PRISMATICA: Toward Ambient Intelligence in Public Transport Environments

Sergio A. Velastin, *Member, IEEE*, Boghos A. Boghossian, *Member, IEEE*, Benny Ping Lai Lo, Jie Sun, and Maria Alicia Vicencio-Silva

***Abstract*—On-line surveillance to improve safety and security is a major requirement for the management of public transport networks and other public places. The surveillance task is a complex one involving people, management procedures, and technology. This paper describes an architecture that takes into account the distributed nature of the detection processes and the need to allow for different types of devices and actuators. This was part of a major European initiative on intelligent transport systems. Because of the dominant nature of closed circuit television in surveillance, this paper describes in detail a computer-vision module used in the system and its particular ability to detect situations of interest in busy conditions. The system components have been implemented, integrated, and tested in real metropolitan railway environments and are considered to be the first step toward providing ambient intelligence in such complex scenarios. Results are presented that not only deal with detection performance, but also on the perception of people who used the system on its effectiveness and potential impact.**

***Index Terms*—Computer vision, distributed systems, public transport, surveillance.**

I. Introduction

A STATED aim in transport policies is to produce significant shifts in traveling patterns from private to public modes. This can contribute to: reducing energy consumption, pollution, and traffic-related deaths/injuries; improve quality of life/health; and reduce levels of social exclusion. The EU-funded project PRo-active Integrated systems for Security Management by Technological, Institutional, and Communication Assistance (PRISMATICA) [1] was part of the effort to make public transport systems more attractive to passengers, safer for passengers and staff and operationally cost effective. An innovative part of this project was the integration of operational, legal, social, and technical aspects.

In the context of personal/asset security and safety, one of the main tools used by public transport networks is extensive closed circuit television (CCTV) systems. Signals from cameras are sent to control rooms where they are monitored by human operators. The rationale is that the ubiquitous presence of cameras can deter potential offenders, reassure passengers, and events that threaten safety or security will be dealt with in a timely fashion. The surveillance of public places is associated with a number of key factors such as:

1) the widespread geographical extent of what needs to be managed;
2) a wide range of behaviors that merit the attention of human operators (that need control or at least recording);
3) the variety of the type of information that needs to be processed to assess a situation, e.g., vision (direct and/or CCTV), sound, traffic data, weather information and knowledge of special events (e.g., football matches in the neighborhood);
4) the need to transmit (processed) information within a hierarchical system of control.

The main limitation in the effectiveness of CCTV surveillance systems is the cost of providing adequate human monitoring cover for what is, on the whole, a fairly tedious job. Consequently, CCTV tends to be used as a reactive tool and the perception that a public transport operator is in charge of its space is lost if no response is obtained when trouble occurs. What is desirable is a proactive approach whereby the likelihood of events can be recognized more or less automatically to guide the attention and action of the human operators in charge of managing a transport network. It is crucial to do so in a way that conceives surveillance systems as decision-support tools for human operators to deal with complex and large environments [1]–[3], in ways by which the technology itself is as transparent as possible. The focus is on ubiquitous processing to provide useable, accurate, and timely security-related information. In other words, the primary purpose is to provide ambient intelligence for the CCTV surveillance task. The technical part of PRISMATICA resulted in a distributed surveillance system that broadly belongs to the class of third generation surveillance systems (3GSS), introduced by Marcenaro *et al.* [3].

In this paper, we report on the work done first as part of the PRISMATICA research project and then on the results of a trial system evaluated in a major public transport facility in London.

Manuscript received September 12, 2003; revised April 1, 2004 and June 14, 2004. This work was supported in part by the Rome Transport Authority (ATAC) under project IPSATAC and in part by the European Commission under project PRISMATICA, whose partners included RATP-Paris, LUL-London, ATM-Milan, STIB-Brussels, PPT-Prague, ML-Lisbon, Kings College London, University College London, INRETS-France, CEA-France, TIS-Portugal, SODIT-France, FIT-Italy, ILA-Germany, and Thales-France. This paper was recommended by Guest Editor G. L. Foresti.

S. A. Velastin is with the Digital Imaging Research Centre (DIRC), Kingston University, Kingston KT1 2EE, UK (e-mail: sergio.velastin@kingston.ac.uk).

B. A. Boghossian is with Ipsotek Ltd., London SW15 2RS, UK (e-mail: boghos.boghossian@ipsotek.com).

B. P. L. Lo was with the Digital Imaging Research Centre (DIRC), Kingston University, Kingston KT1 2EE, UK. He is now with the Department of Computing. Imperial College London, London SW7 2AZ, UK (e-mail: benlo@doc.ic.ac.uk).

J. Sun was with the Digital Imaging Research Centre (DIRC), Kingston University, Kingston KT1 2EE, UK. He is now with the Manufacturing Group, School of Engineering, Warwick University, Coventry CV4 7AL, UK (e-mail: Jie.Sun@warwick.ac.uk).

M. A. Vicencio-Silva is with the Centre for Transport Studies, University College London, London WC1E 6BT, UK (e-mail: mavs@transport.ucl.ac.uk).

Digital Object Identifier 10.1109/TSMCA.2004.838461

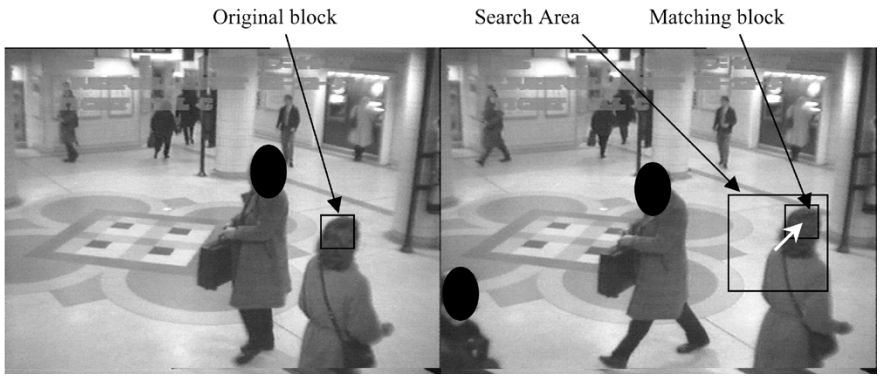


Fig. 1. Illustration of block matching (white arrow shows computed motion vector).

Because of the dominant role that vision plays in systems of this kind, Section II describes in detail one of the main vision modules used in the PRISMATICA system. Section III then provides an overview on how event detection (visual and otherwise) is dealt with as part of a complete distributed system. Section IV describes experimental results dealing with detection capability and also the reactions of personnel who used one of the systems. Section V gives overall conclusions and suggestions for further work.

II. Video-Processing Module

A. Introduction

In this section, we describe in detail a video-processing device that is part of the PRISMATICA system. Various algorithms and systems have been proposed to automate the video-monitoring task, such as the abandoned object-detection system proposed by Boghossian [4], [5], and Sacchi [6], the people-tracking algorithms proposed by Fuentes [7]–[9] and Siebel [10], the congestion-detection algorithm proposed by Lo [11], the analysis of events associated to very large crowds by Boghossian [12], the behavior-analysis system proposed by Rota [13], and the distributed digital camera system proposed by Georis [14]. Public transport environments present major challenges, as identified by the earlier CROMATICA project [15], such as the need to deal with cluttered environments. The basic clues used by human operators [16] are those derived from movement and being able to distinguish between environment fixtures (background) and transient features (moving and stationary foreground: people, objects). An important requirement is that these systems should operate continuously for any typical surveillance camera. Therefore, it is important that the video processing self-adapts to background changes and to provide simple mechanisms for scene description. It is possible to use detailed geometric/semantic scene models e.g., applying three-dimensional (3-D) scene descriptions to exploit spatial constraints and also to combine detection results from different sensors and indicate the location of the incident [17]. However, this is not necessarily practical (especially for large CCTV systems) because, to obtain an accurate 3-D scene model, extensive measurements have to be carried out and any subsequent changes in camera position or angle will lead to labor intensive recalculation of the scene model.

Methods based on statistical background estimation and subtraction [18]–[20], use temporal changes in pixel intensity/color to identify a statistical trend that points to pixel characteristics that occur more frequently. A popular technique is to use mixtures of Gaussian probability distributions to identify background pixels. The underlying assumption is that the main source of variability is that of the background itself (i.e., sparse human activity). However, the type of environments considered here are generally subject to sustained high levels of foreground activity and, therefore, such methods, used on their own, are not necessarily applicable.

This problem can be overcome through the incorporation of image motion into the background/foreground-estimation process [21]. The term motion is used here to refer to an image velocity field (i.e., magnitude and direction), as opposed to what is usually called change (i.e., foreground). Gradient-based methods (from which the term optical flow originated) are popular [22], [23], but are known to have shortcomings in preserving motion discontinuities and deteriorate with decreasing frame rates. In the context of cluttered scenes, Davies and Velastin [24], [25] proposed the use of a block-matching technique [26] to estimate the general trends of motion of crowds by analyzing frequency distributions of velocity directions. Fig. 1 illustrates what is meant by block matching. Given two images in a sequence, a block (neighborhood) is defined in the first image. Then, a search area (centered at the same center pixel position of the same block and larger than the size of the block) is defined in the second image. Within this search area, a block is then found that minimizes a given function of the pixels in the block, i.e., to locate a block that is similar to the one in the first image. The relative displacement between the original block and the matching block defines an image-motion vector. This process is typically used, as part of the motion estimation component, in video-encoding techniques such as MPEG-2 [27]. Bouchafa *et al.* [28] also considered the use of a block-matching technique to detect the direction of motion of crowds. Yin [29] conducted a detailed study and showed that accurate estimation of crowd movements can be obtained through appropriate settings of the operating parameters (size of block, size of search window). More recently, Coimbra *et al.* have demonstrated [30] that it is possible to extract pedestrian presence and motion directly from MPEG-2 motion vectors, assuming that MPEG-2 video streams are available.

A block-matching algorithm can be formulated as follows. Let N be the size of the block (i.e., each block has $N \times N$ pixels), B_r represents the block in the first image, B_c represents a candidate block in the second image, and S is the search area in the second image. A matching block B_m satisfies

$$f(B_m, B_r) \leq f(B_c, B_r), \qquad \forall B_c \in S \tag{1}$$

where $f(B_c, B_r)$ is a function that decreases monotonically with the similarity between blocks B_c and B_r (typically to zero, when both blocks are identical). For example, this function could compute the sum of the absolute pixel-to-pixel differences between the blocks

$$f(B_c, B_k) = \sum_{k=1}^{k=N\times N} \left| g\big(B_c(k)\big) - g\big(B_r(k)\big) \right| \tag{2}$$

where k is a pixel within the block and $g(B(k))$ indicates a property of that pixel (e.g., intensity). This measure (2) is normally called mean absolute error (MAE). Equation (1) implies that the matching block has to be found from the set of all possible candidate blocks. When an exhaustive search is done, we refer to a full search block matching (FSBM) algorithm. Many variants of block-matching algorithms have been proposed (e.g., see [31] and [32]), where the main emphasis has been on reducing the computational expense of finding a best match. However, this is at the expense of finding only a local minimum or making assumptions on the nature of the movement to be detected. The work described here uses the FSBM approach, because of its greater determinism (crucial for real-time applications where, for example, data-dependent delays could have significant negative effects on time-dependent parts of the algorithms) and better results. Parameters are set to those found suitable for typical CCTV installations in metropolitan railways [29] (a block size of 8×8 pixels and a search window of 24×24 pixels, using PAL images digitized at 512×512 pixels). Motion estimation is carried out using nonoverlapping blocks, so that the resulting motion field is of size 64×64 blocks.

