Probabilistic Posture Classification for Human-Behavior Analysis

Rita Cucchiara, *Member, IEEE*, Costantino Grana, Andrea Prati, *Associate Member, IEEE*, and Roberto Vezzani

***Abstract*—Computer vision and ubiquitous multimedia access nowadays make feasible the development of a mostly automated system for human-behavior analysis. In this context, our proposal is to analyze human behaviors by classifying the posture of the monitored person and, consequently, detecting corresponding events and alarm situations, like a fall. To this aim, our approach can be divided in two phases: for each frame, the projection histograms (*Haritaoglu et al.*, 1998) of each person are computed and compared with the probabilistic projection maps stored for each posture during the training phase; then, the obtained posture is further validated exploiting the information extracted by a tracking module in order to take into account the reliability of the classification of the first phase. Moreover, the tracking algorithm is used to handle occlusions, making the system particularly robust even in indoors environments. Extensive experimental results demonstrate a promising average accuracy of more than 95% in correctly classifying human postures, even in the case of challenging conditions.**

***Index Terms*—People tracking, posture classification, probabilistic projection maps (PPMs).**

I. Introduction

EMERGING technologies can offer a very interesting contribution in improving the quality of the life of people staying at home or working indoors. Most of these techniques and the related systems are converging in the new discipline of ambient intelligence that includes ubiquitous computer systems, intelligent sensor fusion, remote control, telehealth care, video surveillance, and many other pervasive infrastructure components.

One important goal of these systems is *human-behavior analysis*, especially for safety purposes: noninvasive techniques, such as those based on processing videos acquired with distributed cameras, enable us to learn about the presence and the behavior of people in a given environment.

In our target application of smart *Domotics*,[1] we aim to monitor the behavior of people in the home (especially for elders with low autonomy) and define potential alarm situations.

Indeed, recent research in computer vision on people surveillance jointly with research in efficient remote multimedia access makes feasible a complex framework where people in the home can be monitored in their daily activities in a fully automatic way, in total agreement with privacy policies. A well-formalized set of alarm situations can be defined and used as the trigger of some actions, such as communication to remote users, control centers or private people. Finally, only in such a situation, remote users can also connect with low-cost devices, such as GPRS phones and PDAs.

In this context of video surveillance, most of the emphasis is devoted to techniques capable of execution in real time on standard computing platforms and with low-cost off-the-shelf cameras. Additionally, in indoor surveillance of people's behavior, the techniques must cope with problems of robustness and reliability. For instance, in videos acquired with a fixed camera, the visual appearance of a person is often cluttered and overlapped with home furniture, other people, and so on.

In this paper, we propose a set of computer vision and motion-analysis techniques to extract objects and events from the scene, and to classify and recognize them in accordance with a previously defined ontology. For instance, we have defined the event "fallen person" that is recognized whenever an object classified as person changes its posture to "laying" and remains in that posture for a given period.

In particular, people detection is achieved by using the system proposed in [2], that provides, frame-by-frame, the list of moving visual objects (VOs) that can be classified as potential people. Moreover, a probabilistic tracking module (inspired by the work of Senior *et al.* [3]) allows us to associate each VO found in the scene to a track T during the time, taking into account problems of shape changes and occlusions. For posture classification (the main topic of this paper), the probabilistic classifier we propose exploits both frame-by-frame information of people's silhouettes and past information from the associated tracks in order to overcome overlapping and cluttering problems.

The novelty of the proposed approach is the definition of two steps:

1) a posture classification performed frame-by-frame. This classification exploits simple visual features, i.e., projections of the blob's silhouette (or VO) onto the principal axes, and a machine learning process to create probabilistic projection maps (PPMs) used in a Bayesian classifier. We called this *VO-based classification*.
2) a "temporally integrated" posture classification exploiting tracking information and thus called *track-based classification*. This is motivated by the concept of "posture state" defined in a *state-transition graph* that takes into account for the classification the reliability of the track and ac-

Manuscript received June 8, 2004. This work was supported by the Domotics for Disability Project of Fondazione Cassa di Risparmio di Modena, Italy. This paper was recommended by Guest Editor G. L. Foresti.

The authors are with the Dipartimento di Ingegneria dell'Informazione, Università di Modena e Reggio Emilia, Emilia **<<AU: POSTAL CODE?>>**, Italy (e-mail: cucchiara.rita@unimo.it).

Digital Object Identifier 10.1109/TSMCA.2004.838501

[1]*Domotics* is a neologism coming from the Latin word *domus* (home) and informatics.

1083-4427/05$20.00 © 2005 IEEE

IEEE Proof

quired knowledge of the people's average behavior in changing their posture.

This posture classifier is used to detect alarming situations, such as a person falling down and laying down for a long time. We will show that the proposed approach is capable of reliably recognizing also postures of people that differ from those used in the training set, provided they are of similar body build, and we will demonstrate that it is quite robust, even in the case of people partially occluded by furniture or other people.

The paper is structured as follows. Section II reports a brief list of the related works in the topic of human posture classification. The initial steps of the system required for the posture classifiers are detailed in Section III. Sections IV and V will describe the VO-based and track-based classifiers, respectively. Moreover, in Section IV the proposed scaling procedure and the mathematical formulation of the (PPMs are reported. The benchmark used and the experimental results achieved are depicted and discussed in Section VI. Conclusions follow in Section VII.

II. Related Works

Since the main topic of this work is human posture classification, we will not report on work related to other topics, such as motion segmentation or tracking, focusing on the relevant approaches reported in the literature on human body posture classification.

Recently, an increasing number of computer vision projects dealing with detection and tracking of human posture have been proposed and developed. An exhaustive review of proposals addressing this field was written by Moeslund and Granum in [4], where about 130 papers are summarized and classified according with several taxonomies. In particular, they consider three different application fields: 1) video surveillance; 2) control; and 3) pure analysis.

The posture classification systems proposed in the past can be differentiated by the more or less extensive use of a two-dimensional (2-D) or three-dimensional (3-D) model of the human body [4]. In accordance with this, we can classify most of them into two basic approaches to the problem. From one side, some systems (like Pfinder [5] or W^4 [1]) use a direct approach and base the analysis on a detailed human body model: an effective example is the cardboard model [6]. In many of these cases, an incremental predict–update method is used, retrieving information from every body part.

Many systems use complex 3-D models, and require special and expensive equipment, such as 3-D trinocular systems [7], 3-D laser scanners [8], thermal cameras [9], or multiple video cameras to extract 3-D voxels [10]. Due to the need for real-time performance and low-cost systems, we discarded complex and/or 3-D expensive solutions. In addition, these are often too constrained to the human body model, resulting in unreliable behaviors in the case of occlusions and perspective distortion, that are very common in cluttered, relatively small environments like a room.

A second way consists of an indirect approach that, whenever the monitoring of single body parts is not necessary, exploits less, but more robust, information about the body. Most of them extract a minimal set of low-level features exploited in more or less sophisticated classifiers. One frequent example is the use of neural networks, as in [11] and [12]. However, the use of neural networks (NN) presents several drawbacks due to scale dependency and unreliability in the case of occlusions. Another interesting example of this class is the analysis of AC-DCT coefficients in the MPEG compressed domain [13]. This has proven to be also insensitive to illumination changes, but the reported examples only classify different standing postures (with different pointing gestures), while we are interested in classifying very different postures, such as standing up and laying on the floor. Eventually, in [14], a Universal EigenSpace approach is proposed; this presents insensitivity to clothing, but it assumes that most of the possible postures (with most of the possible occlusions) have been learned, and this is far from being realizable.

Another large class of approaches are based on human silhouette analysis. Fujiyoshi *et al.* [15] used a synthetic representation (Star Skeleton) composed by outmost boundary points. A similar approach is proposed in [16] where a skeleton is extracted from the blob by means of morphological operations and then processed using a HMM framework. This approach is very promising and has the unique characteristic of also classifying the motion type, but it is very sensitive to segmentation errors and in particular to occlusions. Moreover, no scaling algorithm to remove perspective distortion is proposed, making this approach unfeasible for our target application.

Another approach based on silhouette analysis is reported in [17] and [18], where a 2-D complex model of the human body is matched with the current silhouette by genetic algorithms. In addition to the problems of segmentation errors and occlusions, this approach also suffers from dependency of the model on the view. In [1], Haritaoglu *et al.* add to W^4 framework some techniques for human-body analysis using only information about the silhouette and its boundary. First, they use hierarchical classification in main and secondary postures, processing vertical and horizontal projection histograms from the body's silhouette. Then, they locate body parts on the silhouette boundary's corners.

Our approach is similar to this one, as regards projection features, but, differently from it, is not based on an *a priori* defined model, but exploits a learning phase to build a probabilistic model of body postures.

III. Inputs to the Classifier

The posture classification is based only on the appearance of the person's body and, in particular, on its silhouette. Thus, a reliable blob-extraction algorithm must be used to provide this input to the classifier.

