An Autonomous Vehicle for Video Surveillance of Indoor Environments

Christian Micheloni, *Member, IEEE*, Gian Luca Foresti, *Senior Member, IEEE*,
Claudio Piciarelli, and Luigi Cinque, *Senior Member, IEEE*

***Abstract*—In this paper, the problem of the surveillance and the security of indoor environments is addressed through the development of an autonomous surveillance vehicle (ASV). The ASV has been designed to perform, in addition to the classical robotic tasks (e.g., navigation and obstacle avoiding), the tracking of objects (e.g., persons) moving in indoor environments. The selection of the target object to be tracked can be decided by a remote operator or autonomously by the ASV itself in the case that a suspicious behavior has been detected (e.g., a person entering a forbidden area, etc.). The tracking procedure allows the ASV to maintain the interesting objects in the center of the image and in specific cases to localize particular parts of the object (e.g., face of a person, etc.) in order to recognize it. Experimental results have been performed on different real scenarios where no objects move inside the monitored scene and where a group of people move in a hallway.**

***Index Terms*—Autonomous vehicle, face detection, object tracking, video surveillance.**

I. Introduction

THE RESEARCH done over the past few years in the field of mobile vision has been on many fronts: aircraft [1], [2], autonomous underwater vehicles [3]–[5], and autonomous guidance vehicles (AGVs) moving on the ground [6], [7]. Certainly, the work done on guidance of ground vehicle represents a great part of the research in the context of mobile navigation. In particular, two streams of research can be distinguished in this field on the basis of the context in which the systems are involved: outdoor and indoor navigation.

Regarding the navigation for outdoor vehicles, we could count many works made for the guidance as represented by the NAVLAB system [9], [10] for establishing the position and the boundaries of the roads, the work on vision-guided road-following for "Autobans" [11], [12], the Prometheus system [13], and the navigation on unstructured environments [14].

More interest has pointed toward indoor navigation, and in the last few years, much work has been done. From the first systems, like those proposed by Giralt *et al.* in 1979 [15] and by Moravec in [16], many systems have been developed. For further details on indoor navigation systems, see [17].

Recently, some researchers analyzed the problem of employing AGVs for surveillance purposes in both outdoor and indoor environments. In [18], Lipton and co-workers employed an airborne platform and ground vehicles to develop a multicamera system to monitor activities in cluttered outdoor environments using a distribution network of fixed and mobile sensors.

The video surveillance has certainly played a paramount role in the research of the last decade in which many systems have been proposed [19]–[24]. The application of video surveillance has a high range of purposes, from traffic monitoring [25], [26] to human activity understanding [27], [28]. Video surveillance applications often pay attention to a wide area, so different kinds of cameras are generally used, e.g., fixed cameras [21], [23], [24], omnidirectional cameras [8], [29], [30], and pan and tilt cameras [18], [31]–[34].

The use of these kinds of cameras requires that the number and the placement of the sensors must be fixed in advance to ensure an adequate monitoring coverage of the area of interest. In the context of visual-based surveillance applications, there are many conditions for which deciding *a priori* the placement of sensors puts limits on system performance or significantly increases the costs due to extensive use of sensors. We refer, for example, to those cases in which an alarm situation can occur with the same probability in any area of the monitored environment or to those situations that require the tracking of mobile objects in wide areas (e.g., a vehicle moving on a road or a people moving in a building). In these situations, especially in the context of indoor environments, the employment of mobile robots equipped with specific visual sensors for surveillance purposes can become an important issue. The integration of a mobile robot in a visual-based surveillance system can allow the coverage of all types of environments, can extend the perceiving capabilities of the system (e.g., acquire images of higher quality by reducing the distance from the camera to the target), and can furnish to a remote operator an augmented reality of the observed scene (a target can be observed from different points of view according to its position or trajectory).

Manuscript received July 11, 2003; revised January 20, 2005, February 2, 2006, and April 19, 2006. This work was supported in part by the Italian Ministry of University and Scientific Research within the framework of the project "Ambient Intelligence: Event analysis, sensor reconfiguration, and multimodal interfaces" (2006–2008). The review of this paper was coordinated by Dr. M. Trivedi.

C. Micheloni, G. L. Foresti, and C. Piciarelli are with the Department of Mathematics and Computer Science (DIMI), University of Udine, 33100 Udine, Italy (e-mail: michelon@dimi.uniud.it; foresti@dimi.uniud.it; piccia@dimi.uniud.it).

L. Cinque is with the Department of Computer Science (DSI), University of Rome "La Sapienza," 00198 Rome, Italy (e-mail: cinque@dsi.uniroma1.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2007.891478

In this paper, we have pointed our attention to the problem of the surveillance and the security of indoor environments, and to achieve these purposes, an autonomous surveillance vehicle (ASV) has been designed and developed. The ASV is able to perform, in addition to the classical robotic tasks (e.g., navigation in an indoor environment by avoiding obstacles), the tracking of mobile objects (e.g., persons). The selection of the target to be tracked can be decided by a remote operator or autonomously by the higher level modules of the system when a suspicious event is detected. Since the focus of the current work is not on target selection, the reader can assume without loss of generality that the target is manually selected. The tracking procedure allows the system to maintain the interesting objects in the center of the image and, in specific cases, to localize particular parts of the object (e.g., face of a person, etc.) in order to recognize it.

The use of video cameras placed on a mobile vehicle significantly increases the complexity of both detection and tracking of mobile objects in the scene. The detection requires the development of appropriate methods that are able to take into account the ego-motion of the camera in order to apply change detection techniques. While some benefits can result from the use of stereo systems [53]–[55], this paper is concerned about monocular systems, in which frame-by-frame-based techniques are generally applied to estimate the displacement of two consecutive frames due to the motion of the camera [14], [32], [35], [56], [57], [59]. An image compensation is then applied in such a way that static objects exactly overlap the same objects in the previous frame. A frame-by-frame image subtraction is then applied, and the classical object detection techniques can be used as in the case of fixed cameras [23], [36]. However, these techniques demonstrate some limits in presence of outliers (i.e., object moving in the scene) [37] or when few features are extracted in the image sequence and cannot be applied in the case of a camera mounted on a mobile vehicle. The proposed ASV is able to detect moving objects by means of a direct method [3], [38], [39], [60] by computing the affine transformation for the alignment of the two consecutive frames.

The tracking of mobile objects in the monitored scene requires the ASV to move autonomously in such a way that the tracked object appears in the center of the current image. This constraint is necessary in order to simplify the work of the higher level modules in charge to classify the detected object, understand its behavior, or localize and recognize specific parts. To this end, a Kalman-filter approach has been employed to estimate in real-time the appropriate motion parameters of the ASV.

II. System Description

The proposed system has been build for performing two main activities: vision and control (Fig. 1). The vision process aims to identify all objects moving in the scene and to verify which one must be considered the most important to monitor. To achieve this objective, a motion detection module, a tracking module, and face recognition module are involved.

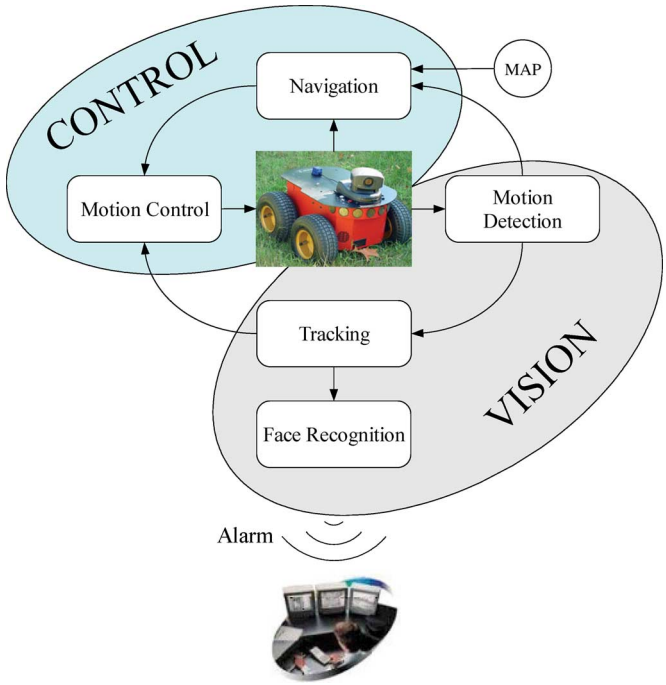


Fig. 1. System architecture.