As mentioned earlier, a drawback of the FSBM is its computational expense. From the calculations presented by Boghossian [4], it can be estimated that for this size of data a 3-GHz Pentium-class processor could achieve only about 2.4 frames/s, assuming that no other processes runs on the processor. To overcome this problem, the video detection has been implemented on a Philips Nexperia PNX1300 dedicated digital signal processor. The processor can compute a sum of differences (1) of four pixels in a single instruction cycle. Thus, the FSBM for a 64×64 motion field is performed at 5.6 frames/s (the consecutive images for block matching are still captured at the full frame rate of 25 frames/s). The use of DSP boards that digitize and process images allows a single PC-type computer to handle up to 14 separate cameras, making it an attractive proposition as a building block in a large surveillance system.

B. Motion Estimation

The first step in the process is to digitize incoming images and compute motion vectors using the FSBM algorithm. When more than one candidate match is found with the same MAE, the one that it is nearest to the center of the block is chosen. This process results in a raw set of vectors illustrated in Fig. 2.

Fig. 2. Typical raw motion vectors (shown in white superimposed on the input image).

Fig. 3. Typical result after applying mean and median filters.

As can be seen, the output of this stage, although it contains information pertaining to pedestrian motion, also shows noise typically arising from the camera and mains frequency interference, digitization and recording media noise. A full analysis of the nature of this noise can be found in [4]. A sequence of spatial filters, similar to that reported in [30], is then applied to the motion vectors to reduce this noise: a mean 3×3 filter, a median 3×3 filter, and a mean 3×3 filter. A typical result is shown in Fig. 3. Additional motion information is available in the form of pixel-to-pixel interframe differences. Using long sequences of known nonpedestrian images (e.g., taken over a period of many hours when the transport network is not in service), interframe noise is modeled by a single Gaussian to find a suitable threshold $(\mathrm{IFth} = 2\sigma)$ to identify areas of significant movement. This results in images like the one shown in Fig. 4. This can be combined with information available from an adaptive background-estimation process (explained later). Given an estimated background, a pixel of intensity $I(x, y)$ is considered

Fig. 4. Typical interframe image (inverted for clarity).

Fig. 5. Extracted motion-vector field after combining it with foreground and interframe conditions.

to be foreground if its luminance contrast (a normalized feature that is less prone to classification error than absolute intensity [7]) with respect to the corresponding background pixel $B(x,y)$ exceeds a predetermined value LCth (also obtained through a Gaussian model of *a priori* observed data)

$$\frac{|I(x,y)-B(x,y)|}{B(x,y)} > \mathrm{LCth} \tag{3}$$

where $I(x,y)$ and $B(x,y)$ are integers in the 0–255 range, for 8-bit images, and $B(x,y)$ is set to 1 when $B(x,y)=0$, to avoid division by zero.

When this occurs, the motion vector estimated by FSBM is considered to be correct (and correspond to foreground). Otherwise, the motion vector at that block position is set to zero if there is no sufficient evidence of interframe motion (as defined above). A typical result from this process is shown in Fig. 5 to illustrate the effectiveness of the approach.

C. Background Estimation

Within the space limitations of this paper, it is not possible to discuss in detail the wide literature that deals with adaptive background estimation in video sequences. Pixel-statistic approaches such as Gaussian mixtures models [33], [34] are one of the most popular techniques in the visual surveillance community, but as pointed out earlier, there are problems when dealing with scenes with sustained activity. The approach consists on combining pixel statistics and motion. Motion information is used to identify the moving parts of the image and, hence, label them as foreground regions (and, thus, not to allow them to distort the computation of statistics of background pixels). The remaining regions of the image are classified as background regions and are involved in estimating and updating the reference-background image within a statistical framework. Moreover, information from higher level processes, i.e., the detection of stationary people/objects, is also used to prevent stationary pedestrians or objects from merging into the estimated-background image. The complete process is shown diagrammatically in Fig. 6 (note how feedback is an important part of the approach).

The statistical part of the background-estimation algorithm is carried out through an m-layer ($m = 25$) array of blocks ($H[x,y,z]$, where $x = 0\ldots63, y = 0\ldots63$ and $z = 0\ldots m-1$). We refer to this as the history array. Each element in this 3-D array has two components (both initially set to zero): an estimated-background intensity $B_z(x,y)$ and a counter $C_z(x,y)$ that holds the number of occurrences (frames) of intensities around that estimated value, as can be seen from (5). At any point in time, the intensity stored on the top layer $B_0(x,y)$ corresponds to the most likely background. When each video frame is digitized and subsampled (from pixels to blocks), a set of candidate-background pixels is identified as those that do not exhibit motion according to the FSBM algorithm and the interframe results, as follows.

$I_n(x,y)$ is a background-candidate block intensity if and only if

$$\begin{cases} |I_n(x,y)-I_{n-1}(x,y)| < \mathrm{IFth}; & \text{and} \\ \mathrm{BM}(I_n, I_{n-1}, x, y) = 0; & \text{and} \\ \text{block } x,y \text{ is not stationary} \end{cases} \tag{4}$$

where $I_n(x,y)$ and $I_{n-1}(x,y)$ are the subsampled image-pixel intensities at frames n and $n-1$, IFth is the interframe threshold (estimated through a prior training phase as described in Section II-B), $\mathrm{BM}(f1, f2, x, y)$ are the block matching horizontal and vertical motion components (in the range $-s\cdots+s$, where s is the size of the search window) at block x,y on consecutive video frames $f1, f2$ and the condition block x,y is not stationary originates from the higher level analysis module (Section II-E4). An intensity-based similarity measure $E_l(x,y)$ is used to compare a candidate background block with all those in the history array

$$E_l(x,y) = |B_l(x,y) - I_n(x,y)| \qquad l = 0,1,\ldots,m-1. \tag{5}$$

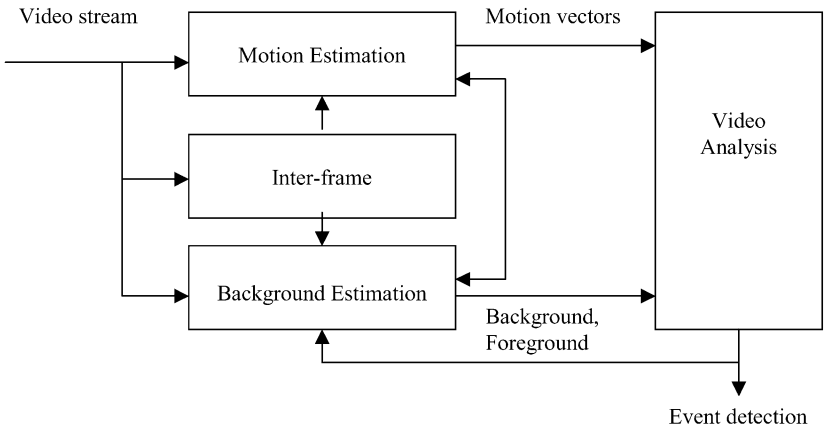


Fig. 6. Overview of process data flow.

Fig. 7. Before background estimation (Paris Metro).

Fig. 8. After background estimation.

The depth m of the history array is used to divide the range of possible intensities (256 for 8-bit luminance), i.e., into m bins. Thus, if $E_l(x,y) \leq 256/m$, then the corresponding counter is incremented, and the corresponding intensity updated by averaging the new value with the existing value (this adds an additional degree of adaptation to illumination and is equivalent to a Kalman background adaptation process with a gain of 0.5 [35])

$$\begin{aligned} C_l(x,y) &= C_l(x,y)+1 \\ B_l(x,y) &= [B_l(x,y)+I_n(x,y)]/2. \end{aligned} \tag{6}$$

If the updated occurrence value is found to exceed that of the top layer, $C_l(x,y) > C_0(x,y)$, then the corresponding entries $H(x,y,l)$ and $H(x,y,0)$ are swapped (i.e., a significant change in the background has been detected). Conversely, if $E_l(x,y) > 256/m$ for all current layers with nonzero counters, then a new record is created in an empty layer k and the corresponding array element is initialized

$$B_k(x,y) = I_n(x,y) \qquad C_k(x,y) = 1. \tag{7}$$

The approach can be classified as one based on a mixture model, but without the numerical complexity of a Gaussian approach that typically limits researchers to modeling around five background populations. This method only uses simple integer operations (division by two can be done by a simple bit shift). Fig. 7 shows a typical image at the start of this process, and Fig. 8 shows the background that has been estimated after 100 frames.

D. Scene Calibration From Motion

Most cameras are located such that all activity takes place on a single-ground plane. The main scene calibration required is that of dealing with the perspective distortion present in typical CCTV cameras normally mounted a couple of meters above the ground level looking down onto the scene at an angle θ, as shown in Fig. 9. Renno *et al.* [36] have shown a method to estimate the position of the ground plane in car-park scenarios, but it requires the tracking of objects. Here, the scene structure is estimated by analyzing the distortion in pedestrian-image motion caused by perspective projection, exploiting the fact that in

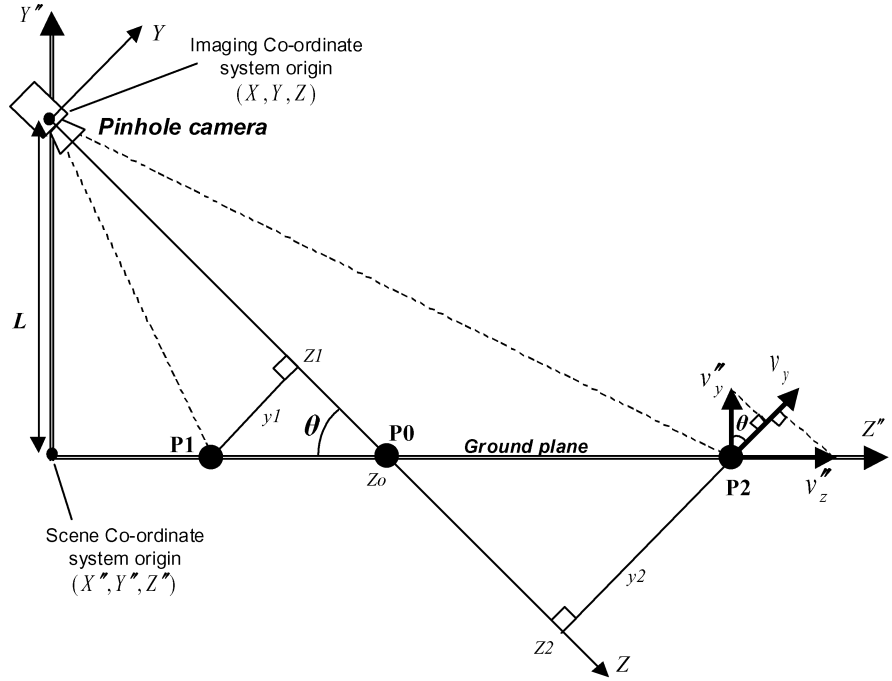


Fig. 9. Pinhole camera model with an elevated camera position.

