types: a) frame by frame and b) frame to background.

CD methods compare two digitized images,  $I_1(x, y)$  and  $I_2(x, y)$ , and generate a binary image  $B(x, y)$  that identifies image sub-areas (called blobs) with significant differences between the two input images. Frame by frame methods consider two successive images of the sequence,  $I_{t-1}(x, y)$  and  $I_t(x, y)$ , while frame to background methods use the current image  $I_t(x, y)$  and a reference one (called the *background*),  $I_{bck}(x, y)$ , which represents the monitored scene without moving objects.

In a typical image sequence representing an outdoor scene, a blob does not always correspond to a single object because of the presence of shadows, light reflections, noise, and partial occlusion [8]. A background image, acquired in absence of moving objects, must be updated continuously to account for both slow and abrupt changes [8].

Every new pixel value is checked against the existing  $k$  distributions and, if the value is within 2.5 standard deviations of a distribution, then the pixel matches that distribution.

### CHANGE DETECTION METHODS FOR MOBILE CAMERAS

Since the camera movement induces an apparent motion in all of the image pixels, the application of standard CD techniques results in poor motion detection and therefore cannot be used. To overcome this problem, various methods have been developed based on the alignment of the images involved in the CD process (see Figure 6).

These methods, also known as image registration techniques, are based on the hypothesis that the changes between two consecutive frames can be approximated with a particular motion model. Three main models are generally used in this context of visual-based surveillance systems: translation, affine, and projective.










The purpose of these techniques is to compute the parameters that best approximate the motion of corresponding pixels

belonging to two consecutive frames,  $I_{t-1}(x, y)$  and  $I_t(x, y)$ . Once the parameters have been estimated, the transformation corresponding to the selected motion model is applied to the previous frame  $I_{t-1}(x, y)$ . This process enables the effects of the ego-motion to be removed before applying the traditional change detection operation.

To compute the parameters of the motion model, three principal directions have been investigated in the literature: a) techniques based on explicit knowledge of the camera motion, b) computation of the optical flow, and c) direct methods.

In [18], Murray and Basu proposed a method to compute the transformation parameters based on explicit information. By means of the data related to the rotation of the camera in pan and tilt angles, they are able to estimate the position of each pixel in the previous frame. They are then able to apply a frame-by-frame CD operation whose output is a set of moving edges, or edges belonging to moving objects.

**[TABLE 1] A COMPARISON, BASED ON THE PERCENTAGE CORRECT CLASSIFICATION (PCC) AND JACCARD METRICS [30], BETWEEN CD THRESHOLDING METHODS. ON THE LEFT, RESULTS FOR A STATIC IR CAMERA ARE SHOWN. ON THE RIGHT, THE MEAN BARYCENTRE ERROR (MBE) HAS BEEN CONSIDERED TO EVALUATE THE PERFORMANCES OF MURRAY [18] AND ARAKI [1] METHODS FOR IMAGE ALIGNMENT IN THE CONTEXT OF THE MOBILE CAMERA.**

								
<b>Static Camera</b>	<b>Ridler [27]</b>	<b>Otsu [19]</b>	<b>Kapur [14]</b>	<b>Rosin [29]</b>	<b>Fen [33]</b>	<b>Mobile Camera</b>	<b>Murray [18]</b>	<b>Araki [1]</b>
PCC	0.4822	0.6243	0.9465	0.6751	0.9697	MBE	11.119	6.109
JACCARD	0.0008	0.0315	0.3823	0.0421	0.6248			

When explicit information about the camera motion parameters is not available, the image alignment can be performed by using optical flow. In [1], Araki et al. proposed a background compensation method based on the estimation of the background motion. This is achieved by tracking feature points on the background and estimating the parameters of an affine transformation from the previous frame to the current frame. The employment of optical flow in advanced visual-based surveillance systems requires the solution of several problems in order to reduce the computational complexity or provide effective methods to reject badly tracked features. To overcome these problems, direct methods [12] have been considered because they allow the estimation of the transformation parameters for image alignment without computing the optical flow (see Table 1).

In [12], Irani et al. estimate the ego-motion by searching for regions that can be approximated as planar and computing the displacement between consecutive frames with a direct method. They address the problem of moving object detection in multiplanar scenes by estimating a *dominant* eight-parameter trans-

formation. In particular, the sum of squared differences (SSD) error measure is minimized:

$$E^{(t)}(\mathbf{q}) = \sum_{(x,y) \in R} (uI_x + vI_y + I_t)^2 \quad (7)$$

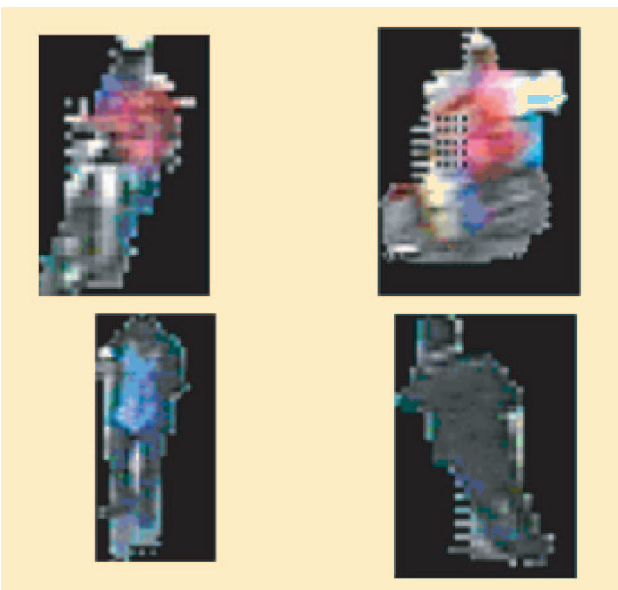
where  $I_x, I_y, I_t$  are the partial derivatives;  $(u, v)$  is the two-dimensional (2-D) motion field;  $R$  is a planar image region; and  $\mathbf{q} = (a, b, c, d, e, f, g, h)$  represents the vector of unknown parameters of the transformation. Then, by registering two consecutive frames according to the computed transformation, the rotational component of the camera motion can be cancelled. Furthermore, the *focus of expansion* (FOE) can be computed from the purely translational flow field in order to estimate the camera translation given its calibration information. Finally, given the 2-D motion parameters and the three-dimensional (3-D) translation parameters  $(T_x, T_y, T_z)$ , the 3-D rotation parameters of the camera are obtained by solving the following system:

$$\begin{aligned} a &= -f_c \alpha T_X - f_c \Omega_Y & e &= -\Omega_Z - f_c \beta T_Y \\ b &= \alpha T_Z - f_c \beta T_X & f &= \alpha T_Z - f_c \gamma T_Y \\ c &= \Omega_Z - f_c \gamma T_X & g &= -\frac{\Omega_Y}{f_c} + \beta T_Z \\ d &= -f_c \alpha T_Y + f_c \Omega_X & h &= -\frac{\Omega_X}{f_c} + \gamma T_Z \end{aligned} \quad (8)$$

which has only six unknowns.

#### PERFORMANCE MEASURES FOR CHANGE DETECTION METHODS

The optimal tuning of all visual processes included in an advanced surveillance system is a complex problem [25]. Receiver operating characteristic (ROC) curves have been used as the basis to evaluate the performance of a system and allow an automatic or quasi-automatic tuning. An ROC curve is generated by computing pairs  $(P_d, P_f)$ , where  $P_d$  is the probability of correct detection and  $P_f$  is the false-alarm probability. Both probabilities depend on the values of the parameters regulating



**[FIG 6] Differences in appearance between views.**

the behavior of a decision module of the system. The global performance curve summarizing the curves obtained under different working conditions is found by imposing an operating condition (e.g., equal bias,  $P_f = 1 - P_d$ ) and by plotting the corresponding values of  $P_e = (1 - P_d + P_f)/2$  against different values of the variable of interest.

Since most video-based surveillance systems aim at *detecting* objects, people, dangerous events, etc., ROC curves represent a good tool for evaluating the system performance after a previous characterization of the subject to be detected.

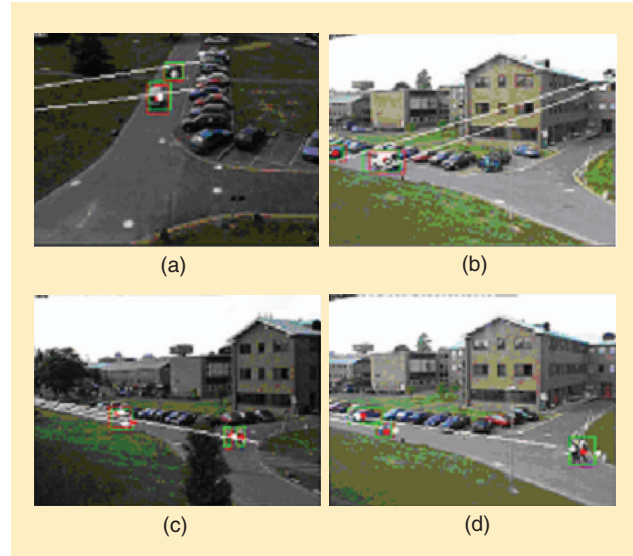
Object detection metrics can be defined in at least two ways. The first determines the segmentation quality, but requires reliable ground truth on the pixels that comprise the object. This information is notoriously difficult to reliably determine for real data. The second method simply identifies the presence of an object using some representative points (typically the blob centroid or bounding box) that are then used to perform tracking. In this case, ground truth can be more easily generated manually using a mouse pointer to locate objects in the scene. Once the ground truth positions are acquired, a trajectory distance metric (such as described by Black et al. [2]) can be adopted to assign computer observations to ground truth. See Table 1 for a performance comparison.

### MULTICAMERA VIEW REGISTRATION

For many tasks, a coherent interpretation of a complex scenario can only be maintained if events (people and other objects of interest) in the scene are identified and tracked correctly. For a single camera, tracking requires correspondence to be determined between pairs of observations separated over time (as in consecutive video frames). In a multicamera environment, if the target is detectable in more than one camera view field, correspondence can be established in a common coordinate frame (e.g., the ground plane) to locate the same target in the different views. If the cameras are not synchronized, then temporal correspondence must also be estimated.

Many techniques exist to combine data and extract useful information from multicamera systems [20]. Consistency across multiple camera views can only be maintained once correspondence between cameras is established; then an enhanced 2-D (usually called 2-D<sup>1/2</sup>) or 3-D rendering of the scene can be generated. Techniques used for monitoring wide spaces make use correspondence between views, combined with some a priori knowledge of the network topology and the environment.

Establishing correspondence is complicated for a number of reasons. First, the targets may not maintain a consistent appearance between views or over time due to changes in the pose, nonuniformity of target appearance, or the location of illumination sources (see Figure 7). In addition, the measurement of target appearance may require accurate calibration of the cameras in order to generate comparable characterizations. Secondly, a predictive model may not accurately represent the target's possible range of motions; whilst a simple linear predictor (for instance,  $x_{t+1} = \alpha x_t + \beta$ ) is subject to error if the motion is



**[FIG7]** (a), (b) Epipolar lines from two cameras with a wide baseline. The white circles indicate the centroid of each object. The red circles represent the object centroid projected by the homography transformation from the other camera view. (c), (d) Epipolar lines from two cameras with a narrower baseline. (Images are taken from the PETS2001 data sequences.)

nonlinear, the use of a nonlinear motion model [generalized as  $x_{t+1} = f(x_t; k)$ ] can create significant uncertainties if the target is undetected over one or more frames.