The motion detection module aims to solve the problem of the detection of moving objects in the scene. This problem has been addressed by applying an image differencing technique after the alignment of two consecutive frames $I(x,t)$ and $I(x,t+1)$. The thresholded output [36] of this process is a binary image $\mathbf{B}(x,y)$ representing the pixels belonging to moving objects. Then, $\mathbf{B}(x,y)$ is analyzed to group pixels belonging to the same object according to a bounding ellipse.

Once the objects have been identified, a tracking module is applied to maintain track of their movements. By maintaining the objects inside the field of view of the camera, the ASV is able to perform face detection and recognition tasks. This is allowed by a face-detection algorithm that, by receiving the bounding ellipses of mobile objects, is able to detect the face pattern of the monitored person. Finally, the pattern is given as input to a face recognition module for the analysis.

The results performed by the tracking module are also used to supply the position of the object to the motion control module. This process of the ASV is performed by two separate modules: the navigation and the motion control modules. Through these modules, the system aims to identify the proper actions that must be sent to the ASV. The navigation task fetches from the ASV the distances of possible obstacles performing a first computation about the self-localization inside a map. To build the map, two main approaches can be taken into account: map-based [40], [41] and map-building-based [42], [43]. The map-based technique works on topological maps previously defined instead, and the map-building approach involves sensors to construct their own topological maps. The self-localization allows the avoidance of obstacles by computing the metrical distances and the movements of the ASV. Since the control

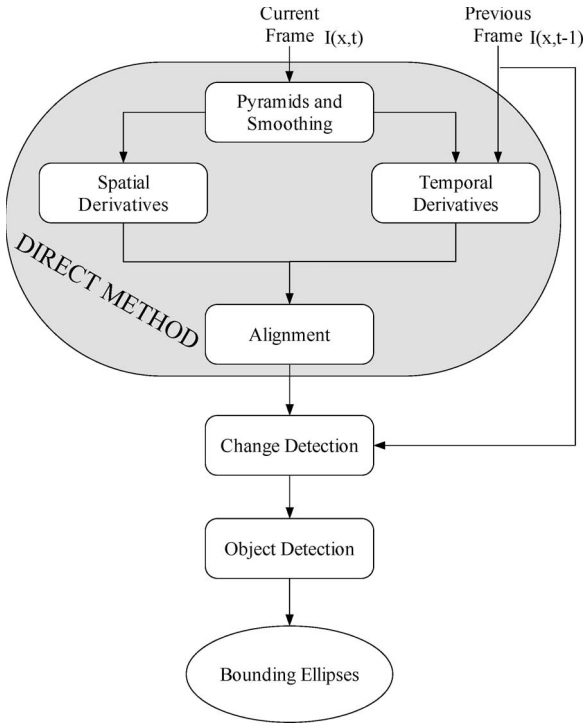


Fig. 2. General architecture of the motion detection module with direct methods and change detection.

activity is not the objective of this paper, see [17] in order to obtain further details regarding techniques for mobile robot navigation.

In the following section, the vision activity will be described in detail by presenting the solution adopted to detect the motion, to track the moving objects, and, finally, to detect and to recognize people moving into indoor environments.

III. Motion Detection

The main activity of this module is the detection of mobile objects inside the scene: a task that cannot be easily solved if a mobile camera is used. The problem consists of the identification of the motion due to the camera actions and the motion of moving objects. The proposed solution, as shown in Fig. 2, is given by applying a direct method [3], [38], [60] to compute the affine transformation for the alignment of two consecutive frames.

From the equation of the optical flow given by Horn and Schunk in [44]

$$I_x \nu_x + I_y \nu_y + I_{\mathrm{t}} = 0 \tag{1}$$

where I_x, I_y, and I_{t} are the spatial and temporal derivatives of the brightness intensity, and ν_x and ν_y are the components of the displacement, and by considering the equation of the affine displacements, we obtain the equation of the affine flow

$$I_x(a_{11}x + a_{12}y + a_{13}) + I_y(a_{21}x + a_{22}y + a_{23}) + I_{\mathrm{t}} \tag{2}$$

and it is possible to define a linear system by considering the (2) for each pixel of the image as follows:

$$\underbrace{\begin{pmatrix} I_{x,1}x_1 & I_{x,1}y_1 & I_{x,1} & I_{y,1}x_1 & I_{y,1}y_1 & I_{y,1} \\ I_{x,2}x_2 & I_{x,2}y_2 & I_{x,2} & I_{y,2}x_2 & I_{y,2}y_2 & I_{y,2} \\ \vdots & & & & & \\ I_{x,n}x_n & I_{x,n}y_n & I_{x,n} & I_{y,n}x_n & I_{y,n}y_n & I_{y,n} \end{pmatrix}}_{\mathbf{A}} \times \underbrace{\begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \end{pmatrix}}_{x} = \underbrace{\begin{pmatrix} -I_{t,1} \\ -I_{t,2} \\ \vdots \\ -I_{t,n} \end{pmatrix}}_{b}. \tag{3}$$

Equation (3) can be written in closed form as $\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{b}$ to compute the parameters $(a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23})$ of the affine transform. Not all the pixels supply enough information to the computation of these parameters; this is the case of pixels in which both spatial derivatives are equal to zero. Therefore, the proposed method takes into account only those pixels whose gradient is greater than a threshold T_{PG}

$$\nabla I(x, y) \geq T_{\mathrm{PG}}. \tag{4}$$

The value of the threshold T_{PG} is automatically set by using a thresholding technique [36] that allows a reduction of 20% the number of pixel involved in the computation of the affine transform. This implies both a lighter computational load and a smaller error in the solution of (3).

Since the equation of the optical flow holds only in the neighborhood of the considered point, the affine flow can be computed accurately only if the displacement between two images is minimal. To overcome this constraint we have adopted two techniques: a) iterative alignment and b) pyramids alignment.

A. Iterative Alignment

The common solution to the problem of the alignment in context of wide displacements is adopting an iterative method. For example, the well-known Newton–Raphson scheme can be employed, but the efficiency of the iteration method does not guarantee a real-time processing. Therefore, two heuristics have been studied to speed up the iteration. The first consists in the initial Cholesky factorization of the matrix $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ in order to avoid an expensive factorization at each iteration, which is not needed since such a matrix does not change during iterations.

The second is represented by the initialization of the iterative process. In particular, by noting that consecutive frames have a high temporal correlation, the parameters of the affine transform are initialized to the values computed in the previous frame. Therefore, the system starts the iterative computation from an initial estimation of the affine motion that is represented by the affine transform computed at the previous frame.

In order to reduce the errors in the parameters computation, the initialization is periodically set to zero (i.e., in the experiments the period adopted is of about 25 frames). The proposed algorithm with the adopted heuristics is therefore the following.

Algorithm 1 Iterative Alignment. The alignment process proceeds until $\Delta\mathbf{a}$ is sufficiently smaller than a threshold Thr.

$\mathbf{a}[\mathrm{t}] = \mathbf{a}[\mathrm{t}-1]$
$\mathbf{L} = \mathrm{Cholesky}(\mathbf{A}^{\mathrm{T}}\mathbf{A})$
Repeat
  Solve $(\mathbf{L}\mathbf{L}^{\mathrm{T}}\Delta\mathbf{a} = \mathbf{A}^{\mathrm{T}}\mathbf{b})$
  $\mathbf{a}[\mathrm{t}] = \mathbf{a}[\mathrm{t}] + \Delta\mathbf{a}$
  $\mathrm{I}(\mathrm{t}) = \mathrm{Transform}(\mathbf{a}, I, t)$
Until $(\Delta\mathbf{a} < \mathrm{Thr})$

B. Pyramidal Alignment

Since the iterative method, which has been adopted in context of wide displacement among the frames, requires a high number of iterations, we propose an alignment method based on multiresolution images. In particular, a pyramid of images is represented by different levels, each characterized by the original image at a different resolution. The scaling factor of the level follows a logarithmic law so that the number of levels is defined by the logarithm of the presumed displacement.