most cases people circulate at similar real-world motion at all depths. The imaging coordinate system (X, Y, Z) has its origin at the center of the camera imaging plane. Since all moving objects in the scene are expected to move along the ground plane $(vy'' = 0)$, the y component can be written as a function of the z component

$$y = (z - z_o) \cdot \tan(\theta) = \frac{z\sin(\theta) - L}{\cos(\theta)} \tag{8}$$

where L is the camera height. Thus, if f is the camera's focal length, the projection $y\prime$ of the y component on the image plane is given by

$$y' = \frac{fy}{f - z} = \frac{f\cos(\theta)}{\sin(\theta)} - \frac{fL}{(f - z)\sin(\theta)}. \tag{9}$$

The measured (image) motion (vx', vy') represents scene motion components (vx'', vy'', vz'') projected onto the imaging coordinates system (X, Y, Z) prior to being imaged onto the image plane, as shown at point P2 in Fig. 9. It can be shown that $vx = vx''$ and $vy = vy''\cos(\theta) + vz''\sin(\theta)$. Equation (9) becomes

$$(vx', vy') = \left(\frac{fvx''}{f - z}, \frac{f\big(vy''\cos(\theta) + vz''\sin(\theta)\big)}{f - z}\right). \tag{10}$$

Typically, the world-motion vertical component (vy'') is zero (or nearly zero), because pedestrians are expected to move (mostly) parallel to the ground plane (apart from small vertical oscillations that result from walking). Therefore, (10) can be reduced to:

$$(vx', vy') = \left(\frac{fvx''}{f - z}, \frac{fvz''\sin(\theta)}{f - z}\right). \tag{11}$$

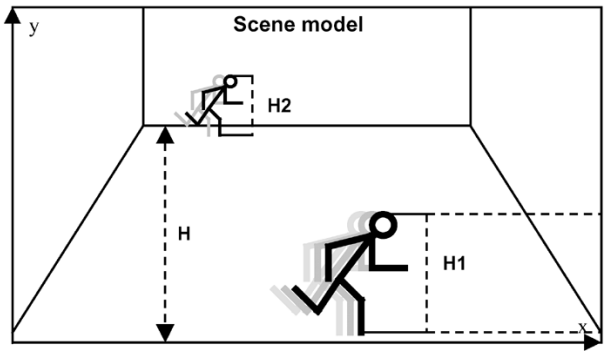


Fig. 10. Scene model (parameters H, $H1$, and $H2$).

Assuming a constant scene velocity $(vx'', 0, vz'')$, constant imaging parameters (f and θ), and that $z \gg f$, the image-object velocity (vx', vy') will be inversely proportional to object depth (z). Fig. 10 shows how scene geometry is represented by three image parameters: the position of the scene back plane (H), the average height of a pedestrian at the front edge of the visible ground plane $(H1)$, and the average height of a pedestrian at the back plane $(H2)$. From the above discussion, it is expected that the measured image-motion components (vx', vy') will follow a profile that depends on image y coordinate. So, for $0 \le y \le H1$, motion is expected to be constant (independent of depth, this section represents the motion vectors generated by pedestrians moving at the front edge of the ground plane and extends along their body height), $H1 \le y \le H + H2$ motion is expected to decrease linearly with y (i.e., inversely proportional to depth) and $H + H2 \le y$ motion is expected to be zero (where no

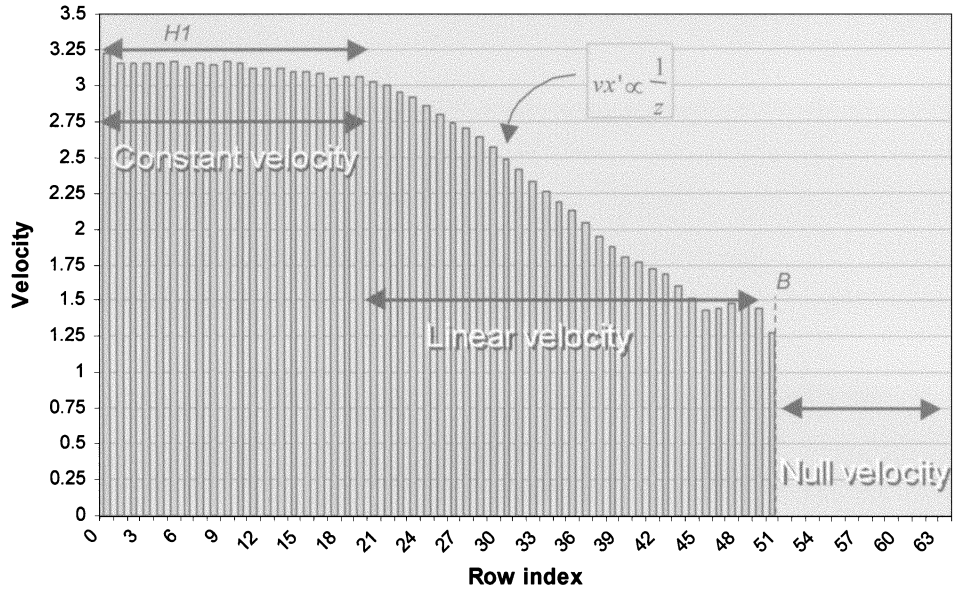


Fig. 11. Velocity profile obtained from live video.

objects are expected to be present). For a given camera position, the measured profile can give an indication of scene geometry.

A velocity profile $V(h)$ is calculated by applying a temporal median filter (to remove outliers) to the average motion magnitudes at each image row (using only nonzero measured motion vectors)

$$V(h) = \text{median}\left[\frac{1}{N}\sum_{i=0}^{N}|v_n'(i,h)|\right] \ldots h = 0,1,2,\ldots,N \tag{12}$$

where N is the number of image rows, h a row index, and $v_n'(i,h)$ is the motion vector at column i and row h of the current digitized frame. An example velocity profile measured from live video (using 64 vertical blocks) is shown in Fig. 11 and can be seen to exhibit the three expected different sections. The parameters that define the scene geometry (perspective) can be estimated from the break points between the three sections of the velocity profile and the measured velocity values. First, the average height of a pedestrian at the front edge of the visible ground plane $(H1)$ is the width of the constant velocity section. Then, the perspective distortion ratio (R) is measured as the ratio between the average velocity magnitude of the constant velocity section $\bar{v}_c$ and the velocity value v_b measured at point B (at the start of the zero-velocity section). Hence, $(H2)$ can be calculated as follows:

$$\frac{\bar{v}_c}{v_b} = \frac{H1}{H2} = R \Rightarrow H2 = \frac{H1}{R}. \tag{13}$$

Finally,

$$H = B - H1. \tag{14}$$

The value of H is derived on the basis that the back plane starts where the furthest pedestrian meets the ground plane. The front edge of the ground plane is a virtual limit for pedestrian motion that was set by the scene model to allow the calculation of $(H1)$ and the perspective distortion ratio (R). Therefore, it is possible for pedestrians to be closer to the camera and hence the visible parts of their bodies will exhibit higher motion magnitudes. This could result in a velocity profile that has no and even extend the linear section to the bottom of the image. In order to correct this problem, the image motion is segmented based on position and direction connectivity, and the regions that overlap with the image bottom row are ignored. This ignores pedestrian motion if their movement is on or below the front edge of the visible ground plane.

To illustrate the performance of this algorithm, a set of tests is shown here (Table I) with eight different scenes (direction of movement, obstacles such as columns and camera position). Estimated scene parameters (R, H) are compared with those measured manually. Scene-structure geometries for scenes with dominant horizontal paths are estimated within an error of 5% as in test (1). However, poorer estimates of the structure parameters are obtained in scenes where the dominant projected direction of motion is vertical, as in tests (7) and (8). This is because the vertical component of the image object velocity vanishes rapidly with depth causing poorer accuracy in the estimation. Also, queues and other obstacles have measurable effects on object (pedestrian) velocities. Therefore, the algorithm might not converge so that the velocity profile would not exhibit the three characteristic sections necessary for this self-calibration procedure, as occurs in test (2) and (3), the latter being a case where the camera has a side view of ticket validation barriers. The time required to estimate these scene parameters depends on the scene-motion properties (e.g., a constantly empty scene can never be calibrated in this way). In practice, whenever possible, the parameters $H, H1, H2$ are estimated through an operator on system installation (by clicking on the image). Then, the algorithm described here runs in the background to continuously update calibration parameters. This is also useful to deal with (and detect) camera movements inevitable over extended periods of time.

TABLE I
SCENE GEOMETRY PARAMETERS ESTIMATION FOR EIGHT DIFFERENT SCENES (RELATIVE ERRORS ARE SHOWN IN BRACKETS)

Index	Test description: Obstacles	Projected motion: Horizontal	Projected motion: Vertical	Projected motion: Diagonal	Queues	Low camera	Manual measurement: Perspective distortion (R)	Manual measurement: Ground plane extent (H, rows)	Automatic estimation: Perspective distortion (R)	Automatic estimation: Ground plane extent (H, rows)
1		✔					2.2	40	2.3 (5%)	40 (0%)
2	✔	✔			✔	✔	-	-	-	-
3	✔	✔				✔	-	-	-	-
4	✔	✔	✔			✔	3.9	49	4.0 (3%)	50 (2%)
5		✔	✔	✔		✔	3.3	43	3.3 (0%)	44 (2%)
6			✔	✔			2.8	39	2.6 (7%)	38 (7%)
7			✔	✔			2.3	46	2.0 (13%)	46 (0%)
8			✔				2.3	45	2.0 (13%)	43 (4%)

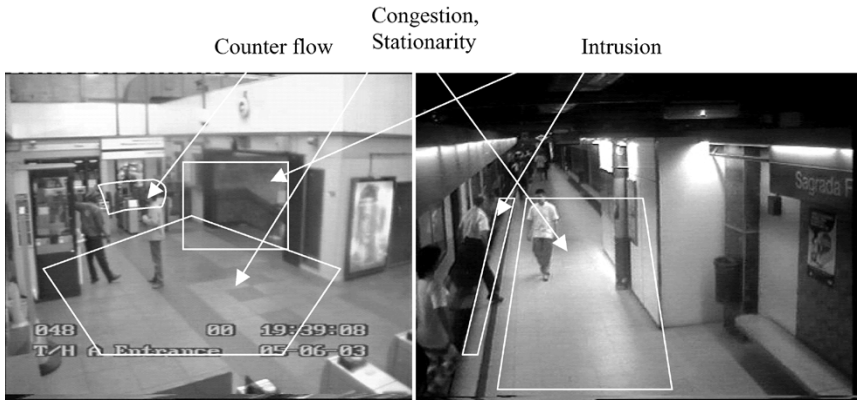


Fig. 12. Some examples of environments where different detection takes place (London, UK, and Barcelona, Spain).

