Vision-Based System for Human Detection and Tracking in Indoor Environment

Yannick Benezeth, Bruno Emile, Hélène Laurent, Christophe Rosenberger

Noname manuscript No.
(will be inserted by the editor)

Vision-based system for human detection and tracking in indoor environment

Y. Benezeth · B. Emile · H. Laurent · C. Rosenberger

the date of receipt and acceptance should be inserted later

Abstract In this paper, we propose a vision-based system for human detection and tracking in indoor environment using a static camera. The proposed method is based on object recognition in still images combined with methods using temporal information from the video. Doing that, we improve the performance of the overall system and reduce the task complexity. We first use background subtraction to limit the search space of the classifier. The segmentation is realized by modeling each background pixel by a single gaussian model. As each connected component detected by the background subtraction potentially corresponds to one person, each blob is indepentently tracked. The tracking process is based on the analysis of connected components position and interest points tracking. In order to know the nature of various objects that could be present in the scene, we use multiple cascades of boosted classifiers based on Haar-like filters. We also present in this article a wide evaluation of this system based on a large set of videos.

Keywords Human detection · background subtraction · tracking · classification · evaluation

This work was made possible with the financial support of the Regional Council of Le Centre, the French Industry Ministry within the CAPTHOM project of the Competitiveness Pole S^2E^2.

Y. Benezeth · H. Laurent
ENSI de Bourges, Institut PRISME, 88 bd. Lahitolle,
18020 Bourges, France E-mail: yannick.benezeth@ensi-bourges.fr
E-mail: helene.laurent@ensi-bourges.fr

B. Emile
Institut PRISME, IUT de l'Indre, 2 av. F. Mitterrand, 36000 Châteauroux, France E-mail: Bruno.Emile@univ-orleans.fr

C. Rosenberger
GREYC, ENSICAEN, Université de Caen, CNRS, 6 bd. Maréchal Juin, 14000 Caen, France
E-mail: christophe.rosenberger@ensicaen.fr

1 Introduction

Potential applications of human detection systems are numerous such as low mobility persons monitoring, home automation, video surveillance, robotics, content based indexing, *etc.* For these applications, it is often necessary to detect humans before seeking highest level information. In the framework of the *CAPTHOM* project, we attempt to develop a human detection system in order to limit the power consumption of buildings and to monitor low mobility persons. Therefore, within this framework, we propose a system which has sufficiently robust performances to be usable in uncontrolled environments, the system must be easily tunable and embeddable. In other words, the system must be an optimal compromise between false detection rate and algorithmic complexity.

If the need of a reliable human detection system in videos is really important, it is still a challenging task. First, we have to deal with general object detection difficulties (background complexity, illumination conditions etc.). Second, there are other specific constraints involved with human detection. First, the human body is highly articulated. Then, human characteristics vary from one person to another (skin color, weight etc.). Clothes (color and shape) and occlusions also increase the difficulties.

Despite of these important challenges, some very promising systems have already been proposed in the litterature. We can for example quote two well-known methods proposed by Dalal and Triggs [7] and Viola *et al.* [38]. These methods usually attempt to detect humans in still images using a well-suited representation of human shapes and a classification method. In another point of view, there are methods which are based on video analysis (*e.g.* [29,34]) in which motion is used

to interpret the content of a scene doing strong assumptions about the nature of objects that could be present. In this paper, we propose a method using advantages of both approaches using tools classically dedicated to object detection in still images in a video analysis framework. We use video analysis to interpret the content of a scene without any assumption while objects nature is determined by statistical tools derived from object detection in images. To do that, we first use background subtraction to detect objects of interest. As each connected component detected potentially corresponds to one person, each blob is then indepentently tracked. The nature of these tracked objects is determined using object recognition methods in the video analysis framework.

In the following, we first present the state of the art in human detection, we then present an overview of the proposed method. Each step of the method is detailed in sections 4, 5 and 6 giving practical details about the implementation. An evaluation, based on a wide video dataset is finally presented.

2 Previous works on human detection

Human detection has received many attention in the past few years because of the huge demand of robust systems for smart surveillance, robotics, content based indexing, automotive safety *etc.* As mentioned by Gavrila in its review paper [15], human activity analysis methods can be divided in three categories, namely (1) the 3D approaches, (2) the 2D approaches with explicit shape model, and (3) the 2D approaches without explicit shape model. Methods in (1) and (2) attempt to recover 3D/2D body parts and posture on pre-localized blobs [41,22] often resulting in a stick-figure representation. While those methods do not require a training phase, they need strong assumptions on the content of the scene and often require the use of non-trivial mathematical models. On the other hand, methods in (3) detect people without explicitly locating their body parts. In fact, based on a training database, these methods extract features (*e.g.* edges, gradients, shape, wavelet coefficients etc.) and, following a recognition step (*e.g.* SVM, Adaboost etc.) separate human from non-human shapes [31,7,38]. A good overview of recent techniques and quantitative comparison can be found in [33].