The proposed alignment method starts to compute the parameters of the affine transform at the highest level (i.e., the level corresponding to the image with lower resolution) and ends at the lowest one. At each level, a new iterative process for the alignment computation is started by initializing the parameters to the value obtained at the previous level. Doing this, due to the scaling factor, we must take care to multiply the parameters a_{13} and a_{23} (translations) by 2 while not changing the other parameters.

When the displacement is small, the pyramids approach is unfavorable since the standard iterative method would converge to the solution in fewer steps. Therefore, the number of pyramid levels is heuristically estimated at each time instant by taking the logarithm of the displacement computed for the previous frame.

A further problem during the alignment phase is represented by those pixels belonging to moving objects that are involved in the computation of the affine transform. We consider these pixels as outliers since they violate the model of the affine flow and are not described by a Gaussian model of noise like pixels belonging to static objects. In order to give minor weight to the equation of the linear system described in (3) corresponding to the outliers, we have employed a robust estimator. The choice of the iteratively reweighted least square (IRLS) [45] as robust estimator allows us to maintain the structure of the described method. With this estimator, the linear system $\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{b}$ becomes $\mathbf{A}^{\mathrm{T}}\mathbf{W}\mathbf{A}\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{W}\mathbf{b}$, where $\mathbf{W} = \mathrm{diag}(w_1, \ldots, w_n)$, and the weights w_i are computed by the estimator as follows:

$$w_i = w(x_i) = \frac{\rho'(x_i)}{2x_i} \tag{5}$$

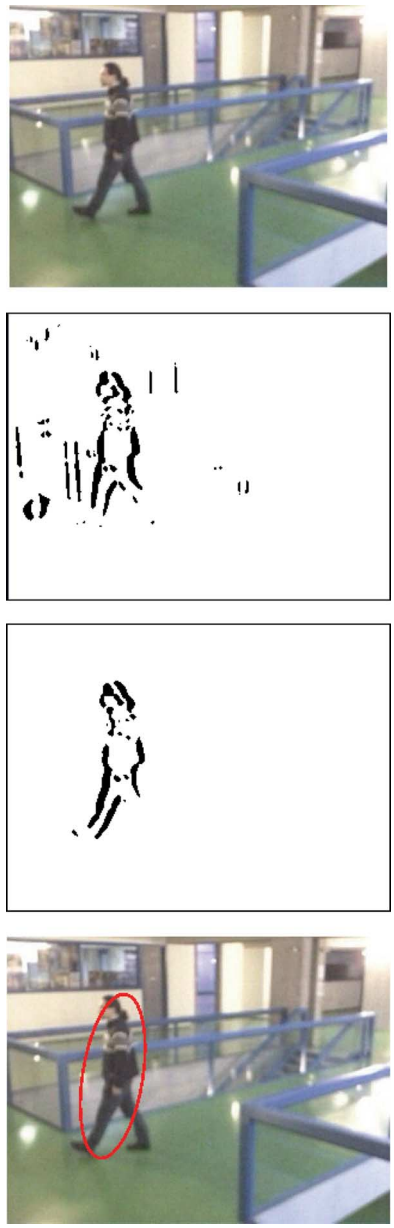


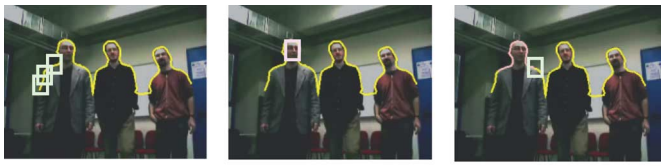
Fig. 3. Result of the alignments. Top row shows a test frame, and middle and bottom rows show the results obtained by the change detection algorithm executed on the alignments performed, respectively, by the least squares and the IRLS methods. The moving person inside the test frame has finally been highlighted with a bounding ellipse.

where x_i is the residual of the pixel i, and ρ' is the first derivative of the Lorentzian estimator function ρ defined as

$$\rho(x) = \log\left(1 + \frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right) \tag{6}$$

where σ is equal to a constant scaling factor. Fig. 3 shows the results of the proposed alignment methods for a test frame.

Once the alignment phase has been successfully executed, a change detection operation is applied on the two considered frames in order to detect the moving pixels. The resulting binary image is then processed by a seeded region growing technique [46] (the moving pixels are the seeds) in order to find clusters



(a) (b) (c)

Fig. 4. Example of iterative computation of the shape analysis on the spline contour related to the detected objects. (a) A sliding window is moved to detect a face on the basis of the shape analysis. (b) The sliding window is shown when a face is detected. (c) The next time instant the procedure continues skipping the contour classified as belonging to a face.

of moving pixels. The resulting cluster are then supposed to represent a single moving object. For each cluster, spatial and central moments are computed in order to find the center of mass and the bounding ellipse of the cluster, as shown in Fig. 3.

Sometimes the assumption that each cluster corresponds to a single object does not hold; for example, two objects that become very close each together can be erroneously enclosed in a single cluster. The system is robust to this kind of errors if the errors appear in a small number of consecutive frames, since in this case, the noisy data are filtered out by the Kalman filters described in Section IV. However, some sequences can show different objects moving together for a relatively long time, e.g., this is the case of a group of two or more people walking together. Since this is a common case for a system tuned to track people as the one proposed in this paper, an *ad hoc* algorithm is proposed in order to handle such situations.

The possible presence of multiple people inside the cluster can be hypothesized looking at the geometry of the bounding ellipse: If the ratio between the major and minor axis falls below a given threshold, the ellipse is assumed to be large enough to possibly contain more people. In such a case, the upper contours of the detected blobs are processed in order to compute a spline contour of the moving objects. A shape analysis method (described in Section V) is iteratively applied on a sliding window that moves along the spline contour. Once the shape analysis method returns a high probability of a face inside the current window, the position is recorded as corresponding to a face, and the process continues skipping the detected face contour belonging to the current window. In Fig. 4, an example of such a computation is shown.

A region segmentation algorithm [58] is then applied on the region of the frame where motion was detected, in order to subdivide the images of the moving people in regions of similar colors. These regions are then clustered together on the basis of the horizontal distance of their barycenter from the faces. This way, an approximate silhouette of each person is extracted, and the original cluster can be split in several moving objects, as shown in Fig. 5.

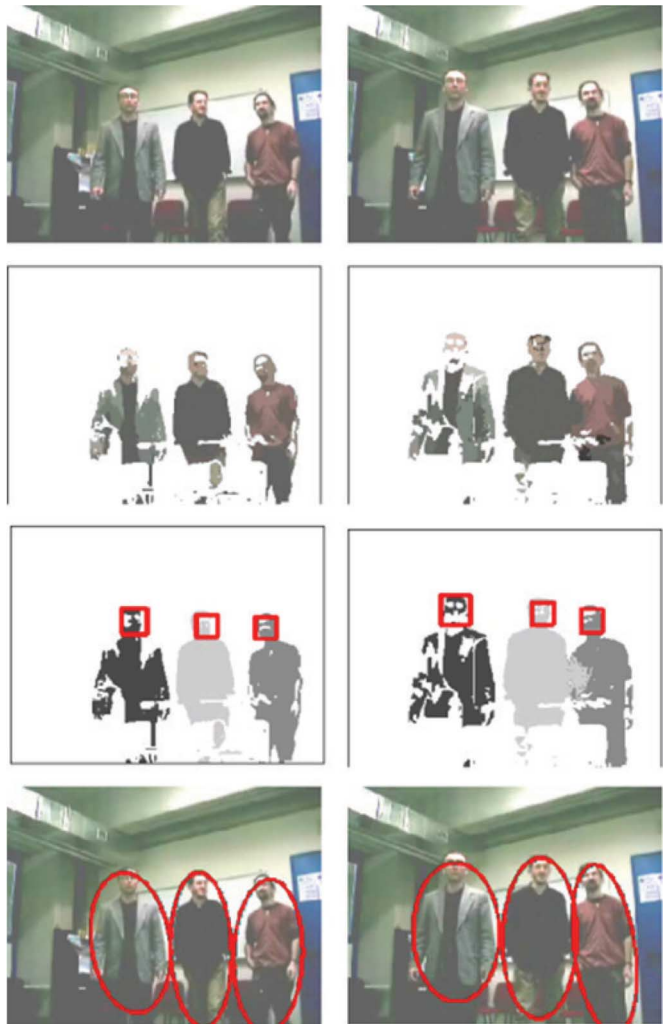


Fig. 5. People detection inside a group. First row: Original sequence; second row: the result of the segmentation algorithm; third row: face and body detection; last row: bounding ellipses for each person composing the group.