E. Event Detection

When a camera is set-up, an operator can define areas of interest (AOI), or an arbitrary set of polygons, where each detection process will take place. The possible detection processes are: 1) overcrowding/congestion; 2) unusual or forbidden directions of motion; 3) intrusion; and 4) stationarity.

The definition of these AOI is typically carried out when a camera is connected to the processing device. Typical examples are shown in Fig. 12 where, for example, intrusion is focused (on the left camera) on an entrance that is out-of-bounds to passengers after a certain time in the evening and counterflow is concentrated on a set of entry gates.

1) Overcrowding and Congestion: Overcrowding refers to the presence of too many people in an area. Congestion refers to the inability of people to move easily within an area. Congestion is regarded as a major problem, especially in busy old metropolitan railway systems as the potential blocking effects of a small crowd that remains stationary (possibly as a result of unseen congestion up/downstream) could be more significant to the presence of a larger, but moving, crowd. Fig. 13 shows an example of congestion near ticket offices. This also illustrates a situation where conventional localization and tracking of pedestrians is unlikely to work. The estimation of crowding levels in public places gained significant interest in the earlier literature [37]–[43], as it plays an important role in ensuring public safety and on measuring levels of service. Here, we follow the earlier work of Velastin *et al.* [43] that proposed the use of a monotonic relationship between the number of observed foreground image features (e.g., edge pixels, vertical edges, foreground pixels, circles, blobs, etc.) and crowding levels (or the

Fig. 13. Example of a congested situation.

Fig. 14. Abnormal direction of motion, shown by white vectors (Paris Metro).

Fig. 15. Intrusion near the edge of a platform (Rome Metro).

number of people in the scene). In the module described here, the approach has been extended so that each foreground block (moving or otherwise) is first given a weight depending on its position on the ground plane (as per the calibration procedure described in Section II-D). If the low-pass filter-weighted sum exceeds an operator-defined threshold, then an alarm is raised (the time constant of the filter is set by prior observation of high frequency oscillations generally associated with moving crowds [24]). The alarm is maintained until the crowding level goes below one half of the triggering threshold (a Schmitt trigger). The detection of congestion operates in a similar manner, except that only moving foreground blocks are selected.

2) Unusual or Forbidden Directions of Motion: The set of motion vectors measured by the process described earlier is used so that in a given AOI, a histogram of motion directions is used to detect a significant peak in a given range of known forbidden directions. Such a range can be either predetermined by an operator or obtained through an off-line learning process based on hidden Markov models. Fig. 14 shows a typical result of detection of counterflow motion on what is designated as a one-way corridor in the Paris Metro.

3) Intrusion: Intrusion (or trespassing) refers to the presence of people or objects in a forbidden area. Typical examples include the avoidance of people crossing a safety line at the edge of a platform or when people are detected in an area that has been set as out-of-limits (e.g., after hours). Through the background estimation process described in Section II-C, image blocks are labeled as being either background or foreground (and if foreground, additionally labeled as moving or stationary). Foreground blocks are then formed into blobs using a region segmentation and histogram projection procedure described in [7]. The scene-geometry parameters obtained through the process explained in Section II-D are applied to refer these blobs to the ground plane. A foreground blob found in an intrusion AOI and that satisfies an operator-defined minimum size constraint (to eliminate small objects such as newspapers) and that has been detected in the area for an operator-defined amount of time (typically 2 s), generates an intrusion alarm. A typical example of a person found to be too near a platform edge (Rome Metro) is shown in Fig. 15. In traditional surveillance, intrusion is mainly associated to keeping areas sterile (free of people or objects). This might be done with simple presence/motion detectors (e.g., infrared), but these are of little use in public transport environments (where presence is common) and are generally poor at localizing the event within an image. Moreover, there is an additional cost in installing and maintaining sensors additional to the existing CCTV infrastructure and, thus, the exploitation of existing cameras to detect this type of event is economically attractive to CCTV operators.

4) Stationary Areas: The presence of stationary people or objects in a public transport environment is a matter of concern to those that manage such spaces. Typical examples include begging, loitering, abandoned packages (a cause for regular station evacuation), and graffiti (its appearance as new foreground can be regarded as a new stationary object). The results from a detailed survey of transport-network operators [16] suggests that the normal maximum period for individuals to remain stationary in underground stations is around 2 min. Detection of stationary objects or people in complex environments has been addressed in the past through three approaches: temporal filtering [44], frequency-domain methods [24], and motion estimation [28]. The typical problems associated with the detection of stationarity in complex scenes are: 1) frequent occlusion of the stationary object by moving pedestrians; 2) occlusion of the stationary object by moving pedestrians wearing shades of color similar to the background; and 3) continuous change in the pose and position of human subjects suspiciously waiting in public places.

We define an array $\mathrm{ST}(x, y)$ that holds the number of frames during which each image block (x, y) of 8×8 pixels is stationary. Each image block is processed by taking it as a candidate for a nonmoving area. If it satisfies two conditions, then it is not background (3) and it experiences no motion $(\mathrm{BM}(f_1, f_2, x, y) = 0)$. Then, cells in the $\mathrm{ST}(x, y)$ array corresponding to candidate blocks are incremented on each new frame, unless they are found to be background *and* there is no motion. These conditions provide immunity against occlusion

(a) (b) (c)

Fig. 16. Dealing with changes in position and pose (images magnified for clarity): (a) Stationary person detected, (b) Person moves to the right, (c) Person redetected after 3 sec.

(a) (b) (c)

Fig. 17. Maintaining detection of stationary people/objects during occlusion (Rome Metro): (a) Stationary person detected, (b) Detection is maintained during occlusion, (c) Detection is maintained after occlusion.

including cases of moving people with grey levels similar to the background. A region-growing algorithm is used to update the array cells to account for changes in position or pose. Image blocks removed from the array due to sudden changes in position are reintroduced to the array at the new positions by this algorithm, allowing slow or overlapping changes to be recovered within a few seconds (typically 3 sec). A final process clusters neighboring blocks that have remained stationary for a period longer than a user-defined value (typically 2 min), into blobs referred to the ground plane. The presence of one or more of blobs exceeding a user-defined size then triggers the detection of this type of abnormal situation. Figs. 16 and 17 show typical examples. The example given in Fig. 17 is particularly interesting as it illustrates how stationary detection is maintained even in the presence of occlusion (by moving passengers). Also, the lower detected blob corresponds to the person's shadow. In contrast with other reported surveillance work, there is an intentional decision not to remove shadows because in some cases these could be the only visible part indicating presence (e.g., if someone is hiding behind a pillar). The detection of stationary blocks closes the loop between this higher level of analysis and the lower level of background estimation, as shown in Fig. 6 and explained in Section II-C.

This section has described in detail how this type of surveillance image can be processed to extract information useful for the CCTV monitoring task. This type of analysis is a necessary part of a surveillance system. However, there are important issues of usability, scalability, and variability of sources that need to be addressed in a practical large surveillance system. This is discussed in the next section.

III. PRIMATICA System

This section describes the main architectural features of the PRISMATICA system. Apart from pure technical aspects, an important aspect of the design of this type of surveillance system is the incorporation of features that take into account how surveillance tasks are carried out by human operators. The purpose of providing unobtrusive, augmented surveillance capability is then to make the best use of the human abilities to make high-level decisions rather than to engage them in tedious random monitoring [45].

A. System Components

An important part of the PRISMATICA project was the investigation of an appropriate instrumented detection/action environment that enables control room operators to obtain timely information to improve personal security, e.g., in metropolitan railway systems. Key requirements include:

1) distribution of processing, given the geographical spread of sensor and the computational power required to extract meaningful information from them (e.g., using computer vision);
2) the integration of different types of devices into a flexible system architecture to mirror the variety of information sources that are needed to support decision-making and

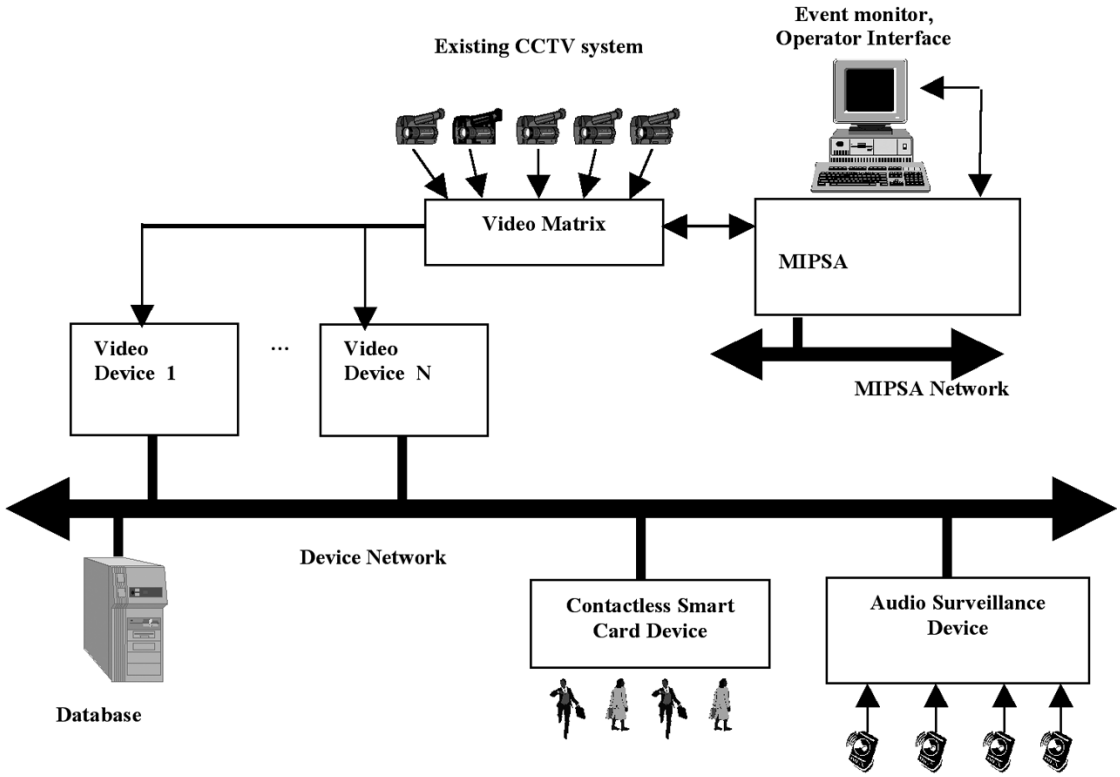


Fig. 18. PRISMATICA system components.

to support future improvements in the development of detection devices;

3) convergence of information into an integrated form of presentation (human computer interface) and retrieval (database);
4) exploit the use of existing site infrastructure (people and hardware) to facilitate early deployment and take-up by public transport operators.

A PRISMATICA system was conceived as a distributed surveillance architecture consisting of a set of diverse *devices*, each of which can contribute added value to the monitoring task, in a localized manner. That is to say, each device deals with a relatively small physical area (e.g., a camera, a microphone, a mobile camera, a mobile panic button) without necessarily being required to handle global information. An analogy is a human guard checking that people do not jump over the gates in a particular area of the station. This is their limited task, dealing with it locally and sending information to a more central supervisor only when needed. The supervisor, on the other hand, could also instruct the guards from time to time to change their task or their location. In PRISMATICA, these devices are capable of processing/analysis, so they are also referred to as *intelligent devices*.