In this third case, the choice of the representation is very important while very difficult because this representation has to be invariant to lighting conditions, gait, occlusions *etc.*. In other words, the representation must have a large inter-classes variability but a small intra-class one. This representation can be global [17] or local [25,31,7]. Global approaches are widely used in pattern recognition but are rarely used in human detection principally because the human body is highly articulated. Nevertheless, Gavrila *et al.* [17] proposed a well-known pedestrian detection system called *protector* based on edge representation and the Chamfer distance. The *VASM* [6] video-surveillance system which uses simple explicit features about detected objects also uses a global representation. Local representations are less sensitive to occlusions. One possible approach is to use a local and sparse representation of human based on local features [26,3,2] in a bag of words framework [13]. Another very popular approach in human detection is the use of a local and dense representation. Oren *et al.* [30] and Papageorgiou and Poggio [31] have first proposed an object detection system, applied in the case of human detection, based on a set of Haar-like filters. Humans are thus described by an over-complete dictionary of difference in intensity of adjacent areas. Viola and Jones [38,23] use Haar-like filters. Dalal and Triggs [7,8] made a serious breakthrough in human representation with the histograms of oriented gradients (HOG). These two mehods are widely used in the literature but we can however quote other representations. For example, Wu *et al.* [42] propose an edgelet representation, Utsumi and Tetsutani [37] use a representation based on a statistical analysis of the distance from Mahalanobis between various parts of the image. More recently Gao *et al.* [14] have proposed the Adaptive Contour Feature (ACF) , and Tuzel *et al.* [36] use covariance features for human detection.

In order to deal with partial occlusions, methods using part-based representation could be used. In this case, the classifier seeks each human body part independently and then fuse detection results [42,44,12,36].

All approaches presented above rely only on static image features despite the potential of motion information for people detection. So, other methods have proposed to use temporal information. For example, Dalal and Triggs [8] have extended their initial histogram of oriented gradients [7] with temporal information. Viola and Jones have also extended [39] Haar-like filters with temporal information. This approach has shown superior quantitative performance in [40].

After the description of human appearance in the feature space, a supervised learning algorithm is commonly used to learn the classification function. If one has a large number of examples of positive images and negative ones, the training algorithm seeks a function able to separate the positive examples and the negative ones within the feature space. While the huge majority of methods are based on SVM (*e.g.* [7,12]) or on boosting (*e.g.* [38]), other approaches have been used. For example, Wu and Nevatia [43] learn a tree structured clas-

sifier. Gavrila [16] proposes a tree structured Bayesian approach that builds on offline clustering of pedestrian shapes. Methods presented above treat the problem of data partitioning and classifier learning separately, Wojek *et al.* [40] propose another approach by using the *MPLBoost* classifier that simultaneously learns the data partitions and a classifier for each partition.

Other methods are based on a different scheme. In these methods, a background subtraction is firstly done in order to detect regions of interest, then based on the detected objects, a model is learned and used for tracking and activity recognition. We can quote here the most well-known methods. Firstly, Stauffer *et al.* [34] build a gaussian mixture model for background subtraction and then learn various pattern of activities based on a codebook generation. Oliver *et al.* [29] propose a computer vision and machine learning system for modeling and recognizing human behaviors in a visual surveillance task. In this case, they propose an eigenbackground model to detect objects in the scene and then a *HMM* framework for activity analysis. Elgammal *et al.* [10] have proposed a non-parametric background subtraction for initializing a part-based segmentation and tracking algorithm. Haritaoglu *et al.* [19] have proposed the W^4 video-surveillance system in which they use background subtraction to initialize body-part segmentation and other appearance models. Wren *et al.* [41] use a single gaussian model for the background subtraction which initialize the person model used for gesture recognition. These methods usually make assumptions about the objects that could be present in the scene.

Methods usually employed to detect human are designed for detection on still images. Other methods, based on video analysis, make strong assumptions about the nature of objects present in the scene. The method proposed in this paper takes advantages of both approaches.

3 Overview of the proposed method

The method presented in this paper is based on three different steps: change detection, tracking and classification.

We firstly perform change detection, this step is very useful as it permits to reduce the search space of classifiers localizing regions of interest in the image. It also permits to reduce the number of false detections. We choose to model each pixel in the background by a single gaussian distribution, so the detection process is a simple probability density function thresholding. This simple model presents a good compromise between detection quality, computation time and memory requirements. We update the background model at three different levels. We firstly proceed at the pixel level updating each pixel with a temporal filter allowing to consider long time variations of the background. Then, we proceed with an image-based update to deal with global and sudden variations. Finally, we do an object-based update to deal with the entrance or the removal of static objects.

Then, once we know the list of connected components detected in the current image, we build the history of their displacements in the image plane. One connected component potentially corresponding to one person, we track each moving object present in the scene. This displacement history is very useful since it enables us to largely increase performances of the overall system. This is done with a the combination of a connected components analysis and the tracking of points of interest.

Once objects of interest are detected and tracked, we determine the nature of these objects answering the following question: do we follow a human? With this intention, we use the classification method defined by Viola and Jones [38]. This method is based on Haar-like filters and adaboost. We finally build a confidence index on the membership of each object to the "human" class.

4 Background subtraction

Background subtraction simplies further treatments by locating areas of interest in the image. With a model of the environment and an observation, we attempt to detect what changed in the current frame. It is a very important step because the following steps will be based on this result. For our application, areas of interest are those in the image where there is a strong probability to detect a person. We present in this section the background subtraction method used in our system.

4.1 Background model

We presented in [4] a comparative study of various background subtraction methods. In this previous work, we argue that simple background subtraction methods (*e.g.* [41]) are often at least as performing than complex ones (*e.g.* [10,29,35]) when considering the detection quality and also computation time and memory requirements. As we work in indoor environments, we argue that a multimodal model (*e.g.* in [34]) is not required in our application. Moreover, the gaussian distribution allows to weight the difference between the

current image and the mean by the covariance matrix which directly depends on the amount of noise.