Our system is structured as a client–server architecture as in Fig. 1. The client–server architecture assures more flexibility and multithread programming allows us to meet real-time requirements. For further details on the architecture, please refer to [19]. The server side contains several pipelined modules: in domotic video surveillance, motion is a key aspect and, thus, object detection and motion analysis are embodied in the first module. The output of this module is the set of VOs, along with their features (shape, area, color distribution, average motion

IEEE Proof

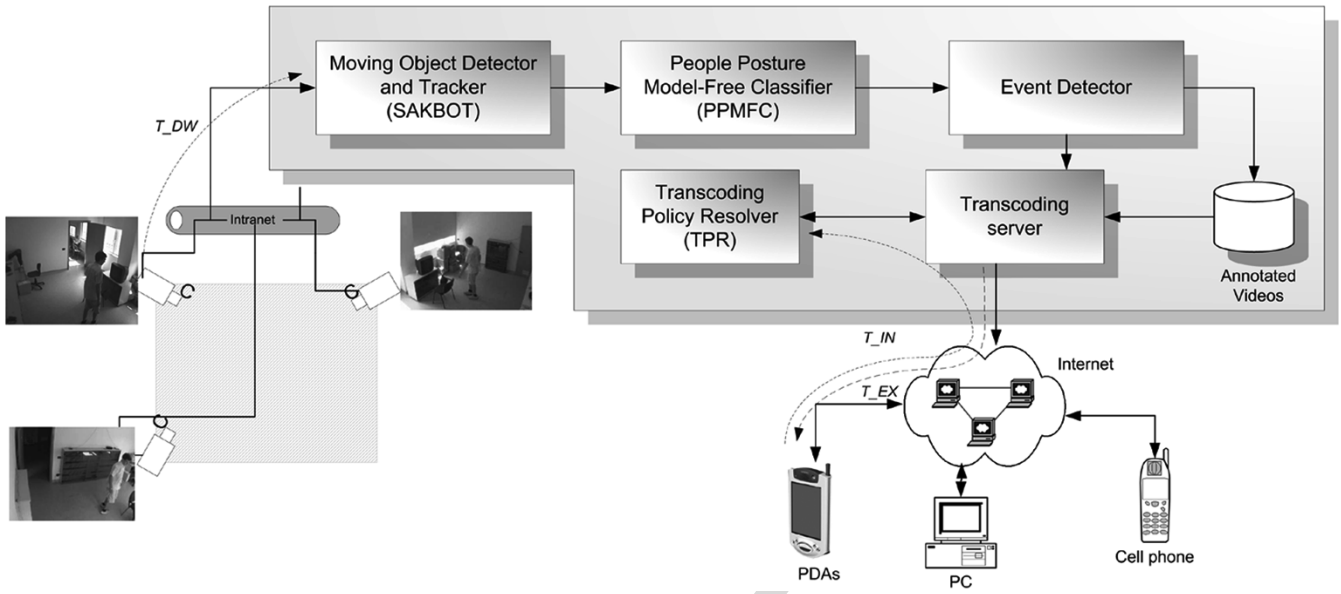


Fig. 1. Scheme of the overall architecture.

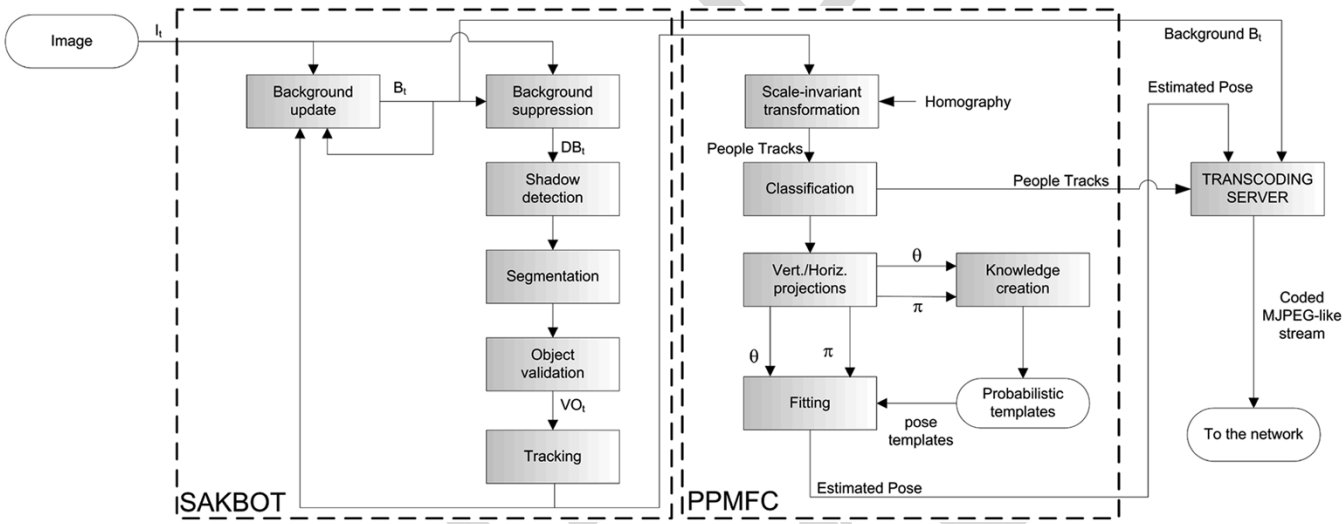


Fig. 2. Detailed scheme of the proposed system.

vector, and so on). These objects are tracked along time and processed to first classify them; the objects classified as people are further processed to detect their posture in the second module (PPMFC **<<AU: DEFINE "PPMFC">>**) and, from it, to identify a given event. Events are modeled as transitions between two states of a finite state machine representing the postures of the person. Thus, the event "falling down" is modeled as the transition between the "standing" (or "sitting") state and the "lying down" state.

The next step of semantic video transcoding is independent from the implementation of the previous modules and calls for a video adaptation to the user's requirements and device's capabilities by applying selectively coding policies depending on the current event and on the objects in the frame. Eventually, a coded stream is sent over the network. The clients can download and decode the stream to reconstruct the adapted video on their device.

In the following, we will detail a little more the people detection and the tracking modules, derived from the system called statistical and knowledge-based object detector (SAKBOT) [2], [20] and the tracking approach, partially derived from the works of Senior *et al.* [3]. Fig. 2 reports a focused version of the system depicted in Fig. 1. The motion detection embedded in the SAKBOT system is based on background subtraction and models the background using statistics and knowledge-based assumptions. In fact, the background model is computed frame by frame by using a statistical function (temporal median) and taking into account the knowledge acquired about the scene in previous frames. In practice, the background model is updated differently if the considered pixel belongs to a previously detected MVO: in this case, the background model is kept unchanged because the current value is surely not describing the background. Moreover, if an object is detected as "stopped" (i.e., the tracking system detects that it was moving and is now stationary) for more than a "timeout" number of frames, its

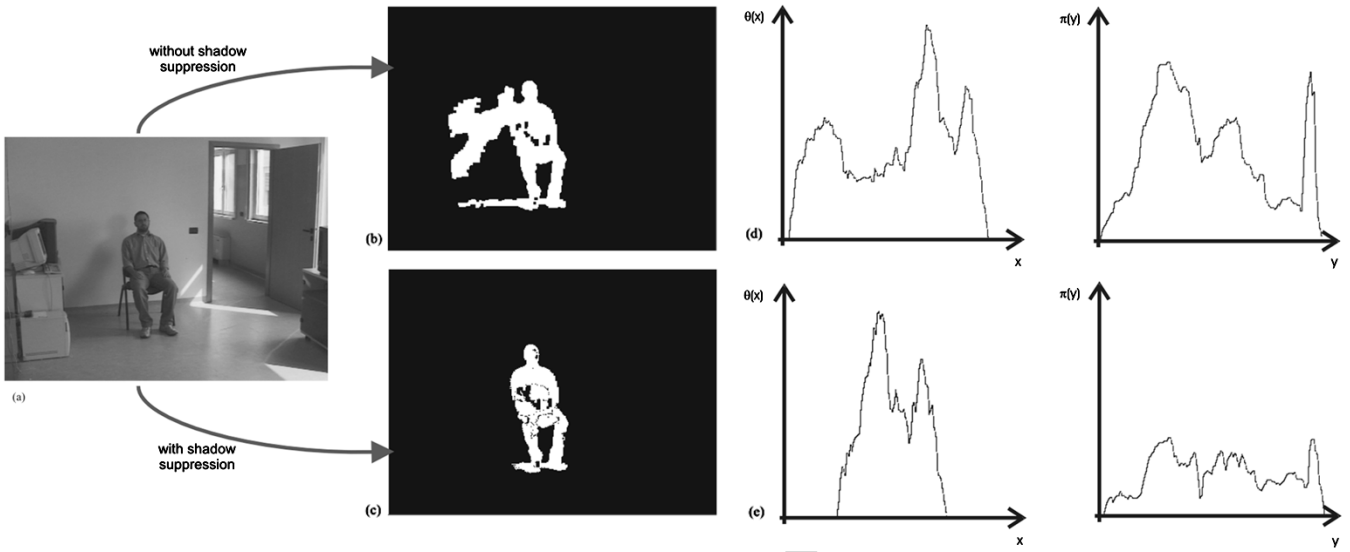


Fig. 3. Comparison of the results achieved by preserving or removing shadows: (a) the input frame, (b) the VO extracted by including shadows and (c) that obtained by removing shadows, and (d) and (e) report the corresponding projection histograms.

pixels are directly inserted into the background, without using the statistics.

SAKBOT also implements an effective shadow-detection algorithm [21]. Shadows are really critical due to many possible artificial light sources that heavily change people's silhouettes; see, for instance, where Fig. 3(a) shows the input frame, and Fig. 3(b) and (c) report the obtained silhouette without or with shadow suppression, respectively. Shadows are detected by means of an appearance model that relies on the fact that cast shadows darken the background that they cover, but slightly change the color. After detection, the shadows are used to get the above-mentioned classification and to separate them from objects. An object validation task is performed to remove small objects and to distinguish between real and apparent (ghost) moving objects. More details can be found in [2].