Failure to detect the target may be a result of segmentation failure or may occur because the target disappears from the camera view field as a result of occlusion. Thirdly, target location is initially measured in pixel coordinates, which are unique to each camera view; some method is required to bring these into correspondence within the regions of overlapped view fields.

While most surveillance systems comprise multiple cameras, these are normally arranged to maximize the coverage of the environment being surveyed. As a consequence, the surveyed environment typically contains a minimum of overlapping view fields, and the tracking of targets through the environment requires correspondence between nonoverlapping (disjointed) camera views.

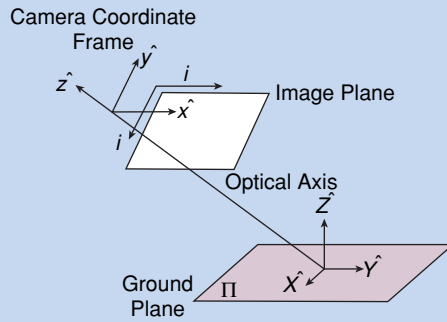
In comparison with the spatial registration afforded by geometric camera calibration for determining overlapping view fields, nonoverlapping view field registration methods are principally based on temporal cross-correlation of object disappearance and reappearance.

However, if the interstitial region between cameras is treated as an occlusion region, then it is possible to track targets as they transit between the two views using a predictive filter (such as Kalman) [2] provided that the two cameras are calibrated to a common coordinate system. As with many other occlusion-reasoning methods, reliability in determining correspondence is inversely proportional to the duration of the occlusion; through wide occlusion regions, the tracker is more likely to lose track.

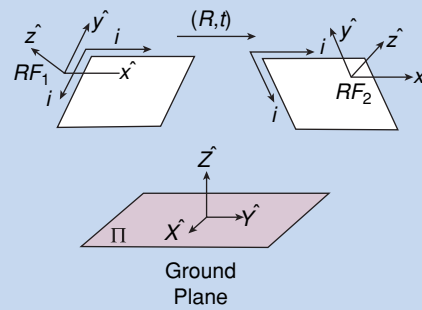


## MULTICAMERA GEOMETRY

Camera geometry is uniquely defined by its intrinsic and extrinsic parameters: i) intrinsics: focal length, pixel resolution, distortion (refer to [7] for details), ii) extrinsics: position of the optical center of the camera and the orientation of its Cartesian frame with respect to an extrinsic reference frame, for instance instantiated on a plane (i.e., the ground plane) [see (a) below].



(a) Single camera geometry.



(b) Multicamera geometry.

In multicamera systems [see (a) and (b) above], more external reference frames are established, defining locales and grouping a limited number of cameras and a semantic location in space. It would be impractical to have a single external coordinate frame for a wide area; also, it is more robust to perform geometric calculations in a locality. The use of locales is also important for a quick semantic annotation of scene dynamics. A point in the reference frame  $RF_2$  can always be transformed into a point in the reference frame  $RF_1$  by means of a rotation matrix  $R$  and a translation  $t$ :

$$P^{RF_1} = R P^{RF_2} + t \quad (1)$$

The rotation  $R$  is between the two reference frames, and the translation is the displacement vector between the two reference frames. The relationship between a point in the camera  $(X, Y, Z)$  frame and the image frame  $(x, y)$  is therefore defined by:

$$\begin{pmatrix} x \\ y \end{pmatrix} = f \begin{pmatrix} X/Z \\ Y/Z \end{pmatrix} \quad (2)$$

## REGISTERING OVERLAPPING VIEWS

Camera calibration for overlapping view fields requires a sufficient number of correspondences in order to establish a geometric transformation between views. Three-dimensional constraints (for instance, the ground plane) can then be applied to allow back-projection of information into a map representation of the scene and to perform spatial and temporal reasoning in a consistent manner.

A common constraint in surveillance is based on planar motion, resulting from the assumption of a piece-wise linear world where the scene is assumed to be formed by facets or locally planar surfaces. The planar motion constraint can be used to estimate the

homography between views; if more than two views are available to the multicamera system, then an approximate Euclidean reconstruction of the scene can be performed. Reconstructing a 3-D model in surveillance usually means

generating a map (effectively a 2-D top view of the scene—see Figure 9) where events occur. The map can then be employed to add semantic detail to the scene.

The mathematics involved for a multiview system under planar motion can be found in

Lee et al. [16] who based their method on the approach of Tsai [36] for the motion of a planar patch.

Correspondences between views are usually established by matching the appearance of interest points [35], regions, or iden-

**MODERN VISUAL SURVEILLANCE SYSTEMS DEPLOY MULTICAMERA CLUSTERS OPERATING IN REAL TIME WITH EMBEDDED SOPHISTICATED AND ADAPTIVE ALGORITHMS.**

tified blobs. Standard matching techniques follow template schemes, where a template from one view is compared with an area of interest in another view. Matching utilizes criteria based on the appearance of the template and the matched area of interest. Appearance criteria either make use of texture or color information [30]. Templates employed in matching can also vary in shape, and matching takes place using either pixel information or statistical measures drawn from the analyzed area of interest based on corners, edges, simple chromatic (histograms, mixture models, or robust statistics), combined chromatic (cooccurrence of colors) [4], or more sophisticated approaches, such as a wavelet-based representation of the underlying object template [21]. Features, such as points of interest (corners, for instance), can also be employed to impose additional geometric constraints. The epipoles can be estimated [7], and they indeed impose stricter geometric constraints or a simplified constraint imposes the projection of a field of view into the image plane of other cameras [20]. Epipolar geometry is usually imposed to constrain template matching between views (see Figure 7). Matches can be employed to refine the estimate of the homography or to establish online correspondence and fuse information from blobs or bounding boxes extracted after the change detection.