From this observation, we chose to model each pixel of the background by a single gaussian distribution [41]. For each pixel s at time t, the background model $B_{s,t}$ is composed of the mean $\boldsymbol{\mu}_{s,t} = \{\mu_{r,s,t}, \mu_{g,s,t}, \mu_{b,s,t}\}$ and the covariance matrix $\boldsymbol{\Sigma}_{s,t}$. We assumed that $\boldsymbol{\Sigma}_{s,t}$ is diagonal:

$$\boldsymbol{\Sigma}_{s,t} = \begin{pmatrix} \sigma^2_{r,s,t} & 0 & 0 \\ 0 & \sigma^2_{g,s,t} & 0 \\ 0 & 0 & \sigma^2_{b,s,t} \end{pmatrix}, \tag{1}$$

where $\sigma^2_{r,s,t}$, $\sigma^2_{g,s,t}$ and $\sigma^2_{b,s,t}$ correspond respectively to variance of red, green and blue components of pixel s at time t. The background model is initialized with $\boldsymbol{\mu}_{s,0} = \boldsymbol{I}_{s,0}$ and the variance of each color component is initialized with the value σ^2_0. The Mahalanobis distance is used to compute difference between the current image and the model:

$$d_M(\boldsymbol{I}_{s,t}, B_{s,t}) = (\boldsymbol{I}_{s,t} - \boldsymbol{\mu}_{s,t})\boldsymbol{\Sigma}^{-1}_{s,t}(\boldsymbol{I}_{s,t} - \boldsymbol{\mu}_{s,t})^T. \tag{2}$$

The detection is finally done thresholding the Mahalanobis distance:

$$\mathcal{X}_{s,t} = \begin{cases} 1 \text{ if } & d_M(\boldsymbol{I}_{s,t}, B_{s,t}) > \tau_1 \\ 0 & \text{whereas.} \end{cases} \tag{3}$$

where τ_1 is a threshold and $\mathcal{X}$ is the foreground motion mask. If $\mathcal{X}_{s,t}$ equals 1, the pixel s belongs to the foreground (or region of interest), whereas if $\mathcal{X}_{s,t}$ equals 0, the pixel s belongs to the background, *i.e.* the static part of the image.

By making the assumption that the covariance matrix is diagonal, the amount of memory used by this method is relatively light. Indeed, the model $B_{s,t}$ is composed only of 6 values per pixel (3 values for the means and 3 values corresponding of the diagonal terms of the covariance matrix).

4.2 Background model update

In real applications, the scene is never completely static. The model must be sufficiently flexible to adapt itself to the various changes of the environment:

1. slow variations of illumination,
2. sudden variations of illumination,
3. the addition or the removal of static objects.

For the first variation, in order to deal with slow variations of the illumination, caused for example by natural change of daylights, the model is updated as follows:

$$\boldsymbol{\mu}_{s,t+1} = (1-\alpha).\boldsymbol{\mu}_{s,t} + \alpha.\boldsymbol{I}_{s,t} \tag{4}$$

and diagonal terms of the covariance matrix are updated with:

$$\begin{cases} \sigma^2_{r,s,t+1} = (1-\alpha)\sigma^2_{r,s,t} + \alpha(I_{r,s,t} - \mu_{r,s,t})^2 \\ \sigma^2_{g,s,t+1} = (1-\alpha)\sigma^2_{g,s,t} + \alpha(I_{g,s,t} - \mu_{g,s,t})^2 \\ \sigma^2_{b,s,t+1} = (1-\alpha)\sigma^2_{b,s,t} + \alpha(I_{b,s,t} - \mu_{b,s,t})^2 \end{cases} \tag{5}$$

The threshold α is a parameter related to the speed at which new observations are taken into account.

Then, for sudden variations of illumination, we use the following criterion:

$$\Omega = \frac{\sum_{s \in S} \mathcal{X}(s)}{S} \tag{6}$$

where S is the number of pixels in the image and $\sum_{s \in S} \mathcal{X}(s)$ corresponds to the number of foreground pixels. If $\Omega > \tau_2$, the percentage of detected foreground pixels changed is too important compared with the image size so the model is reinitialized with $\boldsymbol{\mu}_{s,t} = \boldsymbol{I}_{s,t}$. By doing this, the model is sufficiently flexible to adapt itself to brutal changes of illumination.

For variation 3, a third update is carried out on the object level. This update is useful since it allows to quickly reinitialize the background model face to some specific situations like removal of a "static" object or introduction of a "static" object. Without this update, in the case of an object removal, the object which is not present in the scene anymore is still present in the background model. In the case of an object adding, the detected object, although static, is not in the model. Both situations generate detection errors. Even if the difference between the model and the environment will be slowly reduced by the temporal filter, for our application, we chose to force the update so that the phantom of the mobile object disappears quickly. It means also that static objects are quickly included in the background model. Thanks to further steps described below, we are able to determine the nature of objects and their positions in previous frames. If an object, detected by the background subtraction, is static and is regarded as not being human during a preset number of images, the background model of its corresponding shape is reinitialized using the current image.

Therefore, by carrying out an update with the three different levels, we are able to manage the most current variations of the environment.

4.3 Post-processing

Objects detected by background subtraction correspond ideally to a compact area with smooth borders. False detections are often distributed on all the image and correspond to small clusters of isolated pixels. We use a set of morphological operations to remove isolated pixels and to fill holes in the foreground image. We use a mask of size 3×3 and we use an *opening* followed by a *closing* [5].

Then, foreground pixels are gathered in connected components [1]. In the ideal case, one connected component corresponds to one object of the scene. The figure 1 presents an illustration of a result obtained after background subtraction, filtering and connected components gathering (one color represents one connected component).

Fig. 1 Example of result obtained after background subtraction and post-processing (one color per object).

5 Tracking

In the previous section, we explained how we obtain the list of blobs (connected components) present in each image. Now, we wish to know a history of the displacements of these blobs in the image plan. One blob corresponding potentially to one object, we wish to track independently each object present in the scene by assigning to it a label consistent in time. This history is very useful since it enables to largely increase the performance of the global system. The history allows to smooth in time classification errors. As explained previously, constraints concerning the algorithm complexity and the amount of memory used are very important for our application. It thus does not seem adequate to use a complex model of each object.