In conclusion, we assume to have a lower level module able to provide us, for each frame t, with a list $O^t = \{O_i^t\}$ of VO's), preclassified as people.

Each visual object is associated to a blob mask

$$B = \{b(x,y) \in \{0,1\}, x \in [0, B_x - 1], y \in [0, B_y - 1]\} \quad (1)$$

where B_x and B_y are the bounding box sizes, and $b(x,y) = 1$ if the pixel in the bounding box of (x,y) coordinates belongs to the person's blob and 0 otherwise.

The extracted VOs are processed by a tracking module that must ensure the maintenance of the tracks also in the case of occlusions due to static or moving objects (e.g., furniture or other moving people). The information extracted by the tracking module can be exploited also by the posture classifier, as we will detail in the following. As a matter of fact, there is no reliable frame-by-frame classifier based on single-perspective images able to deal with occlusions, since it cannot classify something that is not visible (such as a person behind another person).

Our tracking module is a suitable adaptation of that proposed by Senior *et al.* in [3], which suggests the use of an incremental and adaptive definition of tracks using a probabilistic and color-based appearance model of the detected blob.

In practice, it provides a set of current tracks for each frame t. A track T is defined as a tuple of values:

$$T = \{B_x, B_y, B, \mathcal{P}, \mathcal{A}, \text{status}, \text{posture}, \ldots\} \quad (2)$$

where

- B_x and B_y represent the bounding box size
- B is the blob mask as reported in (1)
- $\mathcal{P}$ is the probability track mask:

$$\mathcal{P} = \{p(x,y) \in \mathbb{R}, p(x,y) \in [0,1], x \in [0, B_x - 1], y \in [0, B_y - 1]\} \quad (3)$$

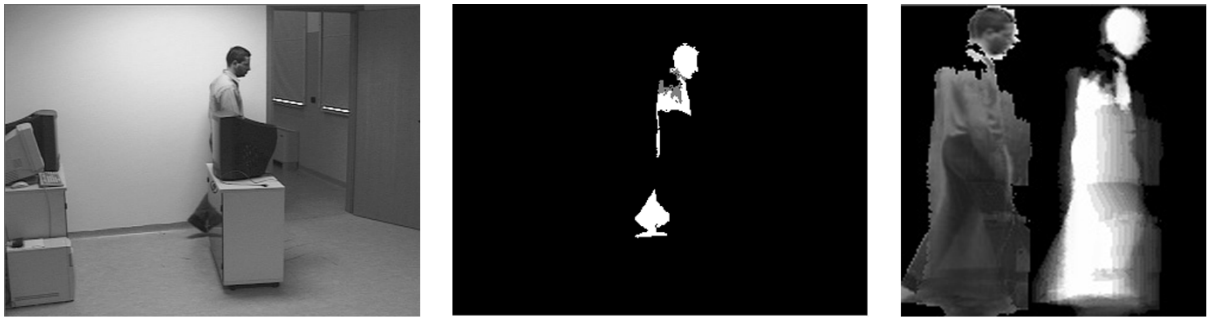
  describing the probability that the point with (x,y) coordinates belongs to the track.
- $\mathcal{A}$ is the color appearance model of the track.

$$\mathcal{A} = \{a(x,y) \in \mathbb{N}^3, a \text{ is an RGB tuple}, x \in [0, B_x - 1], y \in [0, B_y - 1]\} \quad (4)$$

- $\text{status} \in \{\text{motion}, \text{static}\}$ describes the current motion status of the track;
- *posture* is the classified posture for the track: the reliable computation of the posture is the main goal of this work.
- Other features are associated to the tracks that are not interesting in this context.

Fig. 4 describes an example of the lower level modules. Fig. 4(a) is a frame of an indoor environment; Fig. 4(b) is the blob of the VO extracted that is preclassified as a person (whose posture has to be classified); and Fig. 4(c) shows the probabilistic (right) and appareance-based model (left) computed by the tracking module. Please note that the more reliable parts have a high probability (in white), whereas the parts that have been detected as belonging to the VO in the past, but are not detected in the last frame(s), have a lower probability. The

IEEE Proof



(a) (b) (c)

Fig. 4. Examples of the output of the lower level modules: (a) input frame, (b) extracted blob, and (c) appearance model (left) and probability mask (right).

appearance model is the memory of the track integrated during the time.

The tracking module starts performing an object-to-track matching: objects extracted by the background suppression module are considered and associated with the estimated position of the tracks, that are logical objects present in the scene, by checking if their bounding box distance (BBd) is sufficiently low, with respect to a parameter experimentally set

$$\mathrm{BBd}(O_i, T_j) = \min\Big(d_B\big(c(O_i), \mathrm{BB}(T_j)\big), d_B\big(c(T_j), \mathrm{BB}(O_i)\big)\Big) \quad (5)$$

where d_B is the minimum distance between a point and a rectangle, c identifies the centroid of the object, and BB is the bounding box. This produces a object-to-track correspondence matrix and many cases arise: to cope for these, if many objects correspond to the same track, they are merged together into a single macro-object that comprises many connected components. In the context of this article, we will focus on the main part of the tracking algorithm, leaving aside an in-depth analysis of other cases like mutual occlusion handling, splitting and merging strategies. When a track corresponds to the macro-object, it is fitted to the image and its model is updated using the current object. The fitting procedure is performed starting from a previously estimated position, searching for a maximum of the function

$$\mathrm{Fit}(I, \mathcal{A}, \mathcal{P}, i, j) = \sum_{x=0}^{B_x}\sum_{y=0}^{B_y} Gp_{\mathrm{RGB}}(x+i, y+j) \times \mathcal{P}(x+i, y+j) \quad (6)$$

$$Gp_{\mathrm{RGB}}(x, y) = (2\pi\sigma^2)^{-\frac{3}{2}} e^{-\frac{\|I(x,y)-\mathcal{A}(x,y)\|^2}{2\sigma^2}} \quad (7)$$

where I is the image and σ controls the tolerance of the match.

After the fitting phase, the algorithm proceeds by updating the bounding box to the newly detected points, then the probability mask with the equation

$$\mathcal{P}_t(x, y) = \mathcal{P}_{t-1}(x, y)\lambda + (1-\lambda)B(x, y) \quad (8)$$

with $\lambda \in [0, 1]$, then updating the appearance model for the foreground pixels

$$\mathcal{A}_t(x, y) = \mathcal{A}_{t-1}(x, y)\alpha + (1-\alpha)I(x, y) \quad (9)$$

also with $\alpha \in [0, 1]$, and finally, by shrinking the bounding box to contain only pixels that belong to the probability mask over a certain threshold **<<AU: CHECK THIS SENTENCE>>**. In indoor situations, considering the average speed of people moving in the room and the frame rate, we set $\alpha = \lambda = 0.9$.

The changes in the position with respect to the previous frame are stored and used to estimate the next position. The motion model is based on a constant-speed assumption, but enforced by a segmented trajectory scheme. Starting from a reference initial position, we collect a certain number of successive motion vectors and, when we obtain a sufficient number of samples, they are linearly interpolated by finding the least square solution. The solution vector is the estimate for the motion in future frames. This solution is checked to see if the interpolation correctly describes the last vector by verifying the ratio between the two eigenvalues of the principal direction computation and also if the angle or modulus has changed much from the first value. If the solution fails these checks, a new reference position is created and a new direction can be searched. This way, an adaptive finite window is used to infer the future motion of the object.

The following sections will detail the main part of this work, that is the posture classifier.

IV. VO-Based Classification

In accordance with the literature, we define four main postures

$$\mathrm{Main_postures} = \{\mathbf{ST}\text{anding}, \mathbf{CR}\text{ouching}, \mathbf{SI}\text{tting}, \mathbf{LA}\text{ying}\}. \quad (10)$$

Since our approach is based on the histograms obtained by projecting the silhouette onto the x and y axes, and since the silhouette of people sitting with a frontal, left or right view are very different, we have split each state into three view-based subclasses (frontal, left-headed, and right-headed), thus obtaining twelve view-based postures

$$\mathrm{VB_postures} = \{\mathrm{ST}_F, \mathrm{ST}_L, \mathrm{ST}_R, \mathrm{CR}_F, \mathrm{CR}_L, \mathrm{CR}_R, \mathrm{SI}_F, \mathrm{SI}_L, \mathrm{SI}_R, \mathrm{LA}_F, \mathrm{LA}_L, \mathrm{LA}_R\} \quad (11)$$

The approach we will describe in the following is applied by the system over the twelve VB_postures. Nevertheless, we will refer to the main postures only (calling them generally postures) for the sake of simplicity, in the following.