Using the same analogy, a supervisory point is needed to coordinate the action of and to gather information generated from devices so as to assist with the decision-making processes for taking preventive or corrective actions. This analogy, gave rise to the concept of a supervisory computer modular integrated passenger surveillance architecture (MIPSA). This part of the system provides a single point of contact with an operator and a means of controlling and communicating with intelligent devices. This communication takes place over a local area network (to provide scalability), using a CORBA-based architecture that both support control messages (using a protocol encapsulated in XML) and bulk data transfers using sockets. Fig. 18 shows a simplified schematic diagram of a PRISMATICA system and its components. Signals from the existing CCTV system are fed into a video matrix, controlled by the MIPSA. This serves two purposes. First, images are locally digitized by this matrix and captured by the MIPSA for immediate display for an operator (Fig. 19). Second, under operator control specific camera signals can be routed to video devices for processing. To demonstrate that the architecture is suitable to handle diverse sensors, the PRISMATICA system also includes a device to capture signals from smart cards (developed by the Paris Metro). When a passenger carrying one these cards double clicks on its button, the signal is picked up by one or more station beacons that then forward the data to a device that sends the information to the MIPSA. From information on the position of the beacons, the MIPSA can localize the call, generate an alarm message and display images from the cameras that cover the area where the call originated from. An audio-analysis device (developed by Thales Underwater Systems, France) has also been developed to detect unusual sound signatures, typically arising from distress calls, fights, etc. Similarly, upon detection of such events data is sent to the MIPSA which then localizes the event, generates an alarm message, and displays images from relevant cameras.

As shown in Fig. 18, communications to and from the various devices and the MIPSA take place on a local area network, here called the device network. For deployment in multiple stations, a separate network, MIPSA network, is used to coordinate two or more MIPSA systems (a discussion of this facility is outside the

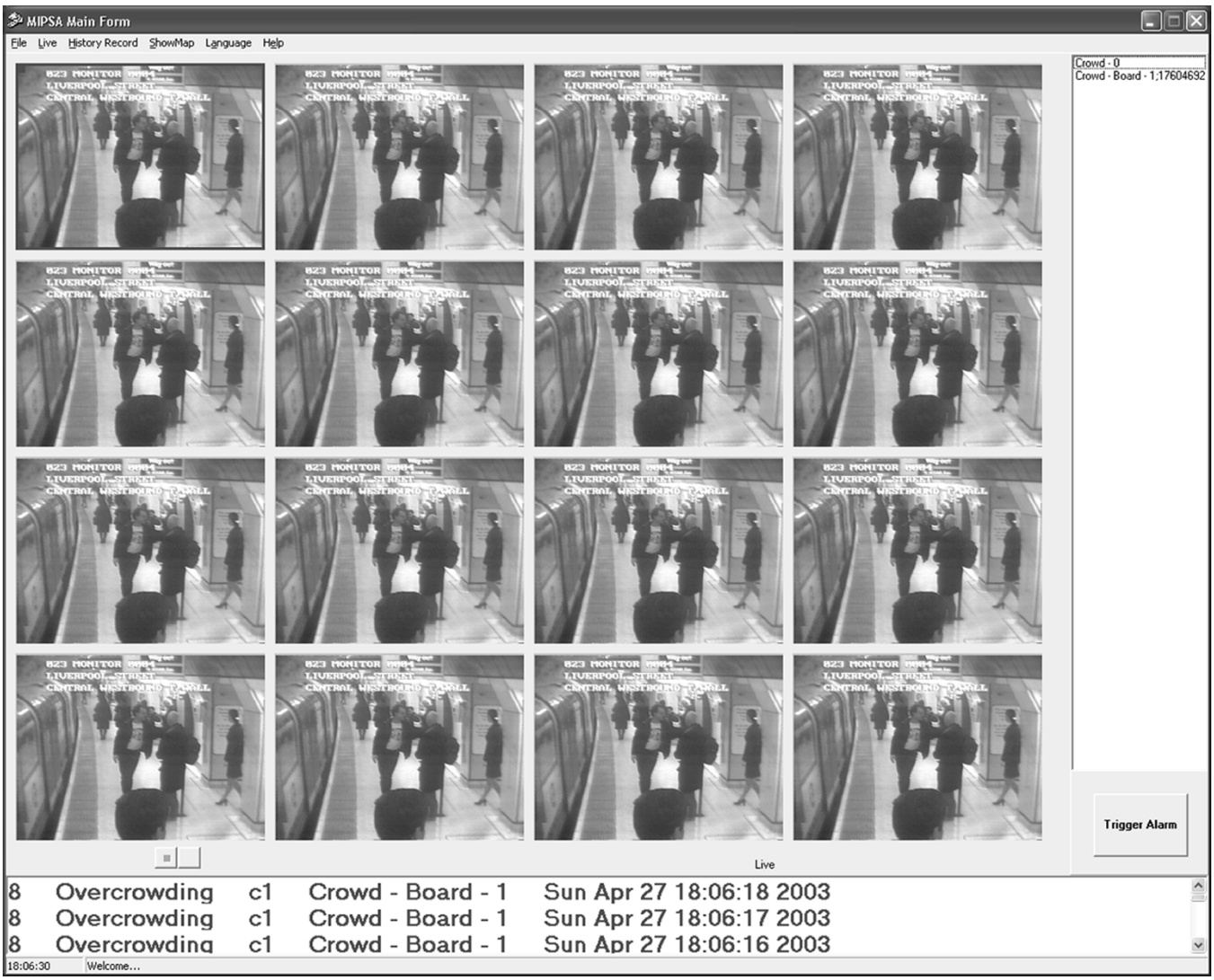


Fig. 19. Video display on the MIPSA monitor.

scope of this paper). The database holds system-configuration data, event annotations and video sections (pre- and postevent) used for query-based retrieval of events/video.

B. Device Model

This section outlines how characteristics of surveillance systems with multiple and diverse sensors have been generalized to make the proposed solution flexible for future deployment. In PRISMATICA, a *device* represents a computing subsystem that complies with a given data protocol and that can handle one or more sensors (or actuators). At one level, a device describes (to the MIPSA) how many sensors it deals with, what events its sensors can detect (and what data it sends upon detection), what type of data can be retrieved from its sensors, what data its actuators accept and how it can be configured. In this context, the device defines a *class* (e.g., a four-camera system Version 3.1) of identical devices. Each device of the same class that has successfully negotiated a connection to the MIPSA is an *instance* of that device class (e.g., Video Device 1 in Fig. 18). In the demonstration PRISMATICA system, the audio device handles four microphones (sensors) independently (the same type of process is applied to each microphone), the smart card device typically deals with six beacons, and each video device handles one camera signal (one sensor). A sensor refers to the processing directly associated with a physical transducer (camera, microphone, fire detector, etc.) or actuator and, therefore, is directly meaningful to an operator and used in a geographical representation of the site. A sensor can measure or detect one or more events. Depending on the processing capability of the sensor, such events can be either simple (the door has been opened) or fairly complex (person at position x, y has been there for 22 min and has been walking up and down in a suspicious manner). Simple sensors will normally be associated with only one event (e.g., fire detected), but it is also possible for a sensor to be capable of detecting a number of events. For example, the sensor for the video device described in Section II detects the following events: overcrowding, congestion, unusual direction of movement, intrusion and stationary person/object. Events can be of different types such as alarm (an incident has occurred), measurement (a continuous quantity such as the number of people in an area), or status (system information such as power failure). Events are further characterized by subtypes such as binary (a

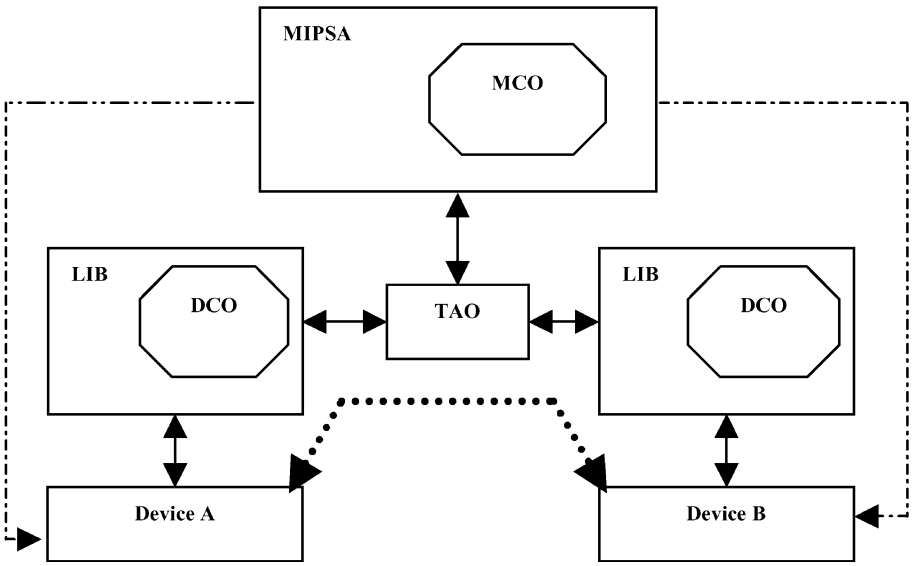


Fig. 20. Overview of MIPSA/device communications architecture (LIB denotes a wrapper that hides CORBA communications from device applications).

message is sent when the event is first detected and then another message is sent when the alarm condition no longer exists) or pulse (a regular set of messages associated with one alarm and associated data, e.g., an indication of a stationary object, its position, and the amount of time it has remained stationary). Finally, the concept of event groups encapsulates the idea that what a user might regard as an event can be a combination or aggregation of evidence captured by one or more sensors. For example, this applies to the smart card device, where the *event* is the detection by one radio beacon that a smartcard button has been pressed, the *sensor* is the radio beacon, the *device* is the beacon data concentrator, and the *event group* is the set of simultaneously activated beacons (by the same card).

C. Communications

One of the main drivers in the design of the PRISMATICA architecture was the ability to add devices dynamically and with little or no reprogramming or reconfiguration of the main server (the MIPSA). This section summarizes how this was achieved. The primary network communications-control layer is provided by ACE/TAO, an implementation of CORBA services, chosen because it is open software and has been optimized for real-time operation [46]–[49]. Schematically, a PRISMATICA unit can be represented as shown in Fig. 20 (shown for two devices A and B). ACE/TAO services are wrapped in a dynamic library (LIB) that provides a simple C-language interface so as to allow developers with no CORBA experience to implement devices (PRISMATICA involved eight different technical teams from different European countries). When the MIPSA starts, it creates a CORBA object referred to as a MIPSA communication object (MCO). This object is also given a name (MIPSA) and registered with what is called a naming service (NS), a standard CORBA facility. This allows any software running on the same network to access the MCO through that name. The MCO is a simple object that can get/send text messages from/to devices and that also provides a network-wide time reference (for the time-stamping of events). When a new device is connected to the network, it creates its own communication object device communication object (DCO). It then locates the MIPSA object and sends it a class registration message. This provides all necessary information to the MIPSA on the capabilities of the device (number of sensors, how it is configured, etc.). The MIPSA instantiates the device by giving it a unique identifier and associating it with the DCO created by the device. At this point, the MIPSA and the device can communicate with one another. For example, the MIPSA sends the device-configuration information stored in the database (this could include connection to a particular camera) from the last time the device was connected to the system. Conversely, if for any reason the device is taken out of the system (e.g., for maintenance), it signs itself out. In short, these mechanisms provide a flexible way of scaling the system up to any number of devices (subject to overall physical limitations, such as network bandwidth). All the interaction between devices and the MIPSA is done through text messages, encapsulated in an XML protocol. This simplifies design, testing and expansion at the price of higher bandwidth requirements. The protocol is sufficiently rich to allow a wide range of devices. Full details on the protocol can be found in [50].