It is possible to initialize the tracking process with the result of the human detection classifier (detection carried out in a sliding window framework *e.g.* [7]) or with the connected component detected with the background subtraction. According to the results of human detection in a sliding detection framework, presented for example in [9], performances of these detectors do not seem sufficient to initialize the tracking for robust applications. It thus seems more judicious to use connected components obtained with background subtraction.

A tracking method based directly on connected components implies some difficulties. For example, when two distinct objects are very close, they form only one connected component and at the opposite, the same object can be represented by several blobs if there is a partial occlusion or if there are holes in the foreground mask. So, to deal with these common cases, we use in addition points of interest tracking. At every time t, we have a list of present blobs and a list of tracked objects in previous frames. So, we attempt to make the matching between these two lists. To do that, we use the matching matrix $\mathcal{H}_t$ defined with:

$$\mathcal{H}_t = \begin{pmatrix} \beta_{1,1} & \dots & \beta_{1,N} \\ \vdots & \ddots & \vdots \\ \beta_{M,1} & \dots & \beta_{M,N} \end{pmatrix} \tag{7}$$

where M corresponds to the number of tracked objects and N to the number of blobs for the current frame. $\beta_{i,j} = 1$ if the tracked object i matched with the blob j, whereas $\beta_{i,j} = 0$.

Each tracked object is characterized by a set of points of interest. These points are tracked, frame by frame. The position of these points, regarding connected components, enables to match tracked objects with detected blobs. The tracking of points of interest is carried out with Lucas and Kanade's method [27]. Two constraints are added to the original method:

1. each point of interest must be on a foreground pixel. Otherwise, this point is removed from the list and a new one is created.
2. When a new interest point is created, we impose a constraint concerning the distance with other points in order to have homogeneous distribution in space of points of interest on the object.

The matching between tracked objects and blobs is made calculating, for each blob, to which object points belong. Let $\gamma_{i,j}$ be the number of points belonging to the i object present on the blob j:

$$\begin{cases} \beta_{i,j} = 1 & \text{if } \gamma_{i,j} > \tau_3, \\ \beta_{i,j} = 0 & \text{whereas.} \end{cases} \tag{8}$$

The threshold τ_3 directly depends on the number of points used to represent an object. In practice, the threshold τ_3 is fixed at 25% of the number of points of interest per object.

An example is given in figure 2 where there are five tracked objects (represented by their points of interest) and five blobs detected. With equation 8, we are able to build the matching matrix $\mathcal{H}_t$. Since there are five objects and five blobs, the matrix $\mathcal{H}_t$ is a 5×5 matrix:

$$\mathcal{H}_t = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (9)$$

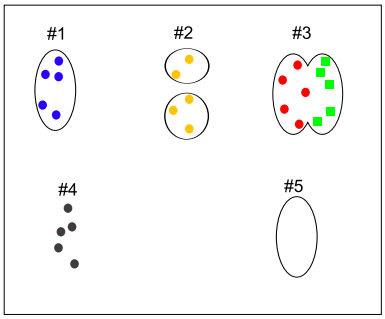

Fig. 2 Illustration of the five cases considered in the tracking. Points represent tracked objects while ovals represent detected blobs.

The figure 2 presents the 5 cases considered in our tracking method. The behavior adopted in each one of these 5 cases is described below.

1. **Matching:** This first case appears when only one blob corresponds to one object. So, we simply update coordinates of the tracked object by those observed at time t.
2. **Splitting:** In this case, an object is represented by several blobs. This can be due to partial occlusions or when several distinct objects sufficiently close in the previous frames to be fused, move away to each other on the current frame. In order to make the distinction between these two cases, each tracked object i has a variable λ_i which corresponds to the number of objects that represents the object i. $\lambda_i = 1$ means that the object i represents one object. Thus, if $\lambda_i > 1$ and the object i is represented by several blobs, we split the object i in several objects and decrease the value of λ_i. If $\lambda_i = 1$, we update the object i by the union of the corresponding blobs (with a constraint about the distance between blobs).
3. **Fusion:** In this case, one blob represents several objects. We do not fuse objects immediately. Coordinates of each object will be assigned with coordinates of the bounding box of the blob but each object will keep its own points of interest during a few tens images. Then, objects are removed and a new object k is created. New points of interest are initialized and the value λ_k is initialized with the sum of the λ of fused objects. This enables to track independently the objects of the scene even with short mutual occlusions.
4. **Deletion:** In this case of figure, an object does not correspond to any blob. If it is the case during several frames, this object is simply removed from the list.
5. **Creation:** In this case, one blob does not correspond to any object of the list, then a new object i is created, its points of interest and the value of λ_i are initialized. We assume $\lambda_i = 1$.

We present in the figure 3 an example of tracking result with a partial occlusion.

6 Classification

We present in this section the classification method used in our system. First, we present briefly the Viola *et al.*'s method [38], then we detail the part-based classification and the confidence index.

6.1 Adaboost and Haar-like filters

In order to recognize humans from any other moving objects or false detection of the background subtraction algorithm, we use the Viola *et al.*'s method [38]. This choice has been motivated by the good performance of this method and especially because of its relatively low computation cost. Actually, its cascade architecture permits to quickly reject false examples and integral images allow to compute Haar-like features with just few operations. Here, 14 Haar-like filters are used and, as shown in figure 4, those filters are made of two or three black and white rectangles. The feature values x_i are computed with a weighted sum of pixels of each component.