IEEE Proof

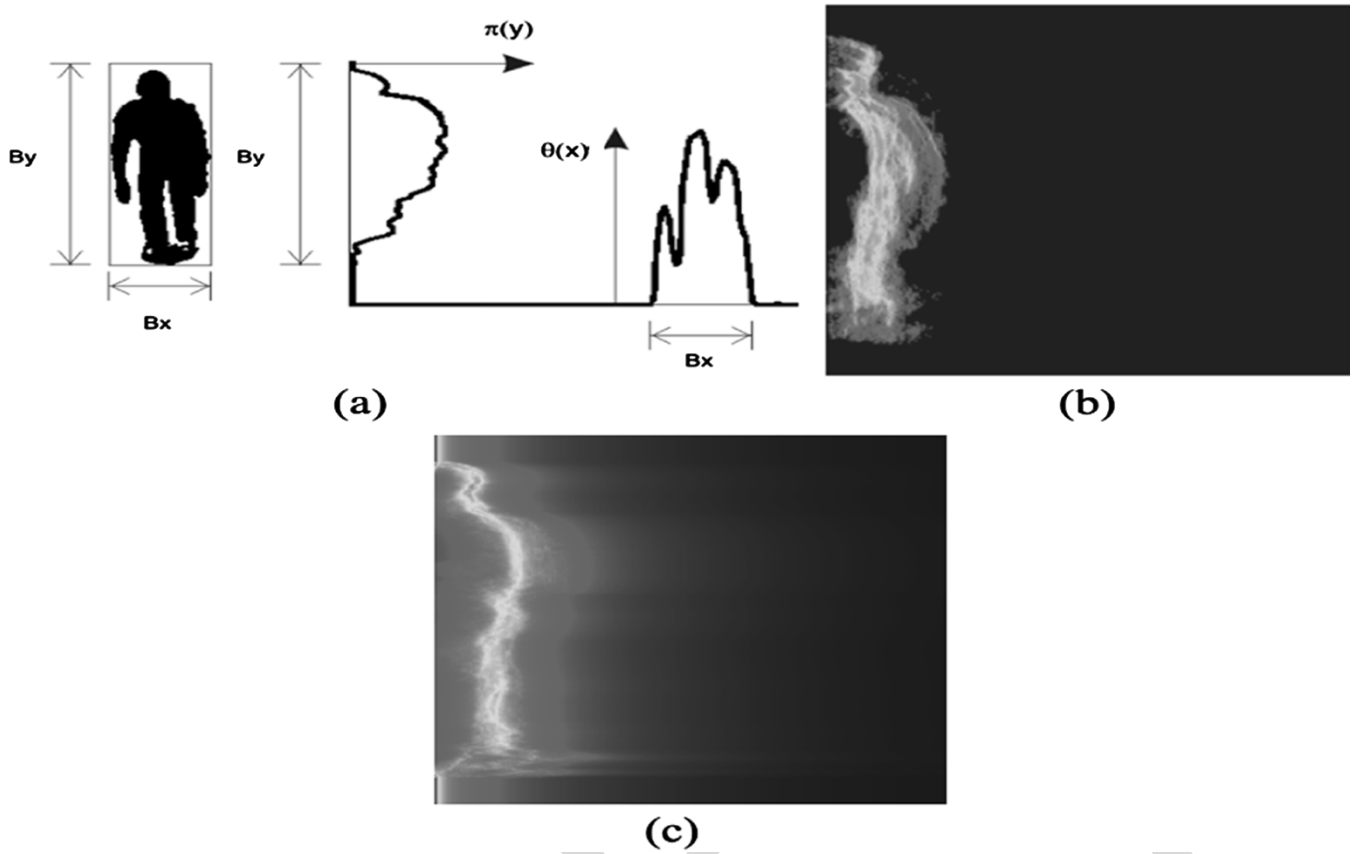


Fig. 5. Example of (a) histogram projections, (b) sparse PPMs, and (c) dense PPMs for $\text{Standing}_{\text{frontal}}$ posture. Brighter colors correspond to higher probabilities.

This classifier exploits the intrinsic characteristic of the silhouette to recall the person's posture and it is based on *projection histograms* that describe how the silhouette's shape is projected on the x and y axes. An example of projection histogram is depicted in Fig. 5(a). Moreover, in Fig. 3(d) and (e), the projection histograms obtained by including shadows or removing them are shown: as it has been already stated, shadows are particularly problematic for projection histograms based on blob's silhouette and, thus, they must be effectively removed.

Though projection histograms are an approximation with respect to a complex 3-D model, they have proven to be sufficiently detailed to discriminate between the postures we are interested in. However, these descriptors suffer two limitations:

- they depend on the silhouette's size;
- they are too sensitive to the unavoidable nonrigid movements of the human body.

The first drawback can be overcome with an initial scaling correction, while the second needs to compare the projection histograms to a suitable model, capable of generalizing the peculiarities of a training set of postures.

The following sections will report the proposed solutions to these two problems.

A. Scaling Procedure

Due to the perspective, the size of the detected VO can differ depending on the distance from the camera. This is particularly true in the case of indoor environments in which the objects moving in the scene are relatively close to the camera. To get rid of this problem, we introduced a scaling procedure.

A similar problem has been addressed in other works. For example, in [1], a normalization factor has been used to scale the histograms, but it does not take into account the actual distance of the blob from the camera and it is, thus, very sensitive to different postures. In [12], the normalization phase is mandatory to feed similar inputs to the neural network, but it has the drawback of considering all the persons as standing. Eventually, the authors of [13] developed a resizing procedure that scales all the images to the same size; this is because they are more interested in detecting where the person points than in classifying his/her posture. Moreover, this example does not take into account the actual 3-D position of the person.

Thus, to be reliable, the scaling must be correlated with the distance of the objects from the camera in the 3-D space. Assuming that the camera is fixed, we exploit camera calibration to compute the distance d between the object and the camera. Choosing a suitable normalization distance D, we define the rate s_f as a scale factor $s_f = d/D$. Applying the scale factor s_f to the blobs is analogous to "moving" all the detected objects to a fixed distance D from the camera. The d measure depends on the position of the support point (SP), that is the contact point of the object with the $Z = 0$ ground plane. SP can be described by the $(x_{\text{SP}}, y_{\text{SP}})$ coordinates in the image plane and the $(X_{\text{SP}}, Y_{\text{SP}}, Z_{\text{SP}})$ coordinates in the real 3-D world, with $Z_{\text{SP}} = 0$. In the case of a person, normally SP corresponds with

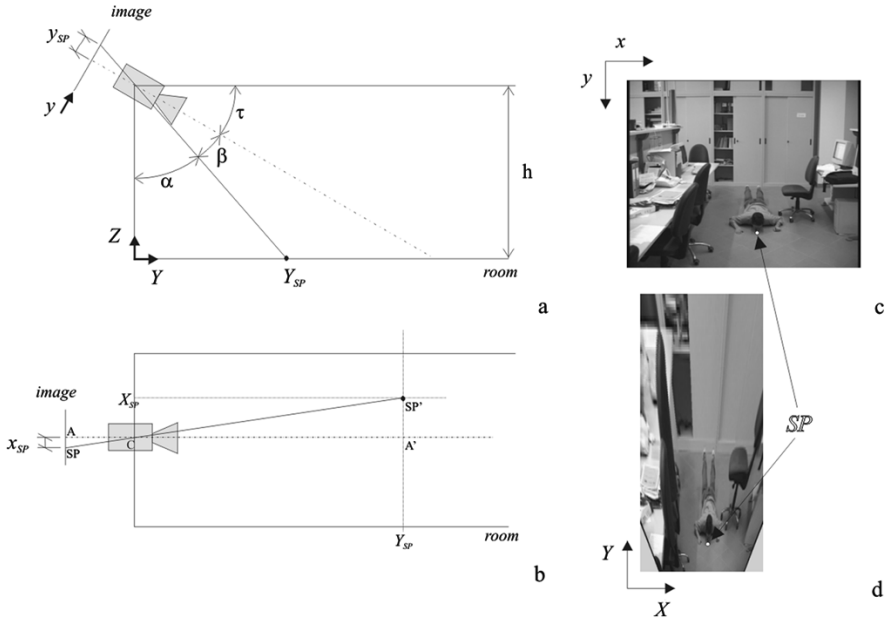


Fig. 6. Model for perspective effect removal. (a) Lateral view and (b) upper view of the pin-hole camera model. (c) The original frame. (d) The undistorted image obtained through an homography on plane $Z = 0$.

the position of the feet, but this is not necessarily true in the case of a person laying down on the floor.

Assuming that the camera's point of view is frontal, SP could be easily computed as the point with the maximum y coordinate (see Fig. 6). [We also assume that the roll angle is null. In fact, we can always return to this condition by rotating the image by the same roll angle (in the opposite direction) before processing.] If more points present the same y coordinate, SP could be randomly selected or computed as the middle point. Once we obtain the SP image coordinates, we can compute the world coordinates of this point by using the projection equations of the pin-hole camera model.

Having the height h of the camera with respect to the floor, the focal lengths f_x and f_y, and the tilt angle τ obtained by camera calibration, we can compute the coordinates of the support point SP

$$Y_{\mathrm{SP}} = h \cdot tg(\alpha), \quad X_{\mathrm{SP}} = \frac{x_{\mathrm{SP}}}{f_x} \cdot Y_{\mathrm{SP}} \tag{12}$$

with

$$\beta = \arctan\left(\frac{y_{\mathrm{SP}}}{f_y}\right), \quad \alpha = \frac{\pi}{2} - \tau - \beta, \quad d = \sqrt{X_{\mathrm{SP}}^2 + Y_{\mathrm{SP}}^2}. \tag{13}$$

After the scale factor, we assume that the sizes B_x and B_y of the blob's bounding box have been normalized. Obviously, this does not mean that all the silhouettes should have the same sizes (as in the case of a person in the standing or crouching position).

B. PPMs

The second problem of the classification based on projection histograms is that it is too sensitive to nonrigid movements of the human body model. This problem has been addressed by constructing the PPMs with a supervised machine-learning phase [22]. By means of manual annotation of the videos, the learning phase is able to build a model that represents the *memory* of the people's appearance in each posture.