IV. Object Tracking

The main goal of the tracking module is to estimate at each time instant the position that the center of mass of the tracked object will assume at the next instant. In this way, the system can determine the motion required to maintain the target at the center of the image. The problem has been addressed by adopting a Kalman filter based on a rectilinear motion model. The state vector $X(k) = [x_c(k), y_c(k), \dot{x}_o(k), \dot{y}_o(k)]^{\mathrm{T}}$ represents the coordinates of the center of mass (x_c, y_c) and the speed of the object into the image plane $(\dot{x}_o, \dot{y}_o)$. The displacement of the center of mass among two time instants is defined as follows:

$$\begin{aligned} \Delta x_c &= \dot{x}_o \Delta t + \dot{x}_t \Delta t \\ \Delta y_c &= \dot{y}_o \Delta t + \dot{y}_t \Delta t \end{aligned} \tag{7}$$

where Δt is time interval between the measurements. The vector $[\dot{x}_t, \dot{y}_t]^{\mathrm{T}}$ represents the apparent speed of the object due to the camera motion, and it is computed from the affine transform parameters as follows:

$$\begin{aligned} \dot{x}_t &= a_{11} x_c + a_{12} y_c + a_{13} \\ \dot{y}_t &= a_{21} x_c + a_{22} y_c + a_{23}. \end{aligned} \tag{8}$$

Therefore, the process model of the Kalman filter has been defined by the following state equation:

$$\underbrace{\begin{pmatrix} x_c(k) \\ y_c(k) \\ \dot{x}_o(k) \\ \dot{y}_o(k) \end{pmatrix}}_{\mathbf{X}(k)} = \underbrace{\begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} x_c(k-1) \\ y_c(k-1) \\ \dot{x}_o(k-1) \\ \dot{y}_o(k-1) \end{pmatrix}}_{\mathbf{X}(k-1)} + \underbrace{\begin{pmatrix} \Delta t & 0 \\ 0 & \Delta t \\ 0 & 0 \\ 0 & 0 \end{pmatrix}}_{\mathbf{B}} \underbrace{\begin{pmatrix} \dot{x}_t(k) \\ \dot{y}_t(k) \end{pmatrix}}_{\mathbf{u}(k)} + \boldsymbol{\omega}_k \quad (9)$$

and by the measure equations

$$\begin{pmatrix} \hat{x}_c(k) \\ \hat{y}_c(k) \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}_{\mathbf{H}} \begin{pmatrix} x_c(k) \\ y_c(k) \end{pmatrix} + \boldsymbol{\upsilon}_k \quad (10)$$

where $(\hat{x}_c, \hat{y}_c)^{\mathrm{T}}$ is the position of the blob center. The quantities $\boldsymbol{\omega}_k$ and $\boldsymbol{\upsilon}_k$ represent, respectively, the process and measure errors and have been considered as Gaussian noises with zero mean and covariance matrices $\mathbf{Q}$ and $\mathbf{R}$.

The Kalman-filter process is defined trough its prediction and updating phases as follows:

PREDICTION

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{x}_{k-1} + \mathbf{B}\hat{u}_k$$

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}$$

UPDATING

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}^T \left(\mathbf{H}\mathbf{P}_k^-\mathbf{H}^T + \mathbf{R}\right)^{-1}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{H}\hat{x}_k^-\right)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^-.$$

Before applying the Kalman filter, some parameters must be initialized: 1) the status vector $\mathbf{X}(0)$; 2) the covariance matrices $\mathbf{R}$ and $\mathbf{Q}$ related to the noise affecting the dynamic and measure systems, respectively; and 3) the covariance matrix $\mathbf{P}(0)$. The covariance matrices are initialized by considering the expected motion. Therefore, since the matrix $\mathbf{P}(0)$ will be updated by the filtering process, greater importance has been given to the initialization of the matrices $\mathbf{R}$ and $\mathbf{Q}$. In particular, the following matrices have to take into account: the measurement error and the process error, respectively. To this end, the initialization is based on the results obtained by Arsenio and Victor in [47] and by Kohler in [48], with the following setup of the covariance matrices:

$$Q = \frac{a_c^2\Delta t}{6}\begin{pmatrix} 2I\Delta t^2 & 3I\Delta t \\ 3I\Delta t & 6I \end{pmatrix} \quad \text{and} \quad R = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix} \quad (11)$$

where $a_c = 12\ \text{pixel/frame}^2$ is the spectral amplitude of the white noise that defines the object acceleration, and σ_x^2 and σ_y^2 are the error variances that could be experimentally estimated simply by considering that the ground truth position of the center of mass generally falls inside the computed bounding ellipse. Let us suppose that the ellipse axes are parallel to the image axis; then, by assuming a and b as the length of the two axes, we obtain

$$\sigma_x^2 = \frac{a^2}{4} \quad \text{and} \quad \sigma_y^2 = \frac{b^2}{4}. \quad (12)$$

Furthermore, by considering the rotation of the bounding ellipse of an angle θ, we give a new definition of the covariance matrix $\mathbf{R}'$ as follows:

$$\mathbf{R}' = \begin{pmatrix} \frac{a^2}{4}\cos^2\theta + \frac{b^2}{4}\sin^2\theta & \left(\frac{a^2}{4} - \frac{b^2}{4}\right)\cos\theta\sin\theta \\ \left(\frac{a^2}{4} - \frac{b^2}{4}\right)\cos\theta\sin\theta & \frac{b^2}{4}\cos^2\theta + \frac{a^2}{4}\sin^2\theta \end{pmatrix} = \mathbf{M}\mathbf{R}\mathbf{M}^{\mathrm{T}} \quad (13)$$

where $\mathbf{M}$ is the rotation matrix. The covariance matrix $\mathbf{R}'$ is updated at each time instant by considering in its computation the parameters (a, b, θ) related to the bounding ellipse computed at the previous step.

The proposed filter is well suited to track the position of a single object and should be applied to every object detected with the procedure described in Section III; this means that every object has its own vector state and Kalman matrices.

V. FACE DETECTION AND RECOGNITION

To address the problem of face detection and recognition, the proposed method (see Fig. 6) processes the bounding ellipse with a shape analysis method. The objective of this computation is to identify the upper silhouette of the considered blob in order to limit further computations to a restricted region of interest (RoI). Once the area of interest has been determined, three different face-detection modules are applied, each one based, respectively, of the following techniques: 1) principal component analysis (PCA); 2) neural networks; and 3) skin regions. Therefore, a fusing process gives an estimation of the face position by exploiting the results of the single face detectors and by considering a temporal tracking information carried out by a Kalman filter process.

The PCA module extracts all the possible patterns from the area of interest and projects each of them to a face space that has been built considering the AR face [49] and by adopting an eigenfaces technique. Precisely, since objects can appear in the image at different scales (due to zoom level or to the distance from the camera), we have introduced a heuristic based on planes of depth corresponding to three different scales on which the PCA method has been trained. In particular, we trained the PCA by scaling the images from the AR database [49] to obtain the following template dimensions: 20×24, 30×36, and 40×49 (see Fig. 7). Once the object has been detected and its bounding ellipse computed, the pattern dimension is selected on the basis of the ellipse area A using the following rule:

$$\text{Depth} = \begin{cases} 0, & \text{if } A < \text{th}_1 \\ 1, & \text{if } \text{th}_1 \le A < \text{th}_2 \\ 2, & \text{if } \text{th}_2 \le A \end{cases} \quad (14)$$