A device connected to the MIPSA can also establish a link to any other device in the system. This caters for situations that might benefit from such direct communications links (e.g., a camera sending data directly to one of its neighbors or an audio device prompting a camera). This link between devices is shown as the thicker dashed line in Fig. 20. Furthermore, devices or the MIPSA may need to send/receive large amounts of data between one another. In this architecture, it is possible to set up socket communication links between devices and the MIPSA (or any other device). There are three types of connection: Multicast (broadcasting), TCP (point-to-multipoint), and UDP (point-to-point, asynchronous). Any device, or the MIPSA, can act as the server or client in a socket connection. Data can be distributed or sent to another device, once the socket connection is established. In Fig. 20, an example of a multicast connection is shown by the thinner dashed lines, where the MIPSA acts as the multicast server, and the devices are the clients. The overall aim has

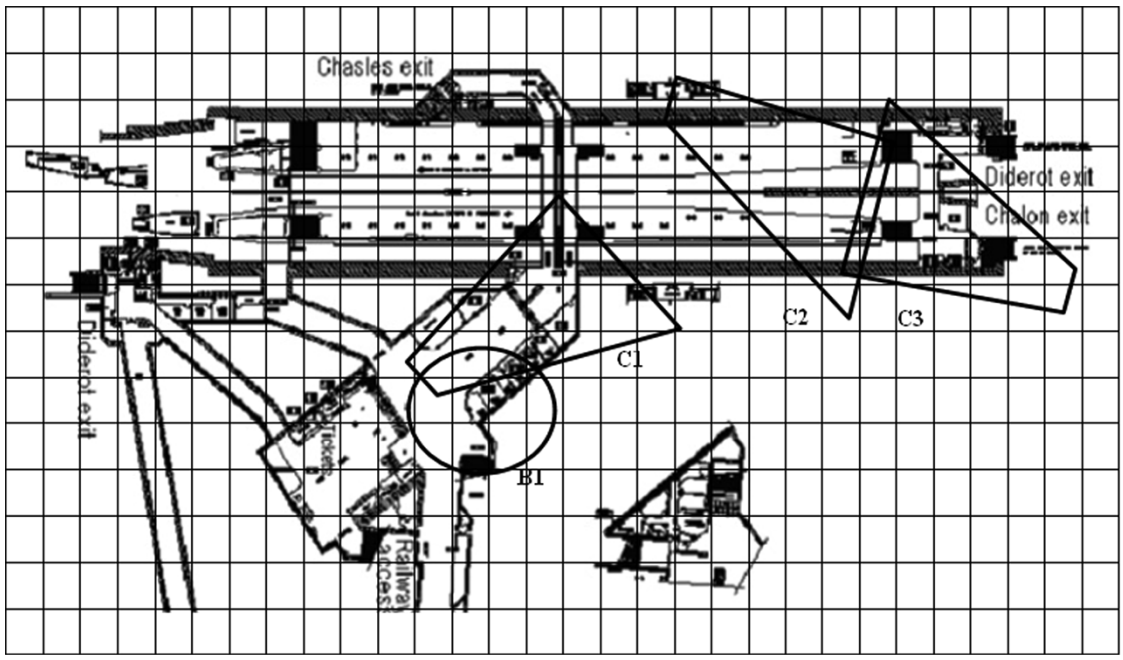


Fig. 21. Schematic representation of sensor position and coverage.

been to emulate as far as possible the different types of communications that can take place in a monitoring environment.

D. Event Visualization

The site under surveillance is represented by a set of maps (isometric and/or plan views) to present familiar views to operators. Each map typically covers a surveillance zone, such as a platform, a ticket hall, a corridor, etc. A schematic example is shown in Fig. 21. When the system is configured, sensors (cameras, microphones, etc.) are placed on these maps and assigned grid coordinates. Their approximate ground coverage is also defined in terms of these coordinates. This gives useful topographical information about the sensors in the site and their overlap (if any). For example, if in Fig. 21 the circle labeled B1 represents the coverage of a smart card beacon and an event is detected by that beacon, the system is able to associate that event with the camera whose view overlaps the beacon's coverage (the trapezoid labeled C1). In the same way, when an event is detected by camera C2, the system can also show the images from camera C3 because they share some coverage. Under routine operation, control room operators can navigate the site by selecting one of the maps and then clicking on an area covered by sensors to obtain further information, such as the real-time images from the corresponding camera(s). An important and often underestimated aspect of surveillance [51] is the use that operators make of logical relationships between sensors (mostly cameras). For example, when an operator sees that a train platform starts to become overcrowded, s/he then normally looks at the number of people getting into the station at entrances, which can be far away from platforms. S/he might also consider information on expected time of arrival of the next train service. This geographically dispersed set of data is important for operators to take pre-emptive actions. The PRISMATICA system deals with this by allowing sets of (disperse) cameras to be grouped. This is called a topological mode (in contrast to the topographical mode described earlier). In this mode, when an event is viewed by an operator, all the cameras in the same group are shown together (this also works in routine monitoring when an operator is navigating the map and clicking on any given camera coverage areas). When an event is detected by a device, it can send graphic primitives to highlight the event on the operator screen. This is another example of the decision, linked to system scalability and maintainability, to allow devices to determine what they do and what information they send.

The detection of an event triggers an audible and visible warning for the operator on the multiple-view screen shown in Fig. 19. The camera where the event originates (either directly from a video device or indirectly from camera coverage overlap as it would occur, e.g., from audio detection) is highlighted on that screen. When the operator clicks/touches on that camera display, the event is shown in more detail, together with the associated cameras. Fig. 22 shows an example (the larger image corresponds to that where the event is detected and the smaller ones are those of topographically related cameras).

E. Topographical Alarm Management

The PRISMATICA system is potentially a large system that consists of many devices. Although only high-level event messages are sent to the MIPSA and shown to the operators, potentially there could be too much information for the operator to process and s/he could not react to a situation immediately. As devices are monitoring different aspects in the environment, an incident in one area could trigger multiple alarms from different devices. For example, in an assault situation, the audio surveillance device may detect people shouting, passengers may push the panic button on their smart cards, and the video device may detect unusual movements. Consequently, the audio surveillance device, the smart card and the video camera will send (separate) alarm messages simultaneously to the MIPSA.

Fig. 22. Event highlighting (London Underground).

Even though the messages denote the same incident, the operator might need to go through all the messages, look at the associated video images, interpret the situation, and take appropriate action. Instead of assisting the operator, these could complicate the surveillance process. What is needed is a means of grouping event evidence from multiple sensors (that independently do not need to know that they have overlapped coverage) to reduce the number of (separate) alarms and to increase the priority given to such event groups. This section describes how this is done using topographical information. This is an extension of what the authors described in [52]. The system also uses a Bayesian framework to fuse multiple device information, but this is not dealt with here, due to space limitations (please refer to [52] for details).

Each map in the system (see Fig. 21) is divided up into cells with coordinates (x, y, z), where z represents the map number. Each event is associated with a priority (an integer value between 1 and $\mathrm{EP}_{\max}$), allocated initially by the device, but modifiable by the operator. For each sensor i registered in the system, a cell in a map is said to be covered by the sensor if $\mathrm{SC}(i, x, y, z) = 1$ (otherwise, $\mathrm{SC}(i, x, y, z) = 0$). At any given time t, an alarm-type event e associated with that sensor is represented by $\mathrm{EP}(i, e, t)$ which can take a value from zero (no event has been detected) to $\mathrm{EP}_{\max}$. Once an event is detected, it is required that the priority level $\mathrm{EP}(i, e, t)$ is kept nonzero by devices for at least twenty sample periods (a sample period is typically 100 ms) so as to account for variations of detection times between sensors. Then, the event status associated with a particular map cell is as follows, where N is the number of sensors in the system

$$P(x, y, z, t) = \frac{\sum_i^N [\mathrm{SC}(i, x, y, z) \times \sum_e \mathrm{EP}(i, e, t)]}{\sum_i^N \mathrm{SC}(i, x, y, z)}. \quad (15)$$

TABLE II
MOTION DIRECTION ESTIMATION PERFORMANCE

Walking direction		True positive	False positive	True negative
Up	Down			
68%	32%	99.6%	0.8%	0.4%

Thus, this calculation accumulates and fuses information from various sensors to reduce the number of multiple alarms associated with a particular area. The value of $P(x, y, z, t)$ is mapped to a set of user-defined colors associated with increasing levels of alarm priority.

IV. EXPERIMENTAL RESULTS

An important aspect of the work described here was the emphasis on a realistic assessment of performance of the developed systems and algorithms. This assessment consisted of two stages. First, the architecture was tested by integrating an audio device, a smart card beacon concentrator device, four of the video devices described earlier, and a separate four-camera video device. This work was conducted in the Paris Metro (Gare de Lyon) and successfully demonstrated the communications mechanisms and protocol. Second, a major deployment of the system took place in the Liverpool St. station. This is the fourth busiest station in the London Underground network, in terms of the number of passengers. It deals with commuter traffic to/from one of the biggest financial centers in Europe and connects with the main railways and buses. There are more than seventy cameras in this station covering approximately 80% of its total area. Stringent regulations meant that it was only possible to set up the system for video detection.

TABLE III
PERFORMANCE FIGURES FOR STATIONARY OBJECT DETECTION

Test	Detection percentage		
	True positive	False positive	True negative
Normal Occlusion	97.9%	0%	2.1%
Occlusion with background colour	87.5%	0%	12.5%
Detection within 2min ± 5sec	100%	0%	0%
Position updating in 3 Sec.	100%	4%	0%

TABLE IV
OVERCROWDING AND CONGESTION PERFORMANCE

Event	Detection percentage		
	True positive	False positive	True negative
Overcrowding	95.6%	4.0%	0.4%
Congestion	98.5%	0.3%	1.2%

A. Verification

This refers to a preliminary evaluation of the detection capability of the system with video recordings made on site (London, Paris, Milan). The performance of the algorithms evaluated against manually obtained ground truths for the data made available at the time by transport operators and indicative of their highest concerns. Unless otherwise specified, verification was carried out with a minimum number of samples m to satisfy an uncertainty of true performance ΔP of 7% and a confidence level α of 95%

$$m \geq \frac{(1+\alpha)^2}{4\Delta P^2} \approx 200 \text{ samples.} \tag{16}$$

For motion direction, an event is defined as a pedestrian entering and leaving the camera's field of view. Table II shows the performance-assessment figures. On the detection of stationary areas, tests were conducted in the following conditions.