The probabilistic approach included in the PPMs allows us to filter out the distracting moving parts of the body (such as the arms and the legs, not important for posture classification), thus solving the above reported problem. In fact, due to the nonrigid motion of the human body, these parts are likely to distract the classifier, resulting in misclassifications of the posture. It must be pointed out (and it will be clearer further on) that this probabilistic approach based on a learning phase shifts the problems to the completeness of the training set. We will demonstrate that, as soon as the training set contains a wide enough variety of samples, this approach achieves an average performance of 95% or above in correct classification.

The mathematical formulation of the PPMs and our static VO-based classifier follows. In accordance with the blob definition reported in (1) and similarly to [1], we can define the vertical and horizontal projection histograms (or projections), respectively, θ and π, as

$$\theta(x) = \sum_{y=0}^{B_y} b(x,y), \quad \pi(y) = \sum_{x=0}^{B_x} b(x,y). \tag{14}$$

In practice, θ and π are two feature vectors associated to the blob B. A value (or bin) of $\theta\,(\pi)$ at the position $\bar{x}\,(\bar{y})$ represents the thickness of the silhouette in the vertical (horizontal) direction. Therefore, a blob B has associated a measure $Ph_B \triangleq (\theta_B, \pi_B)$ that is used in a Bayesian classifier. Given a set of classes of postures $C = \{C_i\}, \quad i = 1, \ldots, K$, the probability to belong to the class C_i is

$$P(C_B = C_i \,|\, Ph_B) = \frac{P(Ph_B \,|\, C_B = C_i)p(C_i)}{\sum_{j=1}^{K} P(Ph_B \,|\, C_B = C_j)p(C_j)}. \tag{15}$$

IEEE Proof

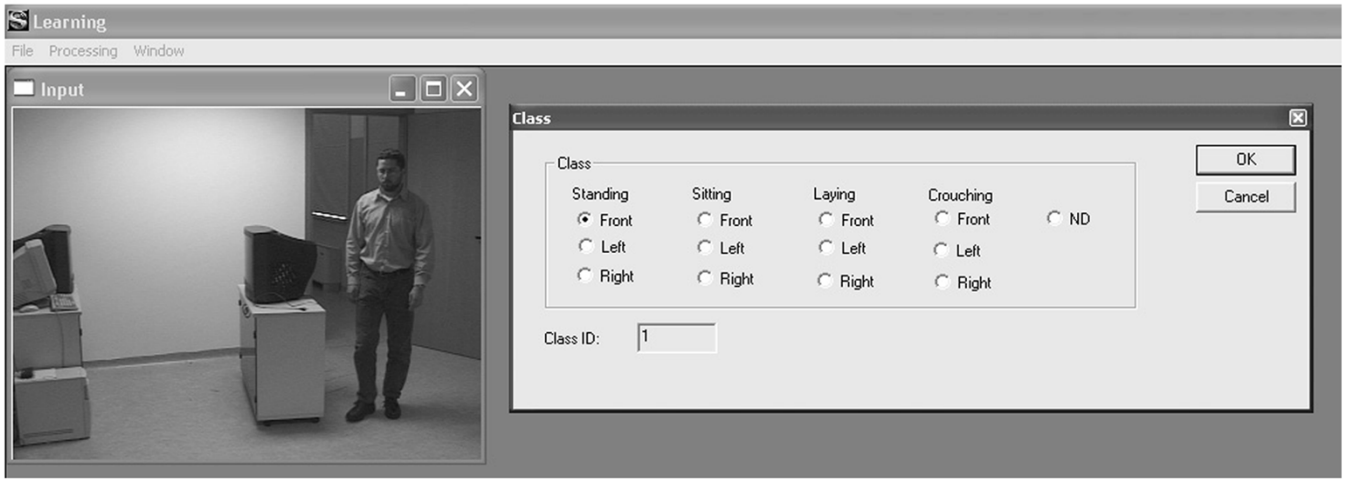


Fig. 7. Snapshot of the training procedure of our system.

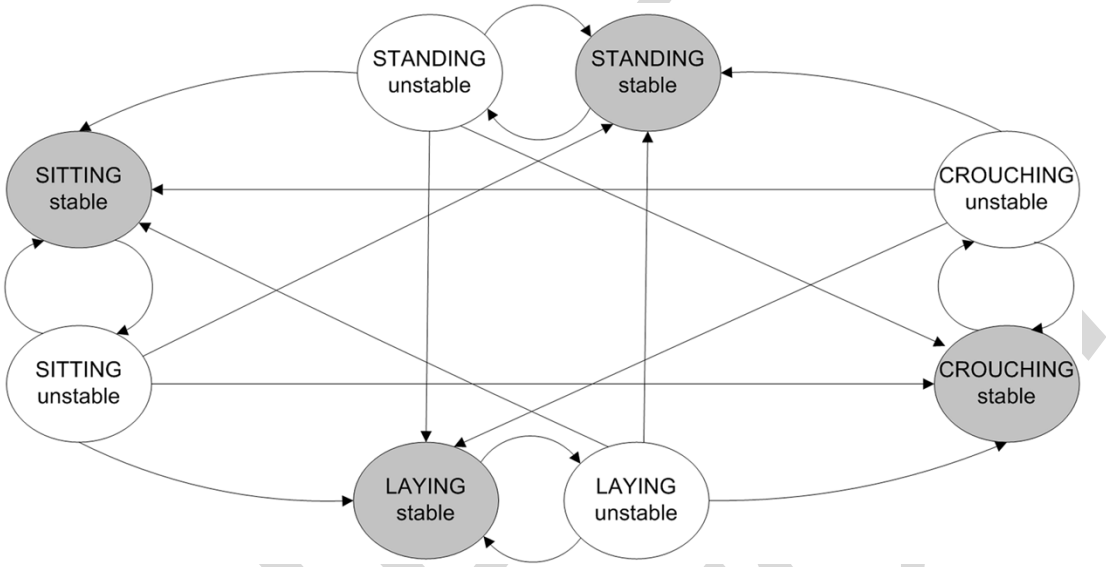


Fig. 8. Posture state-transition graph for the four main postures.

The *a priori* probabilities $p(C_i)$ can be estimated with respect to the habits of the observed person and the type of the supervised room (e.g., in a kitchen, people stay more often standing or sitting than laying down, whereas, in the case of a bedroom, it is more likely vice versa), or can be dynamically modified in accordance with the history of the blob B. For simplicity, we now assume $p(C_i)$ equal for all the classes C_i and independent from the blob B. That is

$$p(C_i) = \frac{1}{K}, \qquad i = 1, \ldots, K. \tag{16}$$

The conditional probability of having the histograms Ph_B assumed to be in the posture class C_i can be computed with the hypothesis that the θ and π measures are independent. Thus

$$\begin{aligned} P(Ph_B \,|\, C_B = C_i) &= P(\theta_B \wedge \pi_B \,|\, C_B = C_i) \\ &= P(\theta_B \,|\, C_B = C_i) \cdot P(\pi_B \,|\, C_B = C_i). \end{aligned} \tag{17}$$

If we assume the $p(C_i)$ as a constant, and neglecting the normalization factor, (15) is the *similarity function* that describes the similarity between the current silhouette and the model of the silhouette in the posture C_i.

In order to calculate (17), the similarity between each projection histogram and the correspondent model must be computed. In a previous work [19], we tested the efficacy of PPMs by computing, in a very simple way, an average distance (arithmetic mean) between the current projection θ_B (or π_B) and the models given by the PPMs.

According to the probability theory and considering θ and π projection as vector of approximately independent measures, the two terms of (17) can be computed as the probability of intersection of the events

$$\begin{aligned} P(\theta_B \,|\, C_B = C_i) &= P\left(\bigcap_{x=0}^{B_x - 1} (\theta_B(x) \,|\, C_B = C_i) \right) \\ &= \prod_{x=0}^{B_x - 1} P(\theta_B(x) \,|\, C_B = C_i) \end{aligned} \tag{18}$$

	Current state C_j			
	ST	CR	SI	LA
$p(ST\|C_j)$	0.30	0.25	0.25	0.10
$p(CR\|C_j)$	0.30	0.25	0.25	0.30
$p(SI\|C_j)$	0.30	0.25	0.25	0.30
$p(LA\|C_j)$	0.10	0.25	0.25	0.30

(a) Example of conditional probabilities to go into a state C_i coming from a state C_j

	ST_R	ST_F	ST_L
$p(ST_R\|C_j)$	0.15	0.10	0.05
$p(ST_F\|C_j)$	0.05	0.15	0.10
$p(ST_L\|C_j)$	0.10	0.10	0.10
$p(CR_x\|C_j)$	0.10	0.10	0.10
$p(SI_x\|C_j)$	0.10	0.10	0.10
$p(LA_x\|C_j)$	0.033	0.033	0.033

(b) Zoom in on the main class ST $(x = R, F, L)$

Fig. 9. Example of $p(C_i)$ dependent on the class. (a) Example of conditional probabilities to go into a state C_i coming from a state C_j. (b) Zoom in on the main class $\mathrm{ST}(x = R, F, L)$

$$P(\pi_B \,|\, C_B = C_i) = P\left(\bigcap_{y=0}^{B_y-1} (\pi_B(y) \,|\, C_B = C_i)\right) = \prod_{y=0}^{B_y-1} P(\pi_B(y) \,|\, C_B = C_i). \quad (19)$$

The probability distributions of the events $P(\theta_B(x) \,|\, C_B = C_i)$ and $P(\pi_B(y) \,|\, C_B = C_i)$ are estimated through a supervised learning phase, during which two 2-D functions $\Theta_i(x,k)$ and $\Pi_i(j,y)$ for each class C_i are created as follows:

$$P(\theta(x) = k \,|\, C_i) = \Theta_i(x,k) \quad (20)$$

$$P(\pi(y) = j \,|\, C_i) = \Pi_i(j,y). \quad (21)$$

$\Theta_i(x,-)$ and $\Pi_i(-,y)$ are the probability distributions of $\theta(x)$ and $\pi(y)$, respectively, assuming to be in the class C_i and we will refer to them as PPMs hereinafter.