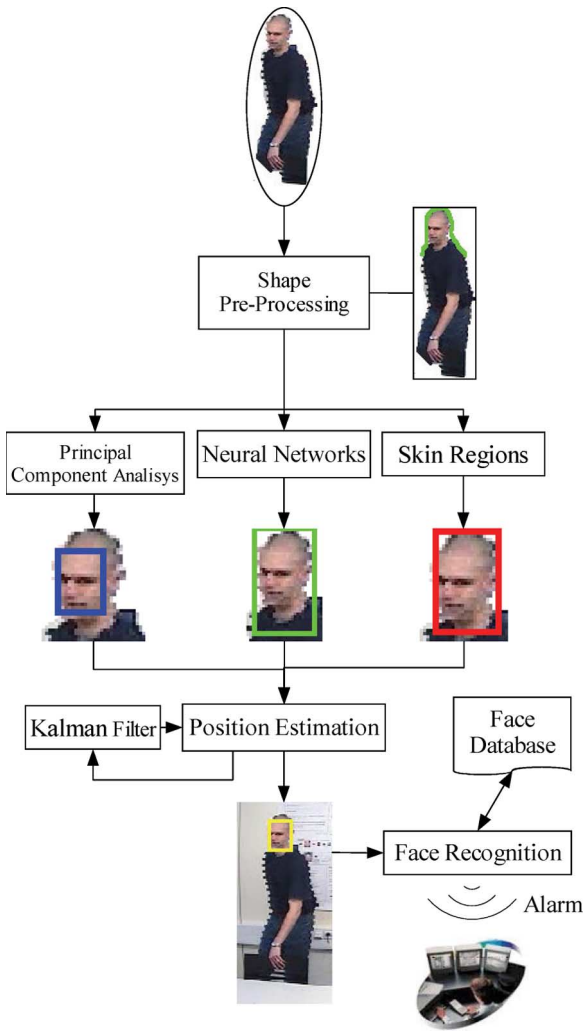


Fig. 6. General architecture of the face-detection and recognition module. Displayed results have been computed on a PETS2002 [62] sequence.

Fig. 7. Example of the three depth at which three different templates for the PCA method have been associated. (a) When the object is far from the camera, the search proceeds on pattern of dimension 20 × 24. (b) When the object is at a medium distance from the camera, the adopted pattern has size of 30 × 36. (c) When it is close to the camera, we adopted a search pattern of 40 × 49.

where depth 0, 1, 2 correspond, respectively, to the three template dimensions, and th_1 and th_2 are two experimentally determined thresholds (i.e., we adopted $\mathrm{th}_1 \in [8500, 9000]$ and $\mathrm{th}_2 \in [9500, 10\,000]$ pixels).

Let y be the projection in the PCA space of the unclassified pattern x, $\overline{x}$ the main face computed on the database patterns, and M the number of meaningful eigenvectors considered. The distances ε^2 and d^2 of y from the face eigenspace

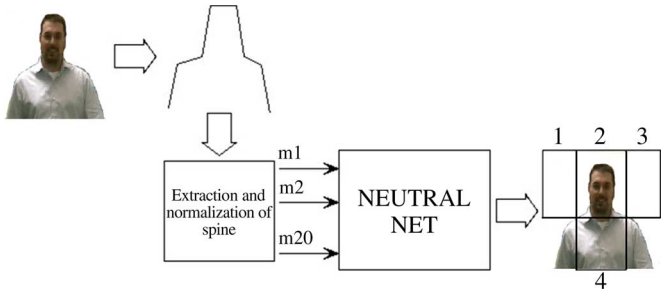


Fig. 8. Outline analysis with neural networks.

(DFFS—Distance From Face Space) and inside the eigenspace (DIFS—Distance In Face Space) are, respectively, defined as

$$\varepsilon^2 = \|x - \overline{x}\|^2 - \sum_{i=1}^{M} y_i^2$$
$$d^2 = \sum_{i=1}^{M} \frac{y_i^2}{\lambda_i^2}. \tag{15}$$

Finally, an error measure representing the closeness of the unknown pattern x to a face pattern has been defined as

$$\varepsilon_{\mathrm{dist}} = d^2 + c\varepsilon^2 \tag{16}$$

where $c = (1/k\lambda_M)$ represents a scaling factor, $k \in [1, \ldots, 5]$ is an experimentally defined constant related to the grade of diversity among the training patterns and the unknown pattern (i.e., different types of environment where the algorithm is applied), and λ_M is the minimum eigenvalue. The error $\varepsilon_{\mathrm{dist}}$ is finally thresholded to detect the center of the face box. If its value is lower than the threshold, no face box is returned; otherwise, a box of the selected dimensions centered in the current position is returned.

The neural network method applies a multilayer perceptron (MLP) neural network trained by a back-propagation algorithm with the n coefficients of a B-spline which approximates the upper blob shape. A number of $n = 20$ coefficients has been determined experimentally. The neural network has four possible output classes. Each class represents a possible position of the head in the upper part of the blob (see Fig. 8). The neural network architecture is defined by a full-connection MLP structure composed of 20 input units representing the spline coefficients, 40 hidden units, and four output units, each one related to the probability of the presence of a face in a specific zone.

The last face detector employed by the system adopts a filter based on the Cb–Cr color space [50]. The skin regions are detected by applying the filter on the RoI and by further polishing the results by means of morphological operators. In order to address the problem relative to multiple skin regions (head, arms, legs, etc.), a selection based on the position and dimension of the skin region boxes is performed. In particular, if the box dimensions are too small with respect to the size of the blob of the person, or the region is found in a position too low with respect to the normal position of the head, then the box is discarded.

The results obtained by each localization method are than processed by a position estimation module. At this step, a unique location of the head is computed by applying an information fusion technique [51]. In addition, since the localization process of the face may be corrupted by occlusions or errors derived by the change detection technique, a face tracker based on a Kalman filter has been developed [47]. Once the pattern of the face has been determined, the principal components are extracted, and the recognition technique proposed in [52] is applied.

VI. Experimental Results

The proposed method has been tested on sequences acquired in an indoor environment. Several experiments have been executed following a strategy that involves an increasing complexity for the tests. The sequences used for the tests have been acquired by a Cohu 3812 charge-coupled-device camera mounted on a Robosoft Pioneer 2-DXe mobile indoor platform and are characterized by images of 320 $\times$ 240 pixels. The tests have been performed on a laptop equipped with an Athlon 2000 + processor and 256 MB of RAM.

A. Motion Detection in Empty Scenes

The first group of tests was performed in order to justify the choice of an affine model of motion and involved the processing of sequences with no moving objects. The presence of false positive results (detection of moving objects) in these tests would be a clear evidence that the affine model is not suitable for a mobile system. In order to numerically evaluate the performance of the affine registration algorithm, a measure related to the logarithmically weighted mean of the intensity values in the difference images was adopted:

$$E = \frac{\sum_i g_i \log p_i}{\sum_i \log p_i}$$

where g_i are the gray values present in the image, and p_i is the number of pixels with intensity value g_i. Since the tests do not involve moving objects, the difference image should ideally be black, and the error measure should be equal to 0.

The first problem with the affine model is that it cannot model the convergence and chirping effects introduced by the rotations of the camera [61], and these effects are more and more evident as the field of view increases. A more suitable model would be the pseudoperspective one

$$x' = a_1 x + a_2 y + a_3 + a_7 x^2 + a_8 xy$$
$$y' = a_4 x + a_5 y + a_6 + a_7 xy + a_8 y^2.$$

Both the affine and the pseudoperspective models have been applied to sequences where the camera rotates with a big field of view. Fig. 9 shows one of these sequences and Fig. 10 shows the error E in both cases. It can be seen that the pseudoperspective model gives only a small increment in system performance, while it has some serious drawbacks: It is computationally more expensive since it has eight parameters to be estimated instead of the six of the affine model and sometimes causes the nonconvergence of the registration algorithm, probably due to the amplification of error noise in the quadratic terms.

Fig. 9. Frames 0–130–260–400 of the rotational sequence used for the evaluation of the affine model.

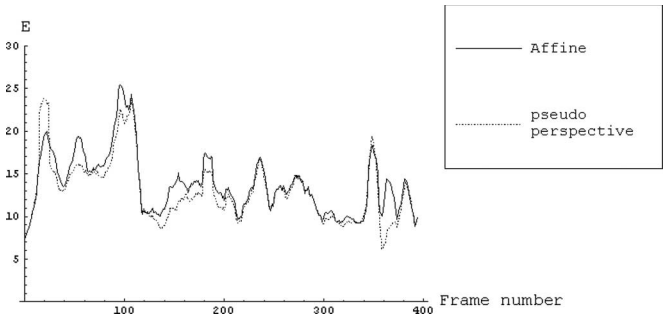


Fig. 10. Alignment error for affine and pseudoperspective models.