1) Normal occlusion. Complete occlusion of the stationary region by moving or standing pedestrians for at least 2 min (the specified stationary period to be detected).
2) Occlusion with the same color as the background. Occlusion with moving pedestrians of grey levels similar to the background shades (only eight cases were considered due to lack of recorded data).
3) Pose and position variations. Movement of limbs and torso or shift in standing location with at least 1% overlapping with original position, with an updating period of 3 s.

An event is defined as a pedestrian standing within the AOI for more than the allowed period (within 2 min ± 5 s). Table III shows the performance figures. Tests to assess performance on detection of overcrowding and congestion were conducted using 3 h of video recorded in a ticket hall (Fig. 5) and a corridor (Fig. 14) and the results are shown in Table IV. Overcrowding is detected instantaneously but, being a global measure, false alarms are generated mainly due to short bursts of loosely distributed crowds. Congestion, based on motion, overcomes some of these problems at the expense of some delay in detection (typically less then 5 s).

B. Operational Performance

The results summarized above give an indication of the detection ability of the vision processes. However, they necessarily correspond to limited conditions and do not address the important issues of usability and reliability over longer periods of time and under operational conditions. To assess such performance in detail, Ipsotek developed and tested a system known as the intelligent pedestrian surveillance (IPS) system with similar detection functionality plus the following additional processes:

1) train presence;
2) significant change;
3) loitering.

1) Train Presence: The detection of presence of a train in the area under surveillance is used to inhibit detection of other events (e.g., overcrowding, intrusion near the edge of the platform) as these situations are normal on arrival/departure of a train. This operates in a similar way to overcrowding detection (Section II-E1) except that prior operator knowledge is used to position its AOI on the tracks.

2) Significant Change: This refers to the detection of movement/presence in an area where it is known that such conditions are extremely unlikely (e.g., above pedestrian heads or train roofs). This event generally arises in underground stations due to sudden changes in lighting conditions (e.g., lamp failure) or camera movement. Detection operates in a similar way to that of overcrowding detection, a low-priority alarm is generated and, more importantly, the background-estimation process is restarted. This feature is particularly useful to provide continuous unattended operation.

3) Loitering: This refers to the sustained presence of one or more people (over a given time limit) in an area. For example, in a corridor, this could indicate unauthorized activity such as selling/begging; near ticket machines, it might indicate the presence of ticket touts (people that illegally resell used tickets), and at the end of the platforms it might be indicative of people considering committing suicide. In relatively uncluttered environments, it is possible to detect loitering by localizing and tracking individuals (e.g., see [10]). In busy public environments, such approaches are still not sufficiently reliable. A full discussion of the method to detect loitering is beyond the space constraints of this paper and will be reported elsewhere. The approach is based on maintaining positional appearance models (luminance and motion) for subregions in the loitering AOI. Weights inversely proportional to motion magnitudes are used, as loitering is characterized by small speeds mixed with periods of stationarity. The consistent presence, over a user-defined period of time, of an appearance pattern (measured through a correlation value)

triggers the detection of loitering. As the method is based on appearance correlation, it is able to maintain detection of people that temporarily leave and then come back into the AOI.

A system for 14 cameras was installed in the Liverpool St. station (London) and evaluated under operational conditions (in a control room) by control-room operators. These tests did not involve staging events or continuous reconfiguration of settings. The following summarizes their main findings.

Detection performance was measured by comparing the incidents detected and logged by the system against the digital recording associated with such incidents. Each logged event was classified by operators as either true or false. Thus, what was measured was the percentage of true and false positives. The system was found to detect true overcrowding/congestion incidents 98% of the time. Another AOI to the operators was the detection of intrusion into closed areas of the station at night. This was found to be detected correctly 81% of the time (but it is estimated that most of the false alarms were associated with a problem of lack of synchronization between the system's and the station CCTV's clocks, where the system would still be set to detect intrusion once the area had been opened to passengers). Detection of abandoned packages was measured to be done successfully 87% of the time. This is regarded by operators as a high success rate, as it has to be compared with a conventional situation where at best only around 10% of the cameras are monitored by staff at any given time. The system performed well in detecting instances of loitering at a level of 82%.

A survey on the usability and likely operational impact of the system was then conducted by the transport operator among managers (35 out of 37, 95%, of which believed that the system can be useful in the Underground) and among the station staff. Staff expressed satisfaction with image quality and with the ease of use of the touch-based operator interface once they have been trained to use it (typically a couple of hours). Most of the staff felt that the system makes the station (85%) and staff (100%) safer, detection of intrusion into closed areas (87%), dealing with overcrowding (58%), that the system is an improvement for the station (100%) and that systems of this type should be installed at most stations (100%). Interestingly, the potential effect on reducing station closures due to abandoned packages was rated lower (43%), in contrast to the objectively measured true/false positives (87%/13%). This discrepancy between perception and detection ability needs investigating further. The ability to detect people near the edge of the platforms was rated lower (28%), but this was traced to the problem of having cameras that are not well-positioned for this task (i.e., aligned with the platform edge) so that occlusion resulting from perspective effects generates an unacceptable number of false alarms. This illustrates that the introduction of CCTV monitoring support is likely to be incremental. With added confidence in automatic detection, it becomes possible to consider installation of cameras for specific purposes, which conventionally is limited by the lack of human resources to monitor such cameras.

V. Conclusion

This paper has described part of the work of a major European effort to improve personal security and safety in public transport systems through enhanced surveillance systems. Although the description focused on technical aspects, the work involved a multidisciplinary approach essential to understanding the context in which surveillance takes place. First, to define the expectations and limitations of those who are surveyed and, second, to integrate solutions within the working practices of human operators and managers of monitoring control rooms. Such types of informed integration is vital within a framework of ambient intelligence, where the main purpose is to provide transparent support in environments where human activity takes place and, hence, where humans are the primary subject for the provision of an enhanced environment. Because of the important role that vision plays in these surveillance systems, this paper first described in detail one of the vision modules used in a PRISMATICA system. The new algorithms presented have been shown to be capable of dealing with the type of image complexity present in metropolitan railway environments, over long periods of time with detection rates exceeding 80%. The particular strengths of these algorithms are the ability to integrate motion as an integral part of background/foreground detection, a method for perspective self-calibration derived from motion and robustness to occlusion, camera shake, and illumination changes. As far as the authors know, this type of long-term evaluation under full operational conditions has not been reported elsewhere. Then, the paper described a generalized surveillance architecture that reflects the distributed nature of the monitoring task and that allows for distributed detection processes, not only dealing with visual processing but also with devices such as acoustic signature detection and mobile smart cards, actuators and a range of possible sensors. The framework was developed reflecting existing management procedures (such as the deployment of ground staff that have specific tasks and report to a central control room when necessary). The PRISMATICA system was tested in London and in Paris. A system with comparable functionality (IPS), was then extensively and successfully tested in one of the busiest underground stations in London (Liverpool St. station). Current work is aimed at improving even further the performance of the vision module, e.g., by exploiting color information (when available), considering how to measure finer behavioral measurements that depend for example on posture and gesture, applying learning processes (such as hidden Markov models) to the learning of what constitutes normal and abnormal behavior, frameworks for fusing diverse types of date and also to coordinate surveillance from one camera to another. There is also still much to be done on incorporating expert-domain knowledge into the automatic monitoring task (e.g., how situations change at the time of day when school children return home). On system aspects, the main efforts are concentrated on finding predictable scalable real-time system architectures [53] that can be applied to widely distributed surveillance systems (geographically and from the point of view of the number of people in charge of monitoring).

Acknowledgment

The authors are grateful to London Underground, the Rome Public Transport Authority (ATAC), the Paris Metro and Newcastle International Airport for providing access to their sites

and staff. They also thank the anonymous reviewers who provided valuable comments to improve this paper.

REFERENCES

[1] S. A. Velastin, B. Lo, J. Sun, L. Khoudour, and M. A. Vicencio-Silva, "Multi-sensory tools to improve personal security in public transport networks," in *Proc. Workshop Ambient Intell. AI*IA 2003—8th Nat. Congress Ital. Assoc. Artif. Intell.*, Pisa, Italy, Sep. 23, 2003, pp. 12–23.

[2] S. A. Velastin, M. A. Vicencio-Silva, B. Lo, and L. Khoudour, "A distributed surveillance system for improving security in public transport networks," *Measure. Contr.*, vol. 35, no. 8, pp. 209–213, 2002.

[3] L. Marcenaro, F. Oberti, G. L. Foresti, and C. S. Regazzoni, "Distributed architectures and logical-task decomposition in multimedia surveillance systems," *Proc. IEEE*, vol. 89, no. 10, pp. 1419–1440, Oct. 2001.

[4] B. A. Boghossian and S. A. Velastin, "Image processing system for pedestrian monitoring using neural classification of normal motion patterns," *Measure. Contr.*, vol. 32, no. 9, pp. 261–264, Sep. 1999.

[5] B. A. Boghossian, "Motion-based image processing," Ph.D. dissertation, Department of Electronic Engineering, King's College London, University of London, London, UK, 2000.

[6] C. Sacchi and C. Regazzoni, "A distributed surveillance system for detection of abandoned objects in unmanned railway environments," *IEEE Trans. Veh. Technol.*, vol. 49, no. 5, pp. 2013–2026, Sep. 2000.

[7] L. M. Fuentes and S. A. Velastin, "People tracking in surveillance applications," presented at the 2nd IEEE Int. Workshop Performance Eval. Tracking Surveillance, Kauai, HI, 2001.

[8] ——, "From tracking to advanced surveillance," presented at the IEEE Int. Conf. Image Process., Barcelona, Spain, Sep. 14–17, 2003.

[9] ——, "Assessment of image processing as a means of improving personal security in public transport," in *Video-Based Surveillance Systems, Computer Vision and Distributed Processing*, P. Remagnino, G. A. Jones, N. Paragios, and C. S. Regazzoni, Eds. Norwell, MA: Kluwer, 2001, ch. 13, pp. 159–166.

[10] N. T. Siebel and S. Maybank, "Fusion of multiple tracking algorithms for robust people tracking," in *Proc. Eur. Conf. Comput. Vision*, 2002, pp. 373–382.

[11] B. Lo and S. A. Velastin, "Automatic congestion detection system for underground platforms," presented at the IEEE Int. Symp. Intell. Multimedia, Video, Speech Process., Hong-Kong, SAR, May 2–4, 2001.

[12] B. A. Boghossian and S. A. Velastin, "Image processing system for pedestrian monitoring using neural classification of normal motion patterns," *Measure. Contr.*, vol. 32, no. 9, pp. 261–264, 1999.

[13] N. Rota and M. Thonnat, "Video sequence interpretation for visual surveillance," in *Proc. 3rd IEEE Int. Workshop Visual Surveillance*, 2000, pp. 59–68.

[14] B. Georis, X. Desurmont, D. Demaret, J. F. Delaigle, and B. Macq, "IP-Distributed computer-aided video-surveillance system," presented at the IEE Intell. Distributed Surveillance Syst. Workshop, London, UK, Feb. 26th, 2003.