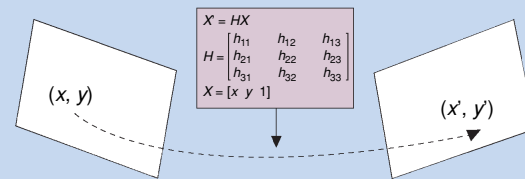
Traditionally, robust techniques to estimate epipolar geometry exist for a short baseline (e.g., binocular stereo); more recently, these have been extended to deal with wider baselines [24]. In [17] the epipolar constraint is used for region-based stereo, where a limited number of epipoles are selected and divided into segments belonging to classified objects. Corresponding segments are then back-projected onto the epipolar plane, and the center of the quadrilateral constructed by correspondences is considered to belong to a matched object.

A semi-automated calibration of overlapping views was proposed in [13]. First, a mapping between each view and a plane (where motion takes place) is estimated, using a manually selected set of parallel lines. The alignment between the recovered plane coordinates is then calculated as an optimization problem, assuming the transformation is affine. A similar approach is proposed in [26] based on minimal knowledge about the scene (namely, height of moving pedestrians and approximate camera height).

### REGISTERING NONOVERLAPPING VIEWS

The spatial relationship between multiple nonoverlapping cameras can be represented in different ways. The topology of the network qualitatively expresses the relative spatial relationship between cameras, explicitly identifying spatially adjacent cameras. A topological map is represented as a graph depicting the cameras as nodes and the adjacency property as the links. Alternatively, a topographical model of the network includes spatial information to relate the ground plane geometry between the camera views. For example, if cameras are calibrated to a common world coordinate system, then a geometric analysis would enable viewpoint adjacency to be computed and an estimate of inter-view path distances to be calculated. However, knowledge of

### HOMOGRAPHY BETWEEN VIEWS



#### Homography.

A homography ( $H$ ) is a  $3 \times 3$  linear projective transform that defines a planar mapping between two overlapping camera views (see the figure above)

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \quad y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

where  $(x, y)$  and  $(x', y')$  are the image coordinates in the first and second camera views respectively. Hence, each pair of corresponding image points between two camera views results in two equations expressed by the coefficients of the homography. Given at least four corresponding points, the homography can be estimated.

Given a set of detected moving objects in each camera view, we can define a match between a correspondence pair when the transfer error condition  $(\mathbf{x}' - H\mathbf{x})^2 + (\mathbf{x} - H^{-1}\mathbf{x}')^2 < \epsilon_{TE}$  is satisfied, where  $\mathbf{x}$  and  $\mathbf{x}'$  are projective image coordinates in the first and second camera views.

The homography between two cameras can be estimated using the motion of tracked objects across overlapping views with a robust estimator (see [16]). Homographies are commonly used to register two views based on calibration points (markers) or are learned by analyzing a large quantity of data and statistically estimating the most likely homography among a large number of possibilities [2], [35].

the spatial adjacency of the cameras does not guarantee that a pathway exists between the two views, such that targets leaving the view field of one camera never appear in the adjacent camera's view field. This might occur because some fixed element in the scene (such as a wall or a building) blocks the pathway between the two viewed regions. One solution would include a geometric model of the environment (e.g., a CAD model), manually tagged to indicate the potential pathways.

The topological or topographical model of the camera network can be created by correlating the observations of targets exiting the view field of one camera and entering that of another. This ensures that the model only expresses actual paths that exist between two views and enables a predictive mechanism to anticipate where the transiting target is expected to reappear. A further refinement of the model adds temporal characteristics,

representing the transit period as the average or a probability density function estimated from a set of observations.

This spatial and temporal information is most commonly extracted by directly corresponding target trajectories across pairs of views. Huang and Russell [11] describe an algorithm for the constrained task of corresponding vehicles traveling along a multilane highway between a pair of widely separated cameras (up to two miles apart). They use spatio-temporal and appearance features of the tracked objects in one view to match to those appearing after some expected transition period in the next view, thus computing a reappearance probability. Matching is performed using an association matrix to describe

**A VISUAL-SURVEILLANCE SYSTEM IS COMPRISED OF A NETWORK OF SENSORS, SOME WITH OVERLAPPING FIELDS OF VIEW, PROVIDING CONTINUOUS (24/7) ONLINE OPERATION.**

all possible pairings, then selecting the most probable based on a global minimization constrained by one-to-one pairings. However, they explicitly label the two cameras (one as upstream and one as downstream) and do not attempt to infer this information.