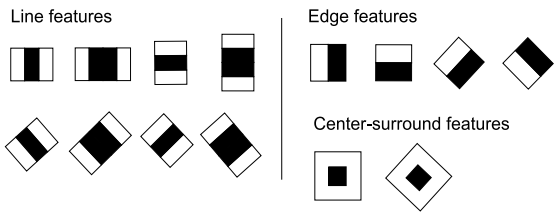

Fig. 4 Haar-like filters used for human detection.

Fig. 3 Illustration of tracking result with a partial occlusion. The first row corresponds to input images with interest points associated with each object (one color per object) and the second row corresponds to the tracking result with a label consistent in time for each object.

Each feature x_i is then fed to a simple one-threshold weak classifier f_i:

$$f_i = \begin{cases} +1 & \text{if } x_i \geq \tau_i \\ -1 & \text{if } x_i < \tau_i \end{cases} \quad (10)$$

where $+1$ corresponds to a human shape and -1 to a non-human shape. The threshold τ_i corresponds to the optimal threshold that minimizes the misclassification error of the weak classifier f_i estimated during the training stage. Then, a more robust classifier is built with several weak classifiers trained with a boosting method [32]:

$$F_j = sign(c_1 f_1 + c_2 f_2 + \ldots + c_n f_n). \quad (11)$$

Then, a cascade of boosted classifiers is built (cf. figure 5). F_j corresponds to the boosted classifier of the j^{th} stage of the cascade. Each stage can reject or accept the input window. Whenever an input window passes through every stages, the algorithm labels it as a human shape.

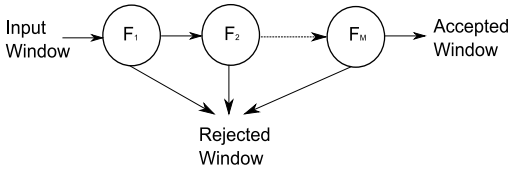

Fig. 5 Cascade of boosted classifiers.

An area of interest around the tracked object is defined with a margin d on each side of its bounding box. This area of interest is analyzed by the classifier with various positions and scales. The figure 6 presents the bounding box of one detected object and the area of interest surrounding it.

In a practical way, the classifier analyzes the area of interest with a shift of 2 pixels in the horizontal and vertical direction. As the size of the person potentially present is not known *a priori* and the classifier has a fixed size, the area of interest is analyzed several times by modifying its scale. The size of the area of interest is divided by a factor of 1.2 between two scales.

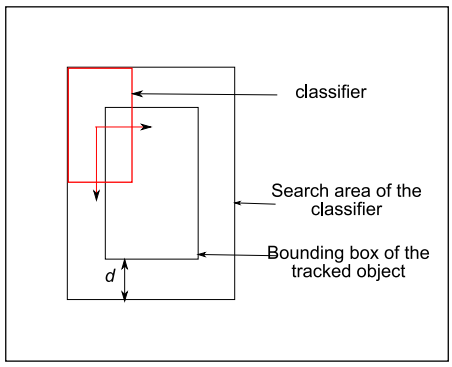

Fig. 6 Illustration of the search area analyzed by classifiers.

By using a sliding window on several scales and positions, there are logically several overlapping detections which represent only one person. To fuse overlapping detections, Gu *et al.* [18] use for example the *Mean-Shift* algorithm. For computation speed, we use the average of results which intersect. The criterion of the competition Pascal [11] is used to detect the intersections:

$$Pas(B_i, B_j) = \frac{card(B_i \cap B_j)}{card(B_i \cup B_j)} \quad (12)$$

where B_i and B_j represent bounding boxes of detections i and j. If $Pas(B_i, B_j) > \tau_4$, detections i and j will be merged. The new detection result will have for centroid, width and height the mean of centroids, means and heights of detections i and j. The number of fused detection will be used to determine a confidence index.

6.2 Part-based classification

In an indoor environment, partial occlusions are frequent. It is thus clearly insufficient to seek in the area of interest only forms similar to the human body in its whole. The upper part of the body (head and shoulders) is often the only visible part. In practice, four classifiers are used:

- the whole body,
- the upper-body (front/back view),
- the upper-body (left view),
- the upper-body (right view).

It is important to use several classifiers for the upper part of the body. Indeed, we empirically noticed that only one classifier for head and shoulders seen with all points of view was not sufficiently robust. In practice, we learn one left view classifier and use its symmetrical for the right view classifier. The size of the classifier of the whole body is 12×36, composed of 27 boosted classifiers and 4457 weak classifiers. This classifier was trained on the well-known INRIA person dataset [7] in which we reduced the size of the context around each person. The classifiers of the upper-body (front/back) are of size 20×20, composed of 23 stages and 1549 weak classifiers. Profile classifiers are of size 20×20, made of 22 stages and 1109 weak classifiers.

6.3 Confidence index

As seen previously, one detection result is a superposition of several detections. It is thus possible to build a confidence index $\mathcal{Y}_{i,t}$ for one classifier at time t for the object i, which depends of the number of detections $\varpi_{i,t}$:

$$\mathcal{Y}_{i,t} = \min(1, \frac{\varpi_{i,t}}{\nu}) \tag{13}$$

where ν is a constant. ν corresponds to the minimum number of detections so that the confidence index is maximum. For one person, there are four confidence indexes $\mathcal{Y}^1_{i,t}$, $\mathcal{Y}^2_{i,t}$, $\mathcal{Y}^3_{i,t}$ and $\mathcal{Y}^4_{i,t}$ corresponding to the detection result of each classifier. Since it is possible to track independently each object in the scene, it is possible to assign a label consistent in time to every tracked object. So, we build an confidence index $\mathcal{I}_{i,t}$ which depends on the confidence index of this same object i at the previous time and the confidence indexes of the four classifiers at the time t:

$$\mathcal{I}_{i,t} = \min(1, \mathcal{I}_{i,t-1} + \alpha_1(\mathcal{Y}^1_{i,t} - \mathcal{I}_{i,t-1}) \tag{14}$$

$$+ \alpha_2(\mathcal{Y}^{2'}_{i,t} - \mathcal{I}_{i,t-1}) - \alpha_3 \mathcal{I}_{i,t-1}) \tag{15}$$

where $\mathcal{Y}^{2'}_{i,t} = \max(\mathcal{Y}^2_{i,t}, \mathcal{Y}^3_{i,t}, \mathcal{Y}^4_{i,t})$. This confidence index varies between 0 and 1, 1 is its maximum. α_1, α_2 and α_3 are three thresholds making it possible to control the speed with which new information are taking into account. $\alpha_1 = 0$ is imposed if there is no detection of the whole-body detector because of a partial occlusion and $\alpha_3 = 0$ if there is at least one detection. These parameters values depend on a compromise between the false detections and the missed detections tolerance and depend also on the performance of each classifier in comparison with the others. Finally, a simple thresholding of this index enables to determine the nature of the tracked object.

7 Experimental results

We present in this section quantitative evaluation results. Our method is compared with:

- **method 1:** We firstly use a state of the art commercial person detector based on *Passive Infrared* (*PIR*) technology (*e.g.* [24]).
- **method 2:** The original Viola *et al.* [38] detection system used in a sliding window framework analyzing every image.
- **method 3:** This method is based on *method 2* in which the search space of the classifier is reduced with background subtraction, called *Viola [38] + BS*.

We present in the following: the dataset used for the evaluation and two different experiments. In the first one, we evaluate the binary output of the algorithm answering the question: *is there someone in the room?* Then we present a second experiment which considers also the number of detected persons and the precision of their localization.

7.1 The video dataset

In order to evaluate the system presented above, industrial partners involved in the CAPTHOM project have expressed their specific needs through a set of reference scenarios. We use three classes of scenarios from which we have built the evaluation dataset:

- **Set 1:** Scenarios involving a normal use of a room. In these scenarios, we need to detect humans that are static or moving, sitting or standing. We use 14 images sequences in 9 different places (offices, meeting rooms, corridors and dining rooms).
- **Set 2:** Scenarios of unusual activities (slow or fast falls, abnormal agitation). We have here 7 images sequences.
- **Set 3:** Finally, scenarios gathering all false detections stimuli (illumination variation, moving objects *etc*). We carried out 8 images sequences here.

In the following, **Set 4** is defined as the union of these 3 sets. In total, we use 29 images sequences in 10 different places. Images have a resolution of 320×240 and have an "average" quality. Each images sequence lasts from 2 to 10 minutes.

7.2 Human detection evaluation

In this section, we present the results obtained by the proposed method and three other ones previously described over various video datasets. Results are presented based on the maximal value of the *f-score* defined by:

$$\textit{f-score} = \frac{2.Precision.Recall}{Precision + Recall}. \tag{16}$$

where $Recall$ and $Precision$ are defined with:

$$\text{Precision} = \frac{\#\text{true positives}}{\#\text{true positives} + \#\text{false positives}} \tag{17}$$

$$\text{Recall} = \frac{\#\text{true positives}}{\#\text{true positives} + \#\text{false negatives}} \tag{18}$$

	PIR [24]	Viola [38]	Viola [38] + BS	Proposed method
Set 1	0.35	0.86	0.89	**0.98**
Set 2	0.27	0.64	0.85	**0.99**
Set 3	0.5	0.3	0.88	**0.92**
Set 4	0.50	0.67	0.83	**0.97**

Table 1 *f-score* corresponding to results obtained on various datasets.

Results presented in table 1 permits to make some remarks. First of all, it appears clearly here that the detector *PIR* does not present sufficient performances. Its principal drawback is that it can detect only temperature variations and so just moving persons. In many scenarios, there are so a lot of missed detections (false negative). Then, with the Viola *et al.* [38] detection system, even if results are appreciably better, performances observed do not allow an industrial application. By using also background subtraction with the Viola *et al.* [38] detection method, the number of false-positives has clearly fallen. Indeed, with this method, the background subtraction makes possible to reduce the search space of the classifier and thus there are logically less false detections. Finally, with the proposed method, which also uses tracking to have a history of displacements of persons, we are able "to smooth" in time detection results and thus to detect one person even if the classifier cannot recognize the person due for example to an unusual posture.

We showed here the advantages of using background subtraction and tracking. The performances of the proposed method are good since this method presents, for a detection rate of 97%, a false detection rate of approximately 3%.

7.3 Global evaluation

The evaluation presented here, has been designed to consider the evaluation of the number of detected persons and the precision of their localizations. This method is directly inspired from the work of Hemery *et al.* [20]. It is based on four stages, namely:

1. the calculation of the matching between ground truth and detection results,
2. the evaluation of localization,
3. the compensation of under and over detections,
4. the calculation of a global score.

So, the first stage is the calculation of the matching between ground truth and detection results. To do that, we build the matching matrix $\mathcal{H}$, defined by:

$$\mathcal{H} = \begin{pmatrix} \beta_{1,1} & \dots & \beta_{1,N} \\ \vdots & \ddots & \vdots \\ \beta_{M,1} & \dots & \beta_{M,N} \end{pmatrix} \tag{19}$$

where M corresponds to the number of objects in the ground truth, and N to the number of detected persons. Each element of this matrix $\beta_{i,j}$ represents the matching value between the object i of the ground truth A_i and the j person detected B_j defined by:

$$\beta_{i,j} = \frac{card(A_i \cap B_j)}{card(A_i \cup B_j)} \tag{20}$$

We consider here the bounding boxes of detection results. The matching matrix $\mathcal{H}$ for detection results of figure 7 is given below as illustration:

$$\mathcal{H} = \begin{pmatrix} 0 & 0.87 & 0 & 0 \\ 0 & 0 & 0.75 & 0 \\ 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0.6 \end{pmatrix} \quad (21)$$

Fig. 7 Detection result example.