Fig. 11. Frames 0–50–100–140 of the translational sequence for the parallax error analysis.

Another important problem with the affine (and pseudoperspective too) model is that it cannot handle the presence of parallax movement, which arises when the camera translates and the scene is not planar. While this is true, it is also true that the parallax movement is negligible when the distance of the objects from the camera is much longer than the translation of the camera between the acquisition of two consecutive frames; this means that the alignment error is proportional to the camera speed.

A test was made on a translational sequence (shown in Fig. 11) involving a typical hallway scene. The sequence was obtained moving the camera at the speed of 0.20 m/s and was then processed at full frame rate, taking one frame every two and taking one frame every four, thus simulating a speed of 0.40 and 0.80 m/s without the need of using a different sequence.

The alignment error E is shown in Fig. 12. It can be seen that the error increases with the increment of speed, but the system never recognizes false positives, thus still giving good results. We estimate that the system can perform well in such an environment with a camera speed up to 1 m/s. For faster speeds, the parallax error becomes nonnegligible, thus imposing an upper limit to the speed of the mobile system. If a faster object has to be followed, it is possible to partially overcome this limit by using the camera zoom, which is an affine transform and does not introduce parallax movement.

In all the tests presented above, the number of iterations needed for the convergence of registration algorithm was always very small—ranging from one to four iterations per frame.

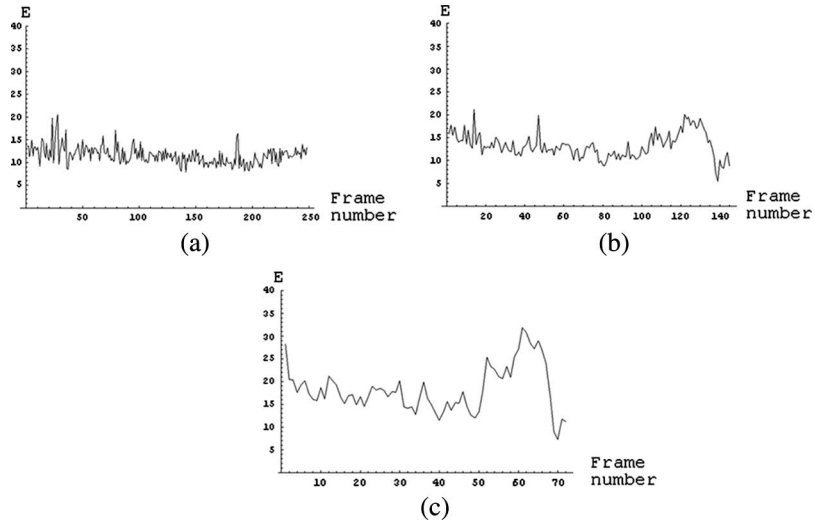


Fig. 12. Alignment error in the translational sequences [(a) slow, (b) medium, and (c) fast].

Fig. 13. Object detection on frames belonging to a test sequence.

B. Object Detection

The second phase in testing the motion detection module has been executed on sequences in which at least one object moves inside the monitored scene. In this context, multiple types of camera motion have been taken into account considering two types of motion for the moving objects: a tangent motion in which the trajectory of the object is orthogonal to the optical axis and a radial motion in which the object moves toward or away from the camera.

This phase has revealed a good behavior of the system since the object detection has been successful in 97.4% of the frames. It is worth noting that the remaining 2.6% of the cases in which the detection failed are related to spots and not to continuous frames.

For these tests, the mean value of the error E is equal to 12.58, which is close to the performances obtained in context of no moving objects. We should focus the attention on the method used to compute this value. Since, in this context, the computation of the error also in the area of the object would alter the results, the area described by the bounding ellipse has been removed from the computation of the parameter E. Regarding the number of iterations needed to converge, the value of 1.92 is again very close to results obtained with empty scenes. An example of object detection for a test sequence is shown in Fig. 13.

C. Object Tracking

To test the performance of the object tracking module, three people have been asked to process the test sequences in order to define the ground truth position of the moving person barycenter. Then, the ground truth position for each frame has been computed as a mean of the position marked by each person.

The result obtained in context of tangent motion points out a good level for the performance of our method. For these kinds of sequences, the mean error in the estimation of the barycenter is equal to 20.85 pixels. This error is principally due to the vertical component, while if we consider only the estimation error performed on the horizontal axis, the performance increases by reducing the value to 12.18 pixels. This is a good behavior, considering the motion involved in this type of sequence.

Regarding the radial motion, the global error has been equal to 16.43 pixels, which is lower than the error performed on the previous sequences. Also, in this context, by considering only the horizontal component, the error decreases to 9.66 pixels.

Globally, the object tracking module has supplied good results, allowing us to maintain the object near the center of the image for the majority of the frames belonging to the test sequences. An example of the results obtained by the object tracking module can be seen in Fig. 14.

D. Face Detection and Recognition

The proposed method has been tested on the images and, in particular, on the bounding ellipse returned by the motion detection module. Several people movements have been taken into account during the acquisition process. The predicted error (PE) metric selected to assess the algorithm efficiency is the Euclidean distance between the center of mass calculated by the system and the ground truth one.

In these tests, three people have been requested to mark the position of the barycenter of the face by pointing to it with the mouse.

The mean PE value is equal to 2.63. This is a good result if we consider the problems related to indoor environments. In particular, the system must deal with reflection due to the

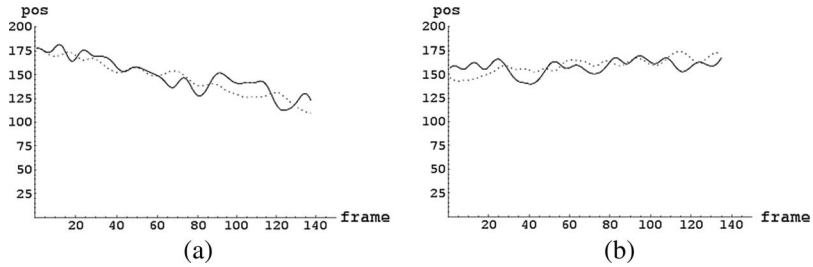


Fig. 14. Example of object tracking. The chart plots the (a) vertical position and the (b) horizontal position of the barycenter of the rightmost person in Fig. 13, as marked by the testers (dotted line) and estimated by the tracking module (continuous line).

glass that surrounds the passageways. These problems reflect on the goodness of the face-detection methods employed. In addition, when occlusions appear on the face patterns, the proposed method cannot be applied, and the system keeps tracking the position of the face by means of Kalman estimation. This information can be exploited by the control subsystem to plan the robot motion for reducing the occlusions.

The small PE value has allowed the execution of the identification of good patterns that often were brought to the center of the face. Although the identification has been applied on good patterns, the obtained results have demonstrated that this field of computer vision has not yet reached an optimal solution. Only in 73% of the cases on over 10^3 frames was the identification correct. This quantity increases if we consider the percentage of people correctly identified through the sequences. The percentage of success in this case has been equal to 80.9%.

VII. Conclusions

In this paper, the use of an autonomous vehicle, called ASV, for the surveillance and monitoring of indoor environments has been exploited. The ASV has been specifically designed to help a remote operator in monitoring wide indoor areas. For such purposes, it is able to move around a specific indoor environment (e.g., a building) and to track moving people. The selection of the target object to be tracked can be decided by the remote operator or autonomously by the ASV itself in the case that a suspicious behavior has been detected (e.g., a person entering a forbidden area, etc.). Additional surveillance procedures like face detection and recognition of interesting target persons can be performed by the ASV.

Several experiments on indoor sequences have demonstrated that the proposed ASV performs a robust detection of the motion inside the monitored scene. Then, to achieve a good identification of the mobile objects and to track them with enough accuracy, the ASV maintains the objects inside the field of view. Finally, the system shows a good face-detection rate for further person identification.

Although the results appear promising, the constraints imposed still limit the exploitation of such a vehicle as a fully autonomous surveillance system. In particular, the face detector applied in cluttered environments does not guarantee optimal performances, especially due to occlusions. Thus, the use of a multiple camera network could be useful to solve the occlusions and to improve the understanding of the monitored scene. Moreover, the exploitation of an affine model as a registration technique limits the maximum speed of the vehicle. Indeed, when the speed remarkably increases, the parallax error cannot be addressed by the proposed method. This problem can be solved using additional information on the scene depth that could be acquired by range sensors mounted on board. Hence, future works concerning the study of cooperative algorithms and the use of 2-D laser range sensors will certainly give the robustness required to efficiently adopt the proposed ASV for the surveillance of indoor environments.