A similarly constrained environment is the interior of a building, tracking people moving through a set of rooms linked by corridors and imaged by multiple cameras [11], [15], [20], [37]. In this case it is not uncommon for the camera

view fields to abut one another or be slightly overlapping; then, coregistration of the target trajectories between two views will result in a zero transit period (e.g., when two cameras image the space on either side of a doorway and targets moving through

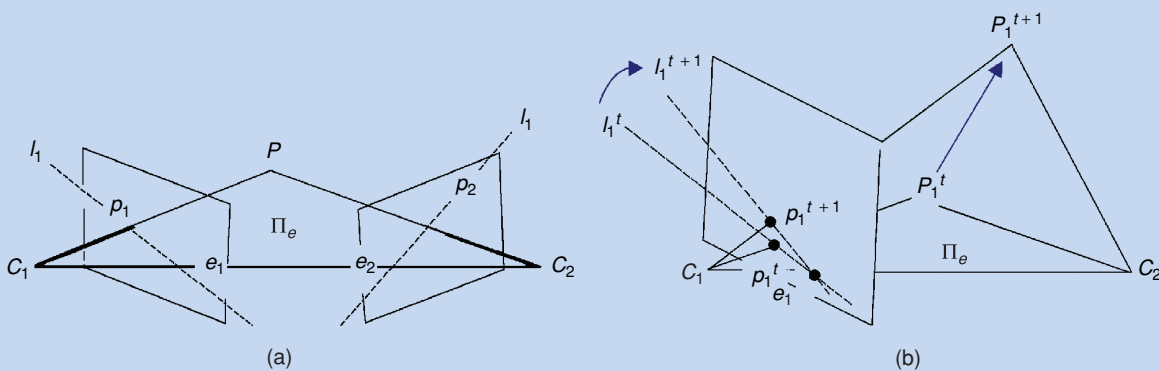
### EPIPOLAR GEOMETRY

Two overlapping views are constrained by the *epipolar geometry*. The optical center (also called center of projection) of a view projects onto what is commonly called the epipole on the image plane of another view. The two centers of projection and a 3-D point define a plane called the epipolar plane. Any point  $P$  in the 3-D world projects onto a point  $p$  in all the image planes of the multicamera system;  $p$  is visible if its projection falls inside the view cone of that particular camera. The line joining a projection point and an epipole defines an epipolar line (for instance  $l_1$  and  $l_2$  in the figure below). Any 3-D point  $P$ , visible by two overlapping views, lies on the epipolar line of the other view (for instance  $P_2$  seen from camera  $C_2$ , projection of the 3-D point  $P$  lies on the epipolar line defined by the projection of  $P$  onto  $C_1$  and the epipole of the image plane 1. This geometric constraint can be used to limit the search for a point of interest in the field of view of other cameras. If the point moves, then the associated epipolar line moves with it, defining a pencil of lines (see the figure below). This additional constraint can be used to track a moving point.

A closed-form relationship between two views can be established. A formula for the essential matrix  $E$ , relating the two views, can be derived by imposing the coplanarity constraint of  $C_1$ ,  $C_2$ , and  $P$ . Using the triple product  $(P_1 - t) \bullet t \times P_1 = 0$ , substituting the geometric relation between two views  $P_1 - t = RP_2$  ( $P_1$  being the 3-D point in the  $C_1$  reference frame) yields  $(RP_2)t \times P_1 = 0$  using

$$t \times P_1 = SP_1, \text{ where } S = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \text{ and } (R, t) \text{ the 3-D transformation between } C_1 \text{ and } C_2 \text{ reference frames.}$$

Substituting, we obtain  $P_2EP_1 = 0$ , where  $E = RS$ . So the essential matrix  $E$  provides a relationship between the epipolar geometry and the extrinsic parameters of the camera system. Using (2) of plane (multicamera geometry), we can write  $p_2Ep_1 = 0$ .



(a) and (b) demonstrate epipolar geometry.

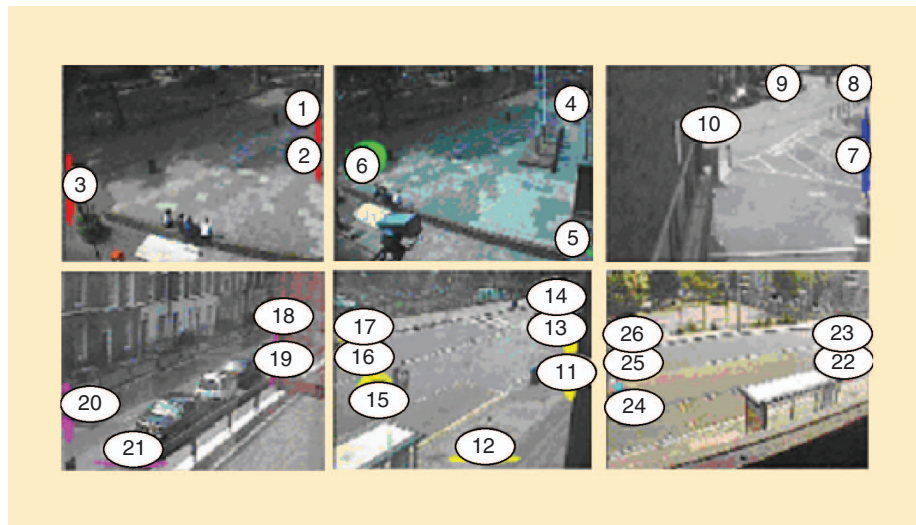
the door reappear instantly in the view field of the second camera). Similarly, corridors, like the lanes on the highway, provide a significant constraint on the typical motion paths; hence, inter-camera motion is more readily detected.

Kettnaker and Zabih [15] adopt a Bayesian approach, maximizing the posterior probabilities to solve the correspondence. They assume a known topology of motion paths between the cameras, as well as a model of the transition times and probabilities. They also attempt to reconstruct the complete trajectories. Color appearance information is combined with the transition probabilities to match the observations from different cameras. As with [11], they use an association matrix (of size equal to the total number of trajectories) to identify observation pairs between two views, thereby “stitching” the trajectories associated with both observations.