The supervised learning phase for the construction of the above-mentioned maps is performed exploiting a training set (TS) of T_i 2-D blobs referred to the ith class $\mathrm{TS} = \{B_i^t\}, t = 1, \ldots, T_i$, where B_i^t are blob masks defined similarly to (1). For each B_i^t, the couple $Ph_i^t = (\theta_i^t(x), \pi_i^t(y))$ of projection histograms is computed as in (14). Then, we construct $\Theta_i(x,k)$ and $\Pi_i(j,y)$ as follows:

$$\Theta_i(x,k) = \frac{1}{T_i} \cdot \sum_{t=1}^{T_i} g\left(\theta_i^t(x), k\right) \quad (22)$$

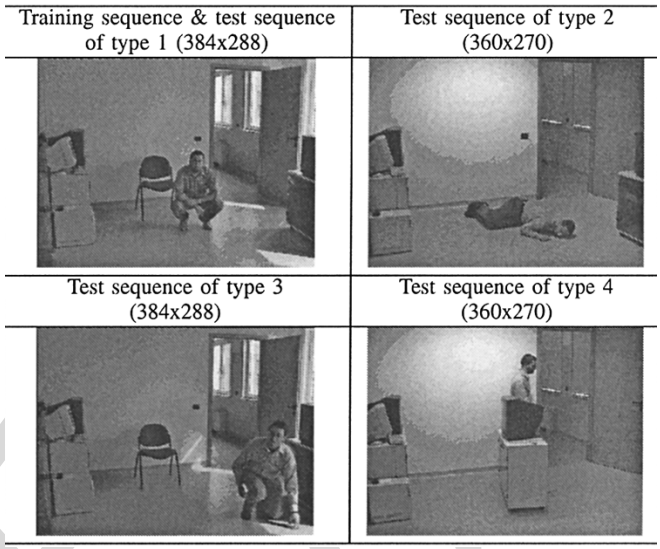
$$\Pi_i(j,y) = \frac{1}{T_i} \cdot \sum_{t=1}^{T_i} g\left(j, \pi_i^t(y)\right). \quad (23)$$

Please note that we cannot generate a PPM by either simply averaging each training set contribution, or using a Gaussian distribution, since the measures computed for each sample (i.e., for each class) are multimodal. Thus, the $g(s,t)$ function must take into account all the variations of the samples. A possible $g(s,t)$ function could be

$$g(s,t) = \begin{cases} 1, & \text{if } s = t \\ 0, & \text{otherwise} \end{cases}.$$

This function simply accumulates all the training set information without generalization and it can be acceptable only if we have an almost infinite training set. In fact, also if the current histogram is very similar to those used during the learning phase, the probability [computed as the product of (18)] could be zero, if only one bin has a value which has never occurred during the training and this is due to the sparseness of the PPMs.

TABLE I
SOME EXAMPLES FROM THE BENCHMARK SUITE



Consequently, the function $g(s,t)$ should be less "rough." We adopted the following function:

$$g(s,t) = \frac{1}{|s-t|+1}. \quad (24)$$

The number 1 in the denominator is inserted to avoid dividing by zero. Eventually, the values of the maps obtained through (24) must be normalized to obtain probability distributions. A comparison between the maps created with these two $g(s,t)$ functions is reported in Fig. 5(b) and (c).

Once the PPMs are created during the learning phase, at the testing stage, the projection histograms obtained by each blob B are compared as described with the PPM of each class and the resulting posture for the blob B is the one that maximizes the conditional probability reported in (15), i.e.,

$$\text{posture}_B = \arg \max_{i=1,\ldots,K} P(C_B = C_i \,|\, Ph_B). \quad (25)$$

Eventually, Fig. 7 shows a snapshot of the environment we used for training. Given a training video shot with a single person, we provide a simple manual annotation of the posture of the actor.

V. TRACK-BASED CLASSIFICATION

The previous section reported the VO-based classifier. We apply it over a large test set of videos and the achieved results

IEEE Proof

TABLE II
ACCURACY FOR EACH TEST TYPE

Test Type	Total Number of frames	Average Accuracy Without STG	Average Accuracy With STG	Postures St	Postures Si	Postures La	Postures Cr
Type 1	1742	98.50%	99.40%	28%	21%	40%	11%
Type 2	18222	96.90%	99.57%	6%	32%	42%	20%
Type 3	6023	90.40%	92.20%	32%	21%	22%	25%
Type 4	3141	90.80%	97.75%	35%	12%	26%	27%

show a good robustness of the approach and a very high correct classification rate [22], at least in ideal situations when the blobs are extracted perfectly. However, by exploiting knowledge embedded in the tracking phase, many possible classification errors due to the imprecision of the blob extraction can also be corrected. These errors can arise when:

- there are frequent transitions between a posture and another: in these cases, the classifier's reliability decreases;
- there are occlusions: in these cases, the person's silhouette cannot be entirely viewed and the projection histograms become less reliable;
- the illumination conditions change: blob extraction based on background suppression is severely affected by this problem, but this can be partially solved by exploiting the tracking.

The former two cases are unavoidable due to the person's behavior and the scene, and can be solved only at a higher level, whereas the latter is mainly due to the lower level tasks. In our system, all these cases are accounted for by defining a *state-transition graph* in which we use two states for each posture, one stable and the other unstable (corresponding to the above cases).

The posture state-transition graph is a graph defined as $\mathrm{Posture_STG} = (E, N)$, where the set of nodes N includes both the stable and unstable states, respectively, S_i and s_i, with $i = 1, \ldots, K$. E is the set of arcs representing the allowed transitions between two states. Fig. 8 reports the Posture_STG for the four main classes reported in Section IV. A similar graph can be derived for all the 12 subclasses.

The transitions between states are guided by two inputs:

- $\mathrm{posture}_B$, i.e., the output of the VO-based classifier of (25);
- confidence (shortened with conf), i.e., a Boolean value that gives a measure of the reliability of the classification of the VO-based classifier (*high* means a reliable classification, low unreliable).

Thus, the transitions of the STG can be modeled as follows:

$$S_i \rightarrow S_i \Leftrightarrow (\mathrm{conf} = \mathrm{high}) \wedge (\mathrm{posture} = i) \quad (26a)$$

$$S_i \rightarrow s_i \Leftrightarrow (\mathrm{conf} = \mathrm{low}) \vee (\mathrm{posture} \neq i) \quad (26b)$$

$$s_i \rightarrow S_j \Leftrightarrow (\mathrm{conf} = \mathrm{high}) \wedge (\mathrm{posture} = j) \quad (26c)$$

$$s_i \rightarrow s_i \Leftrightarrow (\mathrm{conf} = \mathrm{low}). \quad (26d)$$

S_i is the stable state of the posture i that is maintained whenever the classifier confirms the posture i with a good confidence (26a). If the confidence decreases, the classification result is no more reliable, but we maintain the posture estimate by moving into the correspondent unstable state s_i (26b). This can be due to the blob extraction errors or to blob occlusions previously mentioned. Moreover, in order to filter single-frame errors, we prevent the direct transition between two stable states (indeed, the transition $S_i \rightarrow S_j$ is not allowed). Transitions between two stable states must pass through an unstable state, at least for one frame. Obviously, this introduces a short delay in posture change detection. Please note that the loops on the states in (26a) and (26d) are not reported in Fig. 8.

TABLE III
CONFUSION MATRIX (IN PERCENTAGE) AVERAGED OVER THE FOUR TEST TYPES

Classified as	G.T. posture St	G.T. posture Si	G.T. posture La	G.T. posture Cr
St	**94%**	0%	0%	3%
Si	5%	**99%**	0%	3%
La	0%	0%	**100%**	3%
Cr	1%	1%	0%	**92%**

In this general model, all posture transitions are allowed and with the same probability. Nevertheless, in real cases some transitions are quite unlikely. For instance, a direct transition between "standing" and "laying" posture is quite improbable. To include this concept we could inhibit some transitions between unstable and stable states or we could reformulate the $p(C_i)$ of (15). In practice, we could consider $p(C_i)$ as dependent on the current state C_j. As an example, we could assume a fixed transition probability table as the one reported in Fig. 9.

This table can be the result of a study of people's average behaviors. It can obviously be improved by exploiting the training phase to fill in these tables.