The matching is then done thresholding each element of this matrix. By doing this, it is possible that one detection result corresponds to several objects in the ground truth. Indeed, since the algorithm that we propose here is largely based on the background subtraction, this case of figure is relatively frequent when two persons are close in the image. For the example of figure 7, we obtain the following matching matrix:

$$\mathcal{H} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (22)$$

Then, if $\beta_{i,j} = 1$, we evaluate the precision of the localization with the Martin criterion [28]. It has been shown in [21] that this metric is the most reliable one. The localization evaluation between the object i from the ground truth A_i and the detection j, called B_j, is defined by:

$$Mar(A_i, B_j) = 1 - \frac{\min\left(\frac{card(A_i \backslash B_j)}{card(A_i)}, \frac{card(B_j \backslash A_i)}{card(B_j)}\right)}{card(I)} \quad (23)$$

Finally, we compute the mean of the Martin values for all $\beta_{i,j} = 1$ by adding one 0 for each under-detection and each over-detection. In the above example, the value of the total score is:

$$S = \frac{0.87 + 0.75 + 0.6}{5} = 0.44. \quad (24)$$

We present in table 2 means of the global scores for each video dataset of scenarios presented previously.

	Viola [38]	Viola [38] + BS	Proposed method
Set 1	0.44	0.63	**0.67**
Set 2	0.47	0.53	**0.96**
Set 3	0.88	**0.93**	**0.93**
Set 4	0.51	0.7	**0.77**

Table 2 Global evaluation results

Performances obtained with the proposed method are once again higher than other methods. Our method is largely based on connected components obtained with the background subtraction. So, when two people are close in one image, only one connected component will represent them in the image. Thus, in the calculation of the global score, our method will be strongly penalized because there will be one under-detection and one bad localization considered, whereas the two people are detected but represented by only one connected component. In spite of that, performances of our method are better than the other ones. We present in figure 8 examples of results.

7.3.1 Hardware ressources

We present in the figure 9, the repartition of the *CPU* use between the various modules of the algorithm presented above. Of course, this distribution is variable and depends on the size of the area of interest analyzed by the classifier. If no area of interest is detected, the totality of the *CPU* will be used by the background subtraction and other operations (acquisition, initialization of variables *etc.*). Results presented in the figure 9 represent the mean of repartition computing over all videos of our dataset when there is at least one object detected, so various sizes and number of areas of interest are considered.

One can notice that the large majority of the computing time is used by the classifier. Indeed, we use finally 4 classifiers scanning the area of interest. However, this system is able to reach about 10 frames per second using a standard computer.

8 Conclusion

We have presented in this paper a real-time computer vision algorithm for human detection in video. This system is based on three distinct parts. We firstly use

Fig. 8 Examples of results.

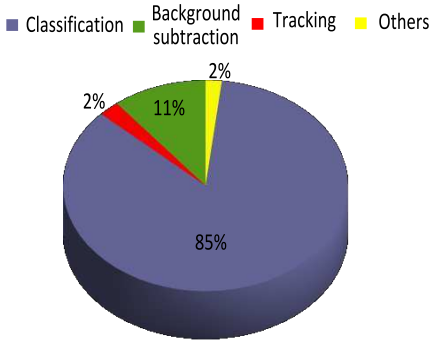

Fig. 9 Hardware ressources

background subtraction to limit the search space of the classifier. As each connected component detected potentially corresponds to one person, each blob is independently tracked. Finally, in order to know the nature of various objects that could be present in the scene, we use multiple part-based classifiers.

Results are very satisfactory since we reach a detection rate of 97% for approximately 3% of false and missed detections. These results are good since the scenarios on which the method was evaluated are representative of "real" scenes. Strengths of this method are due to the background subtraction which make possible to reduce the search space of the classifier and thus to reduce the number of false detections. The various levels of updates of the background model makes possible to manage most cases that could be observed in "real world" (variation of illumination, introduction or removal of static objects *etc*). Objects tracking in the image plan enables to build a confidence index on the recognition which evolves in time and thus allows to limit specific missed detections due to unusual postures.

While results are good enough to plan industrial applications, we are now working on the optimization of the algorithm in order to use it in an embedded hardware.