Porkili and Divakaran [23] solve the same problem using a Bayesian Belief Network to correspond targets moving in a building, principally employing color histogram features for matching. They provide a means of dynamically updating the conditional probabilities over time to account for a changing path usage, and a weighting value to create a “forgetting” function to account for targets that finally leave the camera environment. Wren and Rao [37], having calibrated the cameras in a multicamera network of 17 overlapping cameras using a very large calibration grid, proceed to treat each camera as a simple motion detector capturing events associated with people moving in the view field. They then compute the temporal cooccurrence statistics for an event disappearing from one camera and reappearing in another after some given delay period, accumulating these over all other cameras.

A more challenging, less constrained environment is encountered in unrestricted outdoor scenes, where the view fields are likely to be more widely separated and the expected transit delay increases proportionately, generally increasing the uncertainty of the reappearance time. Javed et al [20] have used a feature matching approach to track pedestrians across multiple cameras, learning a typical track’s spatio-temporal transition probability using a Parzen estimator, based on manually labeled correspondences. Individual tracks are corresponded by maximizing the posterior probability of the spatio-temporal and color appearance, adapted to account for changes between the cameras. An adaptive sliding time window is used to avoid the problem of considering the entire observation set for online use in order to cope with variations in the possible paths that may be taken.

Ellis et al. [6] have proposed a fully automatic, correspondence-free method that learns the topography of the camera



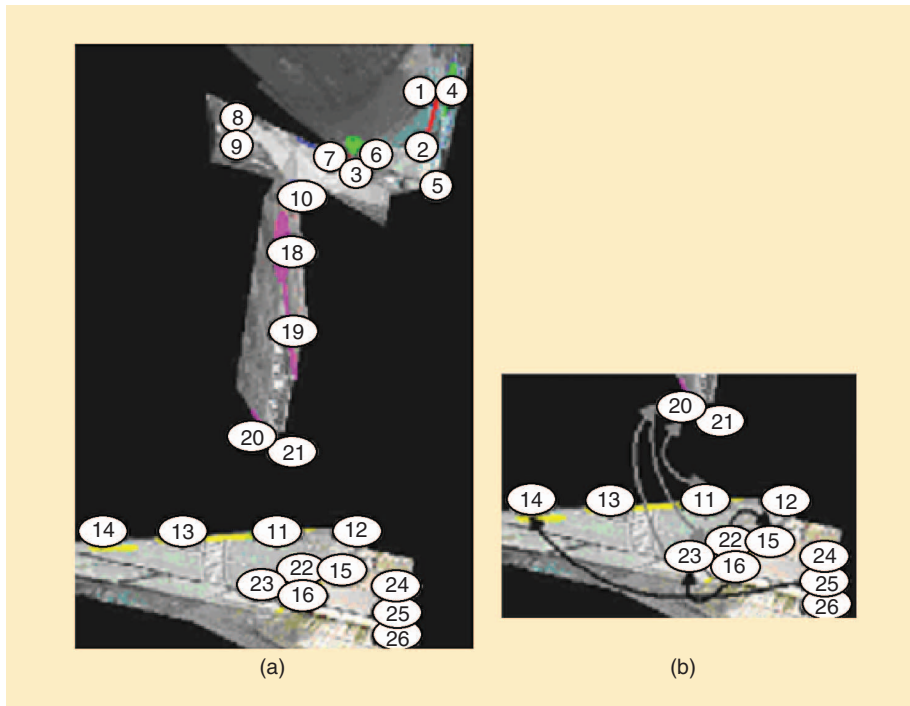
**[FIG8] Detected entry and exit zones for six cameras in the camera network. The zones are numbered as individual nodes of the activity network.**

network by utilizing a large number of observations to provide reliable statistics of transition probabilities. The unsupervised algorithm computes the temporal correlation between targets disappearing from one camera view and reappearing within the view field of an adjacent camera after a consistent time delay. The algorithm operates in two stages, first identifying commonly used regions in the image where targets consistently appear or disappear.

These regions are described as entry and exit zones that occur, for example, at the image borders, occlusion boundaries, or a doorway. The second stage correlates the transition time delay between a set of observed target disappearances against the appearance of new targets in every other camera view. Strong correlations identify the links between adjacent cameras (i.e., the topology) and the average transition period. While the lack of target correspondence means the system relies on targets moving with similar and approximately constant velocities, using a large observation set ensures robust estimation. One benefit of this approach is that by also computing correlations of ‘negative’ time, it is possible to determine the connectivity between entries and exits both within and between camera views. An online version of the algorithm is more practical than the approaches used in [11] and [15] since the size of the association matrix is determined by the number of entry and exit zones, not by the number of individual observations. Figure 8 shows an example of the entry and exit zones (numbered ellipses) that have been identified for a six-camera network; Figure 9 shows these reprojected onto the ground plane, indicating the connectivity established between parts of the network.

### WHAT THE FUTURE HOLDS

Modern visual surveillance systems deploy multicamera clusters operating in real time with embedded sophisticated and adaptive algorithms. Such advanced systems are needed for 24/7 operation: to robustly and reliably detect events of interest



**[FIG9]** (a) Entry and exit zones of the six cameras projected onto a ground-plane view of the scene; (b) detected visible (black) and invisible (grey) links between zones of the cameras 4, 5, and 6.

in adverse weather conditions while adapting to natural and artificial illumination changes and coping with hardware and software system failures.

Visual surveillance enjoys terabytes of data, and large amounts of data offers the potential for learning algorithms that can tune parameters of low-level processing like change detection as well as inter-camera and inter-cluster registration.