To conclude this section, we must discuss how to compute the classification confidence conf above reported by taking into account tracking results. The tracking we have developed exploits a track's probability mask $\mathcal{P}$ (3) to match blobs with past tracks and to solve problems of overlapping. With it and the definition of blob reported in (1), it is possible to compute a BestFitReliability (BFR) function to measure the similarity between a track and the current blob and, as a consequence, a Boolean value conf by thresholding BFR:

$$\mathrm{BFR} = \frac{\sum_{y=0}^{B_y-1}\sum_{x=0}^{B_x-1} p(x,y)b(x,y)}{\sum_{y=0}^{B_y-1}\sum_{x=0}^{B_x-1} p(x,y)} \cdot \frac{\sum_{y=0}^{B_y-1}\sum_{x=0}^{B_x-1} p(x,y)b(x,y)}{\sum_{y=0}^{B_y-1}\sum_{x=0}^{B_x-1} b(x,y)}. \quad (27)$$

The first term in the BFR equation gives a measure of the percentage of the model that is currently found in the scene. In fact, the function is bounded between 0 and 1:0 in the case in which no pixels are found in the scene, and 1 in the case in which all the pixels are matched with the blob. Their presence

IEEE Proof

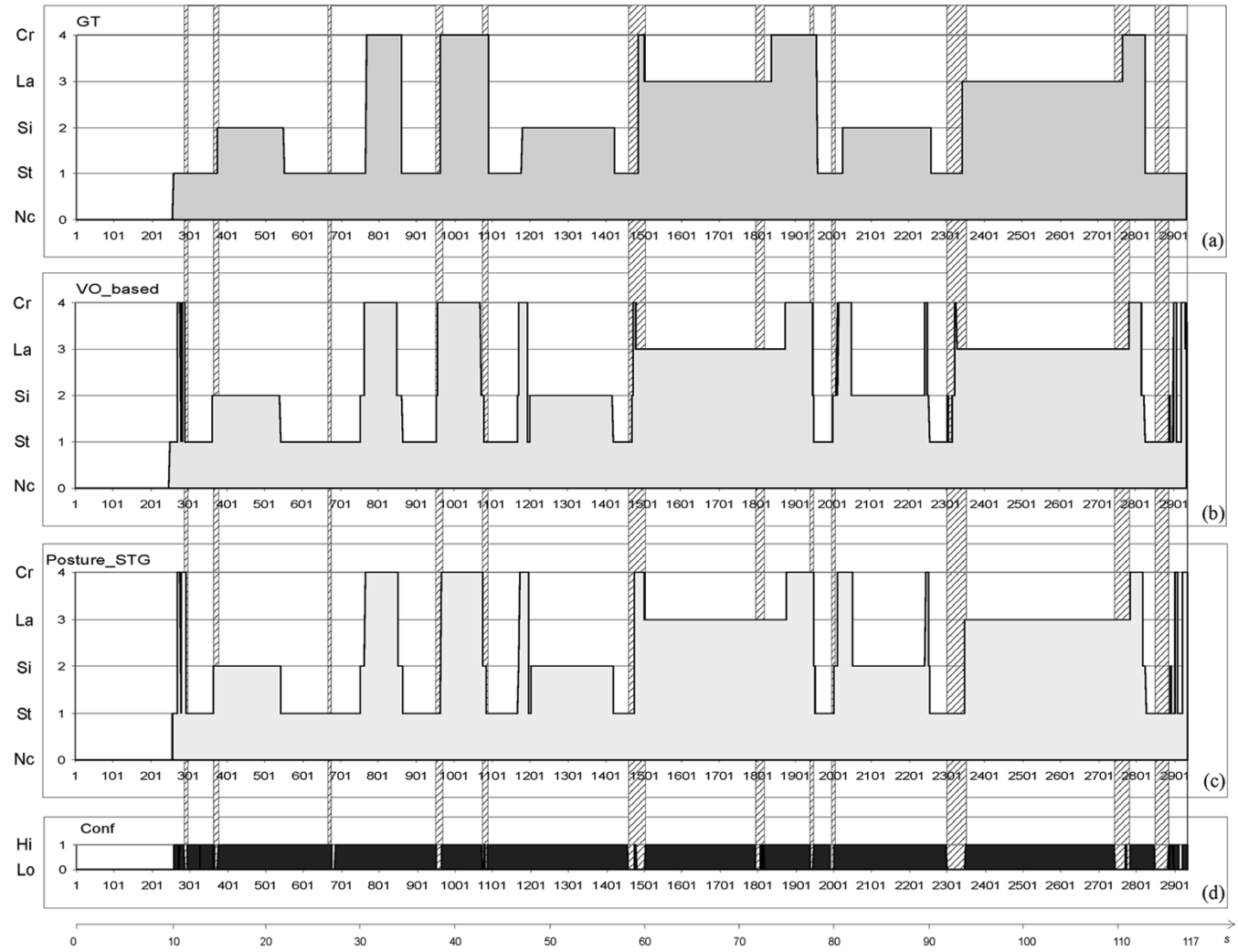


Fig. 10. Graph reporting the comparison on the video of the test type 1.

is weighted by the probability that they belong to the model. This value rises quickly to 1 in cases of sudden enlargements of the model (standing up from a crouching position and similar situations), so the second term accounts for these cases in which we see most of the model, but we are not so confident in its performance during detection. This is obtained by measuring the average probability of the pixels that are currently detected as belonging to that track. Taking the product of the two terms corresponds to the requirement that both conditions have to be satisfied. The classification confidence value conf is set to high if the BFR for the current blob is greater than a threshold.

VI. Experimental Results

The system has been developed to meet real-time constraints; the goal is to process a sufficient number of frames per second (fps) to be reactive and adaptive enough for possible alarms. The classification with the proposed method is not time consuming and the average performance in the tested videos on a standard PC (Pentium 4 with 3 GHz of RAM) is about 15 fps, including also the video acquisition, the segmentation, and the tracking steps.

For the benchmark suite, we use a large set of videos acquired in a room (where the camera has been calibrated) in different days and with different people. Table I reports some examples under different illumination conditions and with different people with different dresses. The benchmark has been selected in order to perform four tests.

1) The testing video is the same as that used for the training. This can be useful in the case of video surveillance applications for home automation in which we could suppose an initial training performed in the specific context and on the specific person.
2) The person used in the training video is the same as the testing videos', but with different clothing and in different conditions. This demonstrates the insensitivity of our system to clothing as in [14].
3) Different persons (but with similar body build)[2] are used for the testing videos;

[2]As it is clear, a classification based on the silhouette's shape is affected by the people appearance, thus children and adult people, for instance, require different PPM models. Nevertheless, it is not sensitive to the position of the people in the room due to the calibration-based scaling.

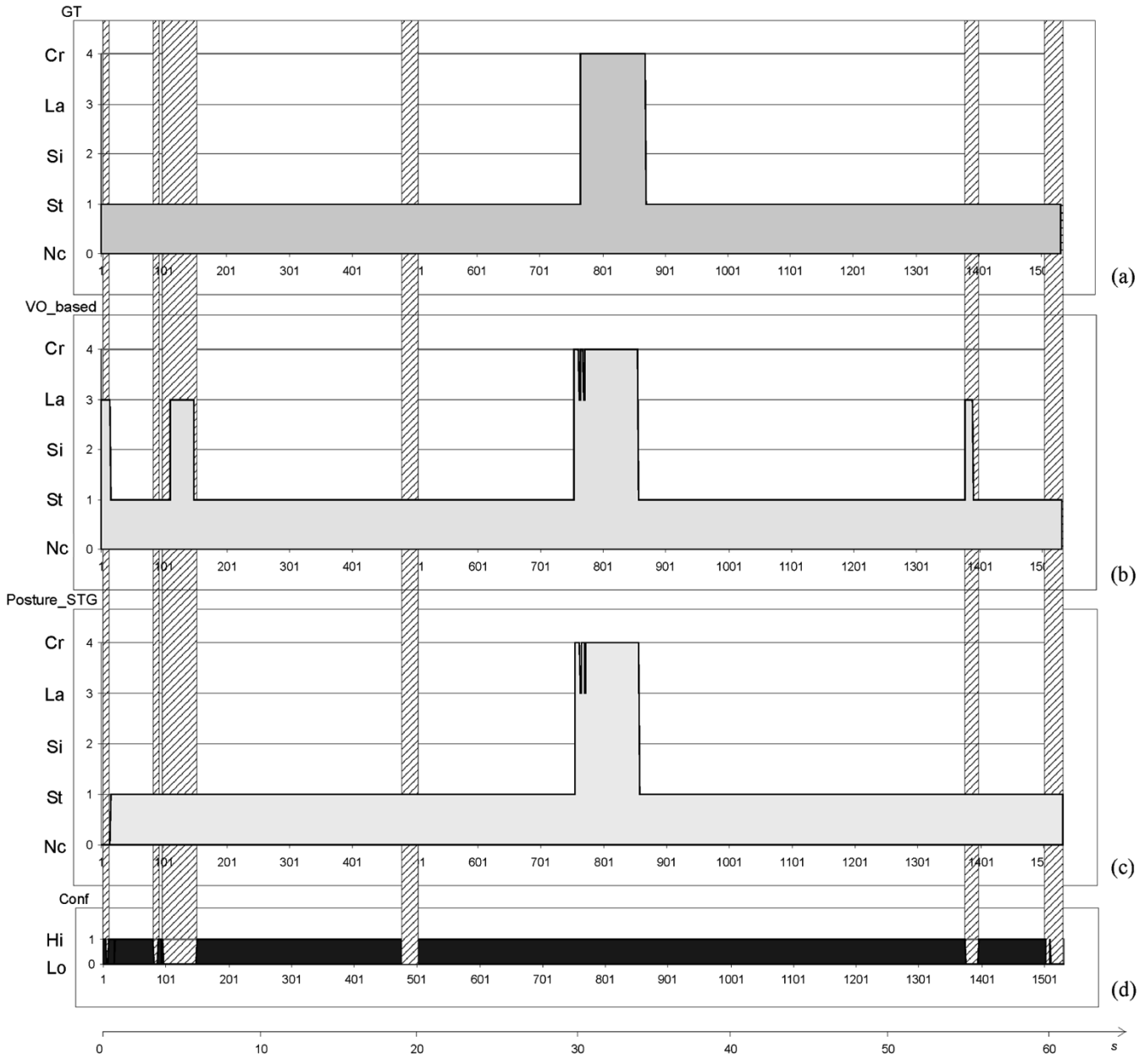


Fig. 11. Graph reporting the comparison on the video of the test type 4.