References

1. D.H. Ballard and C.M. Brown. *Computer Vision*. Prentice Hall Professional Technical Reference, 1982.
2. H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.
3. S. Belongie, J. Malik, and J. Puzicha. Matching shapes. *International Conference on Computer Vision*, pages 454 – 461, 2001.
4. Y. Benezeth, P.M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger. Review and evaluation of commonly implemented background subtraction algorithms. *International Conference on Pattern Recognition*, pages 1–4, 2008.
5. A.C. Bovik. *Handbook of Image and Video Processing*. Academic Press, Inc. Orlando, FL, USA, 2005.
6. R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, and O. Hasegawa. A system for video surveillance and monitoring. Technical report, Robotics Institute, Carnegie Mellon University, 2000.
7. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Conference on Computer Vision and Pattern Recognition*, 1:886–893, 2005.
8. N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. *European Conference on Computer Vision*, 2:428–441, 2006.
9. P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. *Computer Vision and Pattern Recognition*, pages 304–311, 2009.
10. A Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. *European Conference on Computer Vision*, pages 751–767, 2000.
11. M. Everingham, A. Zisserman, C. Williams, L. Van Gool, M. Allan, C. Bishop, O. Chapelle, N. Dalal, T. Deselaers, and G. Dorko. The 2005 pascal visual object classes challenge. *First PASCAL Challenge Workshop*, 2005.
12. P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. *Computer Vision and Pattern Recognition*, 2008.
13. D. Filliat. A visual bag of words method for interactive qualitative localization and mapping. *International Conference on Robotics and Automation*, pages 3921–3926, 2007.
14. W. Gao, H. Ai, and S. Lao. Adaptive contour features in oriented granular space for human detection and segmentation. *Computer Vision and Pattern Recognition*, 2009.
15. D. M. Gavrila. The visual analysis of human movement: a survey. *Computer Vision and Image Understanding*, 73:82–98, 1999.
16. D. M. Gavrila. A bayesian, exemplar-based approach to hierarchical shape matching. *Pattern Analysis and Machine Intelligence*, 29(8):1408–1421, 2007.
17. D.M. Gavrila, J. Giebel, and S. Munder. Vison-based pedestrian detection: The protector system. *Intelligent Vehicles Symposium*, pages 13–18, 2004.
18. C. Gu, J. J. Lim, P. Arbelez, and J. Malik. Recognition using regions. *Computer Vision and Pattern Recognition*, pages 1030–1037, 2009.
19. I. Haritaoglu, D. Harwood, and L.S. Davis. W 4: real-time surveillance of people and their activities. *Pattern Analysis and Machine Intelligence*, 22:809–830, 2000.

20. B. Hemery, H. Laurent, and C. Rosenberger. Evaluation metric for image understanding. *International Conference on Image Processing*, 2009.
21. B. Hemery, H. Laurent, C. Rosenberger, and B. Emile. Evaluation protocol for localization metrics application to a comparative study. *International Conference on Image and Signal Processing*, pages 273–280, 2008.
22. S. Johnsen and A. Tews. Real-time object tracking and classification using a static camera. *in people and detection workshop of the International Conference of Robotics and Automation*, 2009.
23. M. Jones and D. Snow. Pedestrian detection using boosted features over many frames. *International Conference on Pattern Recognition*, pages 1–4, 2008.
24. W. Kahl and R. Settanni. Us patent 4703171: Lighting control system with infrared occupancy detector, 1987.
25. D.G. Lowe. Object recognition from local scale-invarient features. *International Conference on Computer Vision*, 2:1150–1157, 1999.
26. D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.
27. B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.
28. D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *International Conference on Computer Vision*, 2:416–423, 2001.
29. N.M. Oliver, B. Rosario, and A.P. Pentland. A bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence*, 22:831–843, 2000.
30. M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. *Computer Vision and Pattern Recognition*, pages 193 –199, 1997.
31. C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38:15–33, 2000.
32. R.E. Schapire. The boosting approach to machine learning: An overview. *in Workshop on N.E.C.*, 2002.
33. B. Schiele, M. Andriluka, N. Majer, S. Roth, and C. Wojek. Visual people detection: Different models, comparison and discussion. *in people detection and tracking workshop of the International Conference on Robotics and Automation*, 2009.
34. C. Stauffer, W. Eric, and L. Grimson. Learning patterns of activity using real-time tracking. *Pattern Analysis and Machine Intelligence*, pages 747 – 757, 2000.
35. C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. *Computer Vision and Pattern Recognition*, 2, 1999.
36. O. Tuzel, F.M. Porikli, and P. Meer. Pedestrian detection via classification on riemannian manifolds. *Pattern Analysis and Machine Intelligence*, 30(10):1713–1727, 2008.
37. A. Utsumi and N. Tetsutani. Human detection using geometrical pixel value structures. *International Conference on Automatic Face and Gesture Recognition*, pages 34–39, 2002.
38. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition*, 1:511–518, 2001.
39. P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63:153–161, 2005.
40. C. Wojek, S. Walk, and B. Schiele. Multi-cue onboard pedestrian detection. *Computer Vision and Pattern Recognition*, pages 794–801, 2009.
41. C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *Pattern Analysis and Machine Intelligence*, 1997.
42. B. Wu and R. Nevatia. Tracking of multiple, partially occluded humans based on static body part detection. *Computer Vision and Pattern Recognition*, 1:951–958, 2006.
43. B. Wu and R. Nevatia. Cluster boosted tree classifier for multi-view, multi-pose object detection. *International Conference on Computer Vision*, 2007.
44. Q. Zhao, J. Kang, H. Tao, and W. Hua. Part-based human tracking in a multiple cues fusion framework. *International Conference on Pattern Recognition*, 1:450–455, 2006.

Yannick Benezeth is a PhD student at the University of Orléans, France. He reveived the engineer degree from the ENSI de Bourges and the MS degree from the University of Versailles-Saint-Quentin-en-Yvelines, France, in 2006. His research interests include computer vision and video analysis.

Bruno Emile is an assistant professor at IUT of Chteauroux (France). He obtained his Phd from the university of Nice in 1996. He belongs to the PRISME Institut of the University of Orléans in the Image and Signal for System research unit. His research interests concern object detection.

Hélène Laurent is an assistant professor at ENSI of Bourges, France. She obtained her Phd from the University of Nantes in 1998. She belongs to the PRISME Institut of the University of Orlans in the ISS (Images and Signals for Systems) research unit. Her research interests concern evaluation of image processing algorithms.

Christophe Rosenberger is a Full Professor at ENSICAEN, France. He obtained his Ph.D. degree from the University of Rennes I in 1999. He has been working at the GREYC Laboratory in the Computer Security Research Unit since 2007. His research interests include evaluation of image processing and biometrics.