Ad hoc methods [5], [33], even though they achieve very high performance, might not be sufficient for the visual surveillance community. Researchers will need to develop techniques capable of disambiguating people within groups and isolating individual vehicles in traffic, enabling the tracking of individual objects in dense, cluttered scenes. Such systems will be constructed from heterogeneous clusters of sensors, including mobile and fixed cameras. Tracking will be improved by employing multiresolution extraction of features: tracking a multitude of people can be carried out with a wide-angle camera with low resolution; a zoomed active pursuit of an individual can then be carried out to extract more detailed information. The success of change detection methods achieved for static cameras is not directly applicable to active cameras, where the intrinsic and/or extrinsic parameters may change. In adopting frame-by-frame techniques to images acquired by moving cameras, *holes* can appear in the resulting blobs. Future methods are expected from the next generation of frame-by-frame motion detectors.

Visual surveillance research is already moving towards the development of self-adapting, self-calibrating algorithms [6],

[11], [15], [20], [23]. These techniques will tune all camera parameters in real-time, including zoom, focus, aperture, etc. Next-generation visual systems will react to changes as they occur, providing greater control over the parameters to improve motion detection. Similar methods have been proposed to automatically register, synchronize, and calibrate one camera or a number of cameras. It remains to be demonstrated how well these methods will scale to networks of many hundreds or even thousands of cameras. Existing techniques employ information gathered from static scenes, single and multiple moving objects from single or multiple cameras, using robust statistical techniques [2], [26], taking advantage of the wealth of information generated by the camera network. This trend will continue, taking advantage of the ever-improving technologies that support high

computational processing rates, high-bandwidth wireless networks, and online data storage in flexible databases that provide sophisticated query access to the information they have acquired. The capability of self-calibration will considerably simplify installation of future surveillance systems, allowing cameras placed in arbitrary and uncoordinated locations to operate cooperatively, creating a virtual surveillance network.

#### AUTHORS

*Gian Luca Foresti* received the laurea degree cum laude in electronic engineering and the Ph.D. degree in computer science from the University of Genoa, Italy, in 1990 and 1994, respectively. In 1994, he was a visiting professor at the University of Trento, Italy. Since 2000, he has been a professor of computer science at the Department of Mathematics and Computer Science (DIMI), University of Udine, where he is also director of the Laboratory of Artificial Vision and Real-Time Systems (AVIRES Lab). His main interests include: computer vision and image processing, multi-sensor data fusion, artificial neural networks, and pattern recognition. He is author or coauthor of more than 150 papers, contributor to seven books, guest editor of several special issues, and reviewer for several journals. He was general cochair, chair, and member of technical committees at several conferences. He has been coorganizer of several special sessions on computer vision, image processing, and pattern recognition topics at international conferences. He also serves the European Union in different research programs (MAST III, Long Term Research, Brite-CRAFT). He is a Senior Member of the IEEE.

*Christian Micheloni* received the laurea degree cum laude in computer science from University of Udine, Italy, in 2002. Currently, he is a Ph.D. student at the same university. His main interests include active vision, artificial neural networks, and pattern recognition. He is a Student Member of the IEEE.

*Lauro Snidaro* received the laurea degree in computer science from University of Udine, Italy, in 2002. He is currently a Ph.D. student in computer science at the same university. His main interests include data fusion, computer vision, and artificial neural networks.

*Paolo Remagnino* received his laurea degree in electronic engineering from the University of Genoa, Italy, in 1988, and his Ph.D. in computer vision from the University of Surrey, UK, in 1993. He has been a permanent academic with the Digital Imaging Research Centre at Kingston University since 1998. He is the editor of two books on visual surveillance and ambient intelligence. His research interests include computer vision, artificial intelligence, machine learning, and information fusion for the development of intelligent techniques for the unsupervised learning of scene understanding. He is the main promoter at Kingston University of ambient intelligence as a novel paradigm for the implementation of smart and future environments with pervasive and distributed technology. He is a Member of the IEEE.

*Tim Ellis* received the B.Sc. degree in physics from the University of Kent at Canterbury in 1974 and the Ph.D. in biophysics from University of London in 1981. He joined City University in 1979. In 2003, he joined the School of Computing and Information Systems at Kingston University, where he is professor as well as head of school. His research interests include visual surveillance, industrial inspection, colour image analysis, and vision systems hardware. He is a Member of the IEE and IEEE and is a past chair of the British Machine Vision Association.