[15] (1999) *The Kluwer International Series in Engineering and Computer Science. VLSI, Computer Architecture and Digital Signal Processing*

[16] A. Langlais, "Deliverable D2: User needs analysis," in *CROMATICA TR-1016*, Nov. 1996. CEC Framework IV Telematics Applications Programme.

[17] R. T. Collins, A. J. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," *Proc. IEEE*, vol. 89, no. 10, pp. 1456–1477, Oct. 2001.

[18] C. Wern, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, July 1997.

[19] A. E. C. Pece, "Generative model-based tracking by cluster analysis of image differences," *Robotics Autonomous Syst. J.*, vol. 39, no. 3–4, pp. 181–194, 2002.

[20] Y. Raja, S. J. McKenna, and S. Gong, "Segmentation and tracking using color mixture models," in *Proc. Asian Conf. Comput. Vision*, 1998, pp. 607–614.

[21] J. W. Davis and A. Bobick, "The representation and recognition of action using temporal templates," *Proc. Comput. Vision Pattern Recogn.*, pp. 928–934, 1997.

[22] B. K. P. Horn and B. G. Schunk, "Determining optical flow," *Artif. Intell.*, vol. 17, pp. 185–203, 1981.

[23] M. Yeasin, "Optical flow in log-mapped image plane—A new approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 125–131, Jan. 2002.

[24] A. C. Davies, J. H. Yin, and S. A. Velastin, "Crowd monitoring using image processing," *IEE Electron. Commun. Eng. J.*, vol. 7, no. 1, pp. 37–47, 1995.

[25] S. A. Velastin, A. C. Davies, J. H. Yin, M. A. Vicencio-Silva, R. E. Allsop, and A. Penn, "Analysis of crowd movements and densities in built-up environments using image processing," in *IEE Coll. Image Process. Transport Applicat.*, vol. 236, London, UK, 1993, pp. 8/1–8/6.

[26] H. M. Hang, Y. M. Chou, and S. C. Cheng, "Motion estimation for video coding standards," *J. VLSI Signal Process. Syst.*, vol. 17, no. 2–3, pp. 113–136, 1997.

[27] M. Coimbra, M. Davies, and S. A. Velastin, "Pedestrian detection using MPEG-2 motion vectors," presented at the 4th Eur. Workshop Image Anal. Multimedia Interactive Services, London, UK, Apr. 9–11, 2003.

[28] S. Bouchafa, D. Aubert, and S. Bouzar, "Crowd motion estimation and motionless detection in subway corridors by image processing," in *IEEE Conf. Intell. Transport. Syst.*, 1997, pp. 332–337.

[29] J. H. Yin, "Automation of crowd data-acquisition and monitoring in confined areas using image processing," Ph.D. dissertation, Department of Electronic Engineering, King's College London, University of London, Sep. 1996.

[30] M. Coimbra and M. Davies, "A numerical comparison of compressed domain approximations to optical flow," presented at the 5th Int. Workshop Image Anal. Multimedia Interactive Services, Lisbon, Spain, 2004.

[31] S. Jamkar, S. Belhe, S. Dravid, and M. S. Sutaone, "A comparison of block-matching search algorithms in motion estimation," in *Proc. 15th Int. Conf. Comput. Commun.*, 2002, pp. 730–739.

[32] J. L. Chen and P. Y. Chin, "An efficient gray search algorithm for the estimation of motion vectors," *IEEE Trans. Syst. Man. Cybern. C, Appl. Rev.*, vol. 31, no. 2, pp. 242–248, May.

[33] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for realtime tracking with shadow detection," in *Proc. 2nd Eur. Workshop Adv. Video-Based Surveillance Syst.*, Sep. 2001, pp. 149–158.

[34] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," in *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, Aug. 2000, pp. 747–57.

[35] C. Ridder, O. Munkelt, and H. Kirchner, "Adaptive background estimation and foreground detection using Kalman-filtering," in *Proc. Int. Conf. Recent Adv. Mechatron.*, 1995, pp. 193–199.

[36] J. R. Renno, P. Remagnino, and G. A. Jones, "Learning surveillance tracking models for the self-calibrated ground plane," *Acta Automatica Sinica*, vol. 29, no. 3, pp. 381–392, 2003.

[37] C. S. Regazzoni, A. Tesei, and V. Murino, "A real-time vision system for crowding monitoring," in *Proc. Int. Conf. Ind. Electron.*, vol. 3, pp. 1860–1864.

[38] A. J. Schofield, T. J. Stonham, and P. A. Mehta, "A RAM-based neural network approach to people counting," presented at the IEE Image Process. Applicat. Conf. Pub., July 4–6, 1995.

[39] T. Coianiz, M. Boninsegna, and B. Caprile, "A fuzzy classifier for visual crowding estimates," in *IEEE Int. Conf. on Neural Networks, 1996*, vol. 2, Jun. 3–6, 1996, pp. 1174–1178.

[40] C. Ottonello, M. Peri, C. Regazzoni, and A. Tesei, "Integration of multisensor data for overcrowding estimation," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, vol. 1, 1992, pp. 791–796.

[41] C. S. Regazzoni and A. Tesei, "Density evaluation and tracking of multiple objects from image sequences," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, 1994, pp. 545–549.

[42] A. N. Marana, S. A. Velastin, L. da F. Costa, and R. A. Lotufo, "Automatic estimation of crowd occupancy using texture and NN classification," *Safety Sci.*, vol. 28, no. 3, pp. 165–175, 1998.

[43] S. A. Velastin, J. H. Yin, A. C. Davies, M. A. Vicencio-Silva, R. E. Allsop, and A. Penn, "Automated measurement of crowd density and motion using image processing," in *Proc. 7th Int. Conf. Road Traffic Monitoring Contr.*, London, UK, 1994, pp. 127–132.

[44] M. Takatoo, C. Onuma, and Y. Kobayashi, "Detection of objects including persons using image processing," in *Proc. 13th IEEE Int. Conf. Pattern Recogn.*, vol. 3, pp. 466–472.

[45] C. C. Heath, P. K. Luff, and M. Sanchez-Svensson, "Overseeing organization," *Brit. J. Sociol.*, vol. 53, no. 2, pp. 181–203, 2002.

[46] D. Levine and S. Mungee, "The design and performance of real-time object request brokers," *Comput. Commun.*, vol. 21, no. 4, Apr. 1998.

[47] A. Gokhale and D. C. Schmidt, "Techniques for optimizing CORBA middleware for distributed embedded systems," presented at the INFOCOM, New York, Mar. 21–25th, 1999.

[48] D. Levine and S. Flores-Gaitan, "Measuring OS support for real-time CORBA ORBs," presented at the 4th IEEE Int. Workshop Object-Oriented Real-Time Dependable Syst., Santa Barbara, CA, Jan. 27–29, 1999.
[49] N. Wang, D. C. Schmidt, and D. Levine, "Optimizing the CORBA component model for high-performance and real-time applications," presented at the IFIP/ACM Middleware Conf., New York, Apr. 3–7, 2000.
[50] S. A. Velastin, B. Lo, and J. Sun, "A flexible communications protocol for a distributed surveillance system," *J. Netw. Comput. Applicat.*, vol. 27/4, pp. 221–253.
[51] P. K. Luff, C. C. Heath, and M. Jirotka, "Surveying the scene: Technologies for everyday awareness and monitoring in control rooms," *Interacting Comput.*, vol. 13, pp. 193–228, 2000.
[52] B. P. L. Lo, J. Sun, and S. A. Velastin, "Fusing visual and audio information in a distributed intelligent surveillance system for public transport systems," *Acta Automatica Sinica*, vol. 29, no. 3, pp. 393–407, 2003.
[53] M. Valera and S. A. Velastin, "Real-time architecture for large distributed surveillance systems," in *Proc. IEE Symp. Intell. Distributed Surveillance Syst.*, London, UK, 2004, pp. 41–45.

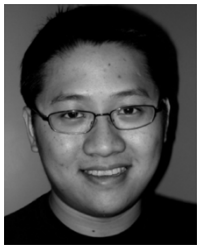
Sergio A. Velastin (M'90) received the B.Sc. degree in electronics, the M.Sc. degree in digital image processing, and the Ph.D. from the University of Manchester Institute of Science and Technology (UMIST), Manchester, UK, in 1978, 1979, and 1982, respectively.

He is currently a Reader (Associate Professor) at the Digital Imaging Research Centre, School of Computing and Information Systems, Kingston University, Kingston, UK. He was a Technical Coordinator of the EU project PRISMATICA. His research interests include computer vision for pedestrian monitoring and personal security, as well as distributed visual surveillance systems.

Prof. Velastin is a Member of the IEE and the British Machine Vision Association.

Boghos A. Boghossian (S'97–M'01) received the B.Sc. degree in electronics engineering and communications from Baghdad University, Baghdad, Iraq, in 1995 and the M.Sc. and Ph.D. degrees in electronics engineering from King's College London, University of London, London, UK, in 1997 and 2001, respectively.

He is currently the Technology Director of IPSOTEK Ltd, London, UK, and also a Chief Design Engineer with Sollatek (UK) Ltd.



Benny Ping Lai Lo received the B.Sc. degree in electrical engineering from the University of British Columbia, Vancouver, BC, Canada, in 1995 and the M.Sc. degree with distinction in electronic research from King's College London, University of London, London, UK, in 2000. He is currently pursuing the Ph.D. degree at Warwick University, Coventry, UK.

He worked as a Project Engineer with Cybermation System Inc., Canada, for two years, and later joined the Mass Transit Railway Corporation, Hong Kong SAR, where he was a Design Officer of infrastructure design—operating control system until 1999. Later, he was a Research Associate at King's College London until 2001 and a Senior Researcher at Kingston University, Kingston, UK, until 2003 on two EU-funded projects: ADVISOR and PRISMATICA. He is currently working as a research associate in Imperial College London on a DTI-funded project, UbiMon.

Jie Sun received the M.Sc. degree in control engineering from the Harbin Institute of Technology, Harbin, China, in 1999.

He was a Computer Network System Engineer with Legend Computer Systems, Beijing, China, for two years. He worked in the European Union-funded project PRISMATICA since its inception, was responsible for developing the user interface and the back-end database for the MIPSA system, and for coordinating the work of a multinational team on communication protocols and interfacing. His current research interests are in network-based surveillance systems and artificial intelligent systems.

Maria Alicia Vicencio-Silva received the B.Sc. degree in electronic engineering from the Universidad Católica de Valparaíso, Valparaíso, Chile, in 1975, the M.Sc. degree in computers and digital systems from the Universidad Técnica Federico Santa María, Valparaíso, Chile, in 1983, and the Ph.D. degree from the University of Manchester Institute of Science and Technology (UMIST), Manchester, UK, in 1989.

From 1973 to 1980, she was a Senior Lecturer of electronic engineering at the Universidad del Norte, Chile, and between 1983 and 1986 was an Assistant Professor of informatic engineering at the Universidad Austral, Chile. Since 1989, she has been a Senior Researcher at the Centre for Transport Studies, University College London, London, UK. Her main research interests are in social aspects of engineering practice, social inclusion, and public transport policies.