References

[1] T. Soni and B. Sridhar, "Modeling issues in vision based aircraft navigation during landing," in *Proc. IEEE Workshop Appl. Comput. Vision*, Sarasota, FL, Dec. 5–7, 1994, pp. 89–96.

[2] I. Cohen and G. Medioni, "Detecting and tracking moving objects in video from and airborne observer," in *Proc. DARPA Image Understanding Workshop*, Monterey, CA, Nov. 20–23, 1998, pp. 217–222.

[3] R. Garcia, X. Cufi, and M. Carreras, "Estimating the motion of an underwater robot from a monocular image sequence," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, Maui, HI, Oct. 29–Nov. 3, 2001, vol. 3, pp. 1682–1687.

[4] J. Rosenblatt, S. Willams, and H. Durrant-Whyte, "Behavior-based control for autonomous underwater exploration," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Francisco, CA, Apr. 24–28, 2000, pp. 920–925.

[5] A. Bennett and J. J. Leonard, "A behavior-based approach to adaptive feature mapping with autonomous underwater vehicles," *IEEE J. Ocean. Eng.*, vol. 25, no. 2, pp. 213–226, Apr. 2000.

[6] M. Hebert, C. Thorpe, and A. Stentz, *Intelligent Unmanned Ground Vehicles: Autonomous Navigation Research at Carnegie Mellon*. Norwell, MA: Kluwer, 1997.

[7] B. Southall, T. Hague, J. A. Marchant, and B. F. Buxton, "Vision-aided outdoor navigation of an autonomous horticultural vehicle," in *Proc. 1st ICVS*, Gran Canaria, Spain, Jan. 13–15, 1999, pp. 37–50.

[8] T. Gandhi and M. M. Trivedi, "Motion analysis for event detection and tracking with a mobile omni-directional camera," *Multimedia Syst.*, vol. 10, no. 2, pp. 131–143, 2004.

[9] C. Thorpe, M. H. Herbert, T. Kanade, and S. A. Shafer, "Vision and navigation for the Carnegie Mellon Navlab," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 3, pp. 362–372, May 1998.

[10] D. A. Pomerlau, "Reliability estimation for neural network based autonomous driving," *Robot. Auton. Syst.*, vol. 12, no. 3/4, pp. 113–119, Apr. 1994.

[11] E. D. Dickmanns and B. Mysliwets, "Recursive 3D road and relative egostate recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 199–213, Feb. 1992.

[12] E. D. Dickmanns, "Computer vision and highway automation," *Veh. Syst. Dyn.*, vol. 31, no. 5, pp. 325–343, Jun. 1999.

[13] D. A. Pomerlau and T. Jockem, "Rapidly adapting machine vision for automated vehicle steering," *IEEE Intell. Syst.*, vol. 11, no. 2, pp. 19–27, Apr. 1996.

[14] 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. Inf. Syst.*, vol. E83-D, no. 7, pp. 1583–1591, Jul. 2000.

[15] G. Giralt, R. Sobek, and R. Chatila, "A multi-level planning and navigation system for a mobile robot: A first approach to Hilare," in *Proc. Int. Joint Conf. Artif. Intell.*, 1979, vol. 1, pp. 335–337.

[16] H. P. Moravec, "The Standford Cart and the CMU rover," *Proc. IEEE*, vol. 71, no. 7, pp. 872–884, Jul. 1983.
[17] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 2, pp. 237–267, Feb. 2002.
[18] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," *Proc. IEEE*, vol. 89, no. 10, pp. 1456–1477, Oct. 2001.
[19] C. Regazzoni, V. Ramesh, and G. Foresti, "Special issue on video communications, processing, and understanding for third generation surveillance systems," *Proc. IEEE*, vol. 89, no. 10, pp. 1355–1539, Oct. 2001.
[20] L. Davis, R. Chellapa, Y. Yaccob, and Q. Zheng, "Visual surveillance and monitoring of human and vehicular activity," in *Proc. DARPA Image Understanding Workshop*, New Orleans, LA, May 13–15, 1997, pp. 19–27.
[21] R. Howarthand and H. Buxton, "Visual surveillance monitoring and watching," in *Proc. Eur. Conf. Comput. Vis.*, Cambridge, U.K., Apr. 13–14, 1996, pp. 321–334.
[22] T. Kanade, R. Collins, A. Lipton, P. Burt, and L. Wixson, "Advances in cooperative multisensor video surveillance," in *Proc. DARPA Image Understanding Workshop*, Monterey, CA, Nov. 20–23, 1998, pp. 3–24.
[23] G. L. Foresti, "Object recognition and tracking for remote video surveillance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 7, pp. 1045–1062, Oct. 1999.
[24] G. L. Foresti, P. Mahonen, and C. Regazzoni, *Multimedia Video-Based Surveillance Systems: From User Requirements to Research Solutions*. Norwell, MA: Kluwer, Sep. 2000.
[25] D. Koller, K. Daniilidis, and H. H. Nagel, "Model-based object tracking in monocular sequences of road traffic scenes," *Int. J. Comput. Vis.*, vol. 10, no. 3, pp. 257–281, Jun. 1993.
[26] Z. Zhu, G. Xu, B. Yang, D. Shi, and X. Lin, "VISATRAM: A real-time vision system for automatic traffic monitoring," *Image Vis. Comput.*, vol. 18, no. 10, pp. 781–794, Jul. 2000.
[27] S. Dockstader and M. Tekalp, "Multiple camera tracking of interacting and occluded human motion," *Proc. IEEE*, vol. 89, no. 10, pp. 1441–1455, Oct. 2001.
[28] S. Park and J. Aggarwal, "A hierarchical Bayesian network for event recognition of human actions and interactions," *Multimedia Syst.*, vol. 10, no. 2, pp. 164–179, Aug. 2004.
[29] J. Gluckman and S. Nayar, "Ego-motion and omnidirectional camera," in *Proc. IEEE Int. Conf. Comput. Vis.*, Bombay, India, Jan. 3–5, 1998, pp. 999–1005.
[30] S. Nayar and T. Boult, "Omnidirectional vision systems," in *Proc. DARPA Image Understanding Workshop*, New Orleans, LA, May 13–15, 1997, pp. 235–242.
[31] A. Davison, I. D. Reid, and D. Murray, "The active camera as a projective pointing device," in *Proc. 6th Brit. Mach. Vis. Conf.*, Birmingham, U.K., Sep. 11–14, 1999, pp. 11–14.
[32] D. Murray and A. Basu, "Motion tracking with an active camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 5, pp. 449–454, May 1994.
[33] R. Okada, Y. Shirai, and J. Miura, "Object tracking based on optical flow and depth," in *Proc. IEEE/SICE/RSI Int. Conf. Multisensor Fusion and Integr.*, Washington, DC, Dec. 8–11, 1996, pp. 565–571.
[34] G. L. Foresti and C. Micheloni, "A robust feature tracker for active surveillance of outdoor scenes," *Electron. Lett. Comput. Vis. Image Anal.*, vol. 1, no. 1, pp. 21–34, 2003.
[35] C. Tomasi and T. Kanade, "Detection and tracking of point features," Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-91-132, Apr. 1991.
[36] L. Snidaro and G. L. Foresti, "Real-time thresholding with Euler numbers," *Pattern Recognit. Lett.*, vol. 24, no. 9/10, pp. 1533–1544, Jun. 2003.
[37] T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto, "Making good features track better," in *Proc. IEEE Conf. Comput. Vis. and Pattern Recog.*, Santa Barbara, CA, Jun. 23–25, 1998, pp. 178–183.
[38] B. K. P. Horn and E. J. Weldon, "Direct methods for recovering motion," *Int. J. Comput. Vis.*, vol. 2, no. 1, pp. 51–76, Jun. 1988.
[39] G. P. Stein and A. Shashua, "Model-based brightness constraints: On direct estimation of structure and motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 9, pp. 992–1015, Sep. 2000.
[40] H. P. Moravec and A. Elfers, "High resolution maps from wide angle sonar," in *Proc. IEEE Int. Conf. Robot. Autom.*, St. Louis, MO, Mar. 25–28, 1985, pp. 116–121.
[41] J. Horn and G. Schmidt, "Continuous localization for long-range indoor navigation of mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Nagoya, Japan, May 21–23, 1995, vol. 1, pp. 387–394.
[42] I. J. Cox, "Modeling a dynamic environment using a Bayesian multiple hypothesis approach," *Artif. Intell.*, vol. 66, no. 2, pp. 311–344, Apr. 1994.
[43] T. Duckett, S. Marsland, and J. Shapiro, "Learning globally consistent maps by relaxation," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Francisco, CA, Apr. 24–28, 2000, vol. 4, pp. 3841–3846.
[44] B. K. P. Horn and B. G. Schunk, "Determining optical flow," *Artif. Intell.*, vol. 17, no. 1–3, pp. 185–203, Aug. 1981.
[45] M. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications to early vision," *Int. J. Comput. Vis.*, vol. 19, no. 1, pp. 57–92, 1996.
[46] J. Fan, D. Yau, A. Elmagarmid, and W. Aref, "Automatic image segmentation by integrating color-edge extraction and seeded region growing," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1454–1466, Oct. 2001.
[47] A. Arsenio and J. Victor, "Active monocular tracking with temporal integration of visual cues," in *Proc. 9th Portoguese Conf. Pattern Recog.*, Coimbra, Portugal, Mar. 20–21, 1997.
[48] M. Kohler, *3D Image Analysis and Synthesis*. New York: Springer-Verlag, 1996, ch. Vision Based Remote Control in Intelligent Home Enviroments, pp. 147–154.
[49] A. M. Martinez and R. Benavente, "The AR face database," Comput. Vis. Center (CVC), Barcelona, Spain, CVC Tech. Rep. 24, Jun. 1998.
[50] D. Chai and A. Bouzerdoum, "A Bayesian approach to skin color classification in YcbCr color space," in *Proc. IEEE Region Ten Conf.*, Kuala Lumpur, Malaysia, Sep. 2000, pp. 421–424.
[51] G. L. Foresti, C. Micheloni, and L. Snidaro, "A robust face detection system for real environments," in *Proc. Int. Conf. Image Process.*, Barcelona, Spain, Sep. 14–17, 2003, pp. 897–900.
[52] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3, no. 1, pp. 71–86, Jan. 1991.
[53] M. Björkman and J. O. Eklundh, "Real-time epipolar geometry estimation and disparity," in *Proc. Int. Conf. Comput. Vis.*, Corfù, Greece, Sep. 20–25, 1999, pp. 234–241.
[54] ——, "Visual cues for a fixating active agent," in *Proc. Int. Workshop Robot Vis.*, Auckland, New Zealand, 2001, pp. 1–8.
[55] ——, "Real-time epipolar geometry estimation of binocular stereo heads," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 425–432, Mar. 2002.
[56] M. J. Black and D. J. Fleet, "Probabilistic detection and tracking of motion boundaries," *Int. J. Comput. Vis.*, vol. 38, no. 3, pp. 231–245, Jul./Aug. 2000.
[57] M. Irani and P. Anandan, "A unified approach to moving object detection in 2D and 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 6, pp. 577–589, Jun. 1998.
[58] P. J. Burt, T. H. Hong, and A. Rosenfeld, "Segmentation and estimation of image region properties through cooperative hierarchical computation," *IEEE Trans. Syst., Man Cybern.*, vol. 11, no. 12, pp. 802–809, Dec. 1981.
[59] P. Anandan, P. J. Burt, K. Dana, M. Hansen, and G. Van der Wal, "Real-time scene stabilization and mosaic construction," in *Proc. DARPA Image Understanding Workshop*, Monterey, CA, Nov. 13–16, 1994, pp. 457–465.
[60] M. Irani and P. Anandan, "About direct methods," in *Proc. Int. Workshop Vis. Algorithms: Theory and Practice ICCV*, Corfù, Greece, Sep. 21–22, 1999, pp. 267–277.
[61] S. Mann and R. W. Picard, "Video orbits of the projective group: A simple approach to featureless estimation of parameters," *IEEE Trans. Image Process.*, vol. 6, no. 9, pp. 1281–1295, Sep. 1997.
[62] VS-PETS, *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, in Conjunction With European Conference on Computer Vision*, May 27–Jun. 2, 2002, Copenhagen, Denmark.