## REFERENCES

- [1] S. Araki, T. Matsuoka, N. Yokoya, and H. Takemura, "Real-time tracking of multiple moving object contours in a moving camera image sequences," *IEICE Trans. Inform. Syst.*, vol. E83-D, no. 7, pp. 1583–1591, July 2000.
- [2] J. Black, T. Ellis, and P. Rosin, "Multi view image surveillance and tracking," in *Proc. IEEE Workshop Motion and Video Computing*, Orlando, FL, 5–6 Dec. 2002, pp. 169–174.
- [3] H.M. Chen, S. Lee, R.M. Rao, M.A. Slaman, and P.K. Varshney, "Imaging for concealed weapon detection," *IEEE Signal Processing Mag.*, vol. 22, no. 2, pp. 52–61, Mar. 2005.
- [4] P. Chang and J. Krumm, "Object recognition with color cooccurrence histograms," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Fort Collins, CO, 23–25 June 1999, pp. 498–504.
- [5] R.T. Collins, A.J. Lipton, H. Fujiyoshi, and T. Kanade, "A system for video surveillance and monitoring," *Proc. IEEE*, vol. 89, no. 10, pp. 1456–1477, Oct. 2001.
- [6] T. Ellis, D. Makris, and J. Black, "Learning a multi-camera topology," in *Proc. Joint Int. IEEE Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Nice, France, 11–13 Oct. 2003, pp. 165–171.
- [7] O.D. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*. Cambridge, MA: MIT Press, 1993.
- [8] G.L. Foresti, "Object detection and tracking in time-varying and badly illuminated outdoor environments," *Opt. Eng.*, vol. 37, no. 9, pp. 2550–2564, 1998.
- [9] G.L. Foresti, C.S. Regazzoni, and R. Visvanathan, "Scanning the issue/technology—Special issue on video communications, processing and understanding for third generation surveillance systems," *Proc. IEEE*, vol. 89, no. 10, pp. 1355–1367, Oct. 2001.
- [10] A. Hampapur, L. Brown, J. Connell, A. Ekin, M. Lu, H. Merkl, S. Pankanti, A. Senior, and Y. Tian, "Multi-scale tracking for smart video surveillance," *IEEE Signal Processing Mag.*, vol. 22, no. 2, pp. 38–51, Mar. 2005.
- [11] T. Huang and S. Russell, "Object identification in a Bayesian context," in *Proc. Int. Joint Conf. Artificial Intelligence*, Nagoya, Aichi, Japan, 23–29 Aug. 1997, pp. 1276–1283.
- [12] M. Irani, B. Rousso, and S. Peleg, "Recovery of ego-motion using region alignment," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 3, pp. 268–272, 1997.
- [13] C. Jaynes, "Multi-view calibration from motion planar trajectories," *Image Vis. Comput.*, vol. 22, no. 7, pp. 535–550, July 2004.
- [14] J.N. Kapur, P.K. Sahoo, and A.K.C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Vision, Graphics, Image Processing*, vol. 29, no. 3, pp. 273–285, 1985.
- [15] V. Kettner and R. Zabih, "Bayesian multi-camera surveillance," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Fort Collins, CO, 23–25 June 1999, pp. 253–259.
- [16] L. Lee, R. Romano, and G. Stein, "Monitoring activities from multiple video streams: Establishing a common coordinate frame," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 758–767, 2000.
- [17] A. Mittal and L.S. Davis, "M2Tracker: A multi-view approach to segmenting and tracking people in a cluttered scene," *Int. J. Comput. Vis.*, vol. 51, no. 3, pp. 189–203, 2003.
- [18] D. Murray and A. Basu, "Motion tracking with an active camera," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 5, pp. 449–454, 1994.
- [19] N. Otsu, "A threshold selection method from gray level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, 1979.
- [20] O. Javed, Z. Rasheed, K. Shafique, and M. Shah, "Tracking across multiple cameras with disjoint views," in *Proc. Ninth IEEE Int. Conf. Computer Vision*, Nice, France, 2003, pp. 952–957.
- [21] C.P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Proc. Int. Conf. Computer Vision*, Bombay, India, 4–7 Jan. 1998, pp. 555–562.
- [22] J.R. Parker, "Gray-level thresholding in badly illuminated images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 8, pp. 813–819, 1991.
- [23] F.M. Porikli and A. Divakaran, "Multi-camera calibration, object tracking and query generation," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Baltimore, MD, vol. 1, 6–9 July 2003, pp. 653–656.
- [24] P. Pritchett and A. Zisserman, "Wide baseline stereo matching," in *Proc. Int. Conf. Computer Vision*, Bombay, India, 4–7 Jan. 1998, pp. 754–760.
- [25] *Proc. Joint Int. IEEE Workshop Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Nice, France, 11–13 Oct. 2003.
- [26] J.R. Renno, P. Remagnino, and G.A. Jones, "Learning surveillance tracking models for the self-calibrated ground plane," *Acta Autom. Sin.—Special Issue on Visual Surveillance of Dynamic Scenes*, vol. 29, no. 3, pp. 381–392, 2003.
- [27] T.W. Ridler and S. Calvard, "Picture thresholding using an iterative selection method," *IEEE Trans. Syst., Man, Cybern.*, vol. 8, no. 8, pp. 630–632, 1978.
- [28] J.W. Roach and J.K. Aggarwal, "Determining the movement of objects from a sequence of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 2, no. 6, pp. 554–562, 1980.
- [29] P.L. Rosin, "Thresholding for change detection," *Comput. Vis. Image Understanding*, vol. 86, no. 2, pp. 79–95, 2002.
- [30] P.L. Rosin and E. Ioannidis, "Evaluation of global image thresholding for change detection," *Pattern Recognit. Lett.*, vol. 24, no. 14, pp. 2345–2356, 2003.
- [31] C. Schmid, R. Mohr, and C. Bauckhage, "Comparing and evaluating interest points," in *Proc. IEEE Int. Conf. Computer Vision*, Bombay, India, 4–7 Jan. 1998, pp. 230–235.
- [32] M.A. Shah and R. Jain, "Detecting time-varying corners," *Comput. Vis., Graph. Image Process.*, vol. 28, no. 3, pp. 345–355, 1984.
- [33] L. Snidaro and G.L. Foresti, "Real-time thresholding with Euler numbers," *Pattern Recognit. Lett.*, vol. 24, no. 9–10, pp. 1533–1544, June 2003.
- [34] C. Stauffer and E. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 747–757, 2000.
- [35] G. Stein, "Tracking from multiple view points: Self-calibration of space and time," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Fort Collins, CO, 23–25 June 1999, pp. 521–527.
- [36] R.Y. Tsai and T.S. Huang, "Estimating three-dimensional motion parameters of a rigid planar patch," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, no. 6, pp. 1147–1152, 1981.
- [37] C.R. Wren and S.G. Rao, "Self-configuring lightweight sensor networks for ubiquitous computing," in *Proc. Int. Conf. Ubiquitous Computing*, Seattle, WA, 12–15 Oct. 2003, pp. 205–206.