4) This is the same as 3) above, but includes occlusions to complicate the posture classification.

A detailed test of the accuracy of the system accounts for the number of correctly classified postures with respect to the total number of frames. Table II reports the average accuracy (measured as the number of postures correctly classified divided by the total number of reliable—with high confidence-ground-truthed postures) for the four types of test above mentioned. Table III illustrates the confusion matrix (in percentage) for the four main postures obtained by averaging among the four types of experiments.

In addition, the graph in Fig. 10 compares the results (on the video of type 1), with and without the state-transition graph, with respect to the ground-truth. The graphs show the postures on the ordinates (where Nc stands for "not classified") and the frame number on the abscesses. The upper graph (a) reports the manual ground-truthed classification, while the two successive graphs are (b) the classification obtained by using only the VO-based classifier or (c) the VO-based classifier with the track-based STG. The graph at the bottom (d) reports, frame-by-frame, the confidence value. This value is then used to highlight (with slashed rectangles), the sections of the video in which confidence says that the track has a low reliability.

In our tests, we set the BFR threshold to 0.7%. The BFR threshold depends on the requirements of the system and it can be set as a tradeoff between the timeliness and the reliability of the responses.

The graph in Fig. 11 compares the results on the video of type 4, with and without the state-transition graph, with respect to the ground-truth. In particular, intervals in which confidence is

low correspond to object occlusions. In this graph, the contribution of the STG is more evident than in presence of occlusions, freezes the old state, and avoids misclassifications.

As it is possible to see from Tables II and III and graph in Fig. 10, the accuracy achieved is very high and the use of the STG significantly improves the results, especially in the more challenging case in which occlusion severely affects the performance of the static VO-based classifier (see, for instance, frames 2299 to 2348 in Fig. 10). As foreseeable, the more critical posture is *crouching* that is often confused with the other postures (see the confusion matrix) and the worst result is obtained in the test of type 3 in which the crouching posture is dominant **<<AU: CHECK THIS SENTENCE>>**.

VII. Conclusion

In this paper, we proposed a human posture classifier that, starting from a visual object extracted by low level tasks and classified as a person, models human posture by means of PPMs. The classification is performed by using a Bayesian framework based on a measure of the similarity between the PPMs and the current projection histograms of the silhouette of the analyzed VO. This static, VO-based classification is further analyzed by also taking into account the integration along time. For this purpose, a state-transition graph and a confidence measure are used.

The system has been tested on different situations, changing the testing sequences in order to demonstrate the robustness of the approach to different persons, occlusions, and illumination conditions. The average accuracy is 94.15% when we use only the VO-based classifier, and rises to 97.23% if we also use the state-transition graph.

The most interesting feature of this proposal is the generality of the method that is based only on the object's appearance from a given point of view. The system can be easily generalized in many different contexts or for recognizing other human body postures.

Acknowledgment

The authors are thankful to L. Panini and G. Tardini for their valuable help in ground-truth tests and implementation.

References

[1] I. Haritaoglu, D. Harwood, and L. Davis, "Ghost: A human body part labeling system using silhouettes," in *Proc. Int. Conf. Pattern Recognition*, vol. 1, 1998, pp. 77–82.

[2] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts and shadows in video streams," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1337–1342, Oct. 2003.

[3] A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, and R. Bolle, "Tracking people with probabilistic appearance models," in *Proc. Int. Workshop Perform. Eval. Tracking Surveillance Syst.*, 2002, pp. 48–55.

[4] T. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Comput. Vision Image Understanding*, vol. 81, no. 3, pp. 231–268, Mar. 2001.

[5] C. Wren, 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, Jul. 1997.

[6] S. Ju, M. Black, and Y. Yacob, "Cardboard people: A parameterized model of articulated image motion," in *<<AU: LOCATION OR PAGE RANGE?>> 2nd Int. Conf. Automatic Face Gesture Recognition*, 1996.

[7] S. Iwasawa, J. Ohya, K. Takahashi, T. Sakaguchi, K. Ebihara, and S. Morishima, "Human body postures from trinocular camera images," in *Proc. IEEE Int. Conf. Automatic Face Gesture Recognition*, 2000, pp. 326–331.

[8] N. Werghi and Y. Xiao, "Recognition of human body posture from a cloud of 3-D data points using wavelet transform coefficients," in *Proc. IEEE Int. Conf. Automatic Face Gesture Recognition*, 2002, pp. 70–75.

[9] S. Iwasawa, K. Ebihara, J. Ohya, and S. Morishima, "Real-time human posture estimation using monocular thermal images," in *Proc. IEEE Int. Conf. Automatic Face Gesture Recognition*, 1998, pp. 492–497.

[10] I. Mikic, M. Trivedi, E. Hunter, and P. Cosman, "Human body model acquisition and tracking using voxel data," *Int. J. Comput. Vision*, vol. 53, no. 3, pp. 199–223, Jul.–Aug. 2003.

[11] S. Pinzke and L. Kopp, "Marker-less systems for tracking workin postures—Results from two experiments," *Appl. Ergon.*, vol. 32, no. 5, pp. 461–471, 2001.

[12] J. Freer, B. Beggs, H. Fernandez-Canque, F. Chevriet, and A. Goryashko, "Automatic recognition of suspicious activity for camera based security systems," in *Proc. Eur. Convention Security Detection*, 1995, pp. 54–58.

[13] B. Ozer and W. Wolf, "Human detection in compressed domain," in *Proc. IEEE Int. Conf. Image Process.*, 1998, pp. 77–82.

[14] M. Rahman, K. Nakamura, and S. Ishikawa, "Recognizing human behavior using universal eigenspace," in *Proc. Int. Conf. Pattern Recognition*, vol. 1, 2002, pp. 295–298.

[15] H. Fujiyoshi and A. Lipton, "Realtime human motion analysis lly by image skeletonization," in *<<AU: LOCATION OR PAGE RANGE?>> Proc. IEEE Workshop Applicat. Comput. Vision*, 1998.

[16] I.-C. Chang and C.-L. Huang, "The model-based human body motion analysis system," *Image Vision Comput.*, vol. 18, no. 14, pp. 1067–1083, 2000.

[17] Y. Li, S. Ma, and H. Lu., "Human posture recognition using multi-scale morphological method and Kalman motion estimation," in *Proc. Int. Conf. Pattern Recognition*, vol. 1, 1998, pp. 175–177.

[18] C. Hu, Q. Yu, Y. Li, and S. Ma, "Extraction of parametric human model for posture recognition using genetic algorithm," in *Proc. IEEE Int. Conf. Automatic Face Gesture Recognition*, 2000, pp. 518–523.

[19] R. Cucchiara, C. Grana, A. Prati, and R. Vezzani, "Enabling PDA video connection for in-house video surveillance," in *Proc. 1st ACM Workshop Video Surveillance*, 2003, pp. 87–97.

[20] R. Cucchiara, C. Grana, G. Neri, M. Piccardi, and A. Prati, "The Sakbot system for moving object detection and tracking," in *Video-Based Surveillance Systems—Computer Vision and Distributed Processing*. Norwell, MA: Kluwer, 2001, ch. 12.

[21] A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara, "Detecting moving shadows: Algorithms and evaluation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 918–923, Jul. 2003.

[22] L. Panini and R. Cucchiara, "A machine learning approach for human posture detection in domotics applications," in *Proc. IEEE Int. Conf. Image Anal. Process.*, 2003, pp. 103–108.



Rita Cucchiara (M'99) received the Laurea in electronic engineering and the Ph.D. degree in computer engineering from the University of Bologna, Bologna, Italy, in 1989 and 1993, respectively

She has been an Associate Professor of computer engineering at the University of Modena and Reggio Emilia, Emilia, Italy, since 1998. Since 2003, she has been qualified to be a Full Professor in computer engineering. Her current interests include computer vision and pattern recognition for video surveillance, domotics, medical imaging, and multimedia computer architecture. She currently serves as a reviewer for many important journals in computer vision and computer architecture.

Prof. Cucchiara is Member of the Italian chapter of the International Association of Pattern Recognition, the Italian Association of Artificial Intelligence, the ACM, and the IEEE Computer Society. She is a member of scientific committees of international conferences and editorial boards of journals.

IEEE PROOF



Costantino Grana received the Laurea and Ph.D. degree in computer engineering from the University of Modena and Reggio Emilia, Emilia, Italy, in 2000 and 2004, respectively.

He is currently a Post-Doc in the ImageLab Group, University of Modena. His research interests include medical imaging, motion analysis, and color-based application in computer vision.



Andrea Prati (S'99–A'01) received the Laurea and Ph.D. degree in computer engineering from the University of Modena and Reggio Emilia, Emilia, Italy, in 1998 and 2002, respectively

He is currently a Research Assistant with the University of Modena and Reggio Emilia. During the final year of his Ph.D. studies, he spent six months as a Visiting Scholar at the Computer Vision and Robotics Research Lab, University of California, San Diego, where he worked on a research project for traffic monitoring and management through computer vision. His research interests are mainly on motion detection and analysis, shadow removal techniques, video transcoding and analysis, computer architecture for multimedia and high-performance video servers, video-surveillance, and domotics.



Roberto Vezzani received the Laurea in computer engineering in 2002 from the University of Modena and Reggio Emilia, Emilia, Italy, where he is currently pursuing the Ph.D. degree.

His main research interest is in the area of people-posture classification.

IEEE Proof