Christian Micheloni (S'02–M'07) received the Laurea degree (*cum laude*) and the Ph.D. degree in computer science, both from University of Udine, Udine, Italy, in 2002 and 2006, respectively.

Since 2000, he has taken part in European research, being under contract for several European projects. He has coauthored more than 30 scientific works published in international journals and refereed international conferences. He serves as a reviewer for several international journals and conferences. His main interests involve active vision for the understanding of dynamic scenes from images acquired by a moving camera and neural networks for the classification and recognition of the objects moving within the scene. He is also interested in pattern recognition techniques for both the automatic tuning of the camera parameters for improved image acquisition and for the detection of faces. All these techniques have mainly been developed and applied for video surveillance purposes.

Gian Luca Foresti (S'93–M'95–SM'01) was born in Savona, Italy, in 1965. He received the Laurea degree (*cum laude*) in electronic engineering and the Ph.D. degree in computer science, both from University of Genoa, Genoa, Italy, in 1990 and in 1994, respectively.

Since 2000, he has been a Professor of computer science with the Department of Mathematics and Computer Science, University of Udine, Udine, Italy, where he is also Director of the Artificial Vision and Real-Time Systems Laboratory. His main interests involve multisensor data and information fusion, computer vision and image processing, and artificial neural networks and pattern recognition. Some of techniques he has proposed have found applications in the following fields: video-based surveillance systems, autonomous vehicle driving, road traffic control, human behavior understanding, and visual inspection. He has authored or coauthored more than 200 papers published in international journals and refereed international conferences. He has contributed to several books in his area of interest, and he is coauthor of the books *Multisensor Surveillance Systems: The Fusion Perspective* (Kluwer, 2003) and *Ambient Intelligence: A Novel Paradigm* (Springer, 2005).

Dr. Foresti was General Chairman and member of Technical Committees at several conferences where he has been coorganizer of several special sessions on data fusion, image processing, and pattern recognition. He was Finance Chair of the 2005 IEEE Conference on Image Processing, Italy. He was Guest Editor of a Special Issue of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS on *Ambient Intelligence* and of a Special Issue of the PROCEEDINGS OF IEEE on *Video Communications, Processing, and Understanding for Third Generation Surveillance Systems*. He serves as a reviewer for several international journals and for the European union in different research programs. He is member of International Association of Pattern Recognition (IAPR).

Claudio Piciarelli received the M.Sc. degree in computer science from the University of Udine, Udine, Italy, in 2003, where he has been working toward the Ph.D. degree since 2005.

Since 1999, he has been working on several national and European projects in collaboration with the Artificial Vision and Real-time Laboratory, University of Udine. He has coauthored more than ten works published on international journals and conferences and has been a reviewer for several international conferences as well as for international journals. His main research interests include computer vision, artificial intelligence, pattern recognition, and machine learning. He is currently mainly involved in the use of active vision in surveillance systems and in behavior analysis for automatic detection of anomalous events.

Luigi Cinque (M'89–SM'96) received the degree in physics from University of Napoli, Napoli, Italy, in 1983.

From 1984 to 1990, he was with the Artificial Intelligence Laboratory (Alenia S.p.A), working on the development of expert systems and knowledge-based vision systems. In 1990, he joined the Department of Computer Science, University "La Sapienza," Rome, Italy, as Assistant Professor. Currently, he is Full Professor of computer science. His current interests include computer vision, parallel image analysis and architectures, shape and object recognition, image sequences analysis, and image databases. He has held visiting positions with the University of Maryland, College Park, and the University of Washington, Seattle. He is the author of over 150 scientific publications in international journals and conference proceedings and has been Guest Editor of several special issues about imaging technology.

Prof. Cinque is currently Associate Editor of *Pattern Recognition and Pattern Recognition Letters*. He is a member of IAPR and the Association for Computing Machinery. He has been on the program committees of many international conferences in the field of imaging technology and multimedia computing.