Multimed Tools Appl (2017) 76:6801–6819
DOI 10.1007/s11042-016-3352-z

People counting via multiple views using a fast information fusion approach

Mikaël A. Mousse[1,2] · Cina Motamed[1] · Eugène C. Ezin[2]

Received: 24 July 2015 / Revised: 15 December 2015 / Accepted: 9 February 2016 /
Published online: 22 February 2016

Abstract Real-time estimates of a crowd size is a central task in civilian surveillance. In this paper we present a novel system counting people in a crowd scene with overlapping cameras. This system fuses all single view foreground information to localize each person present on the scene. The purpose of our fusion strategy is to use the foreground pixels of each single views to improve real-time objects association between each camera of the network. The foreground pixels are obtained by using an algorithm based on codebook. In this work, we aggregate the resulting silhouettes over cameras network, and compute a planar homography projection of each camera's visual hull into ground plane. The visual hull is obtained by finding the convex hull of the foreground pixels. After the projection into the ground plane, we fuse the obtained polygons by using the geometric properties of the scene and on the quality of each camera detection. We also suggest a region-based approach tracking strategy which keeps track of people movements and of their identities along time, also enabling tolerance to occasional misdetections. This tracking strategy is implemented on the result of the views fusion and allows to estimate the crowd size dependently on each frame. Assessment of experiments using public datasets proposed for the evaluation of counting people system demonstrates the performance of our fusion approach. These results prove that the fusion strategy can run in real-time and is efficient for making data association. We

✉ Mikaël A. Mousse
mousse@lisic.univ-littoral.fr

Cina Motamed
motamed@lisic.univ-littoral.fr

Eugène C. Ezin
eugene.ezin@imsp-uac.org

[1] EA 4491 - LISIC - Laboratoire d'Informatique Signal et Image de la Côte d'Opale, Université Littoral Côte d'Opale, 62228 Calais, France

[2] Unité de Recherche en Informatique et Sciences Appliquées, Institut de Mathématiques et de Sciences Physiques, Université d'Abomey-Calavi, Abomey-Calavi, Bénin

also prove that the combination of our fusion approach and the proposed tracking improve the people counting.

Keywords Visual surveillance · People counting · Homography · Data association · Tracking · Overlapping cameras

1 Introduction

Counting people in crowds is a crucial and challenging problem in video surveillance. There is a great interest for monitoring all types of environments. This could have many goals, e.g., security, resource management, urban planning, or advertising. A good estimation of the number of people in a crowd is a key indicator of the crowd security and safety and it can be extremely useful information. In a crowd, it is also very important to have a perfect view of every individual to detect potential anomalies. Counting people in the crowds is difficult because there are many occlusions. Several people counting approaches have been proposed in the past twenty years.

1.1 Related works

Most of the literature concerning people counting rely on a single view approach, due to the wide availability of a single surveillance cameras and to the relative ease of implementation [5, 6, 16, 34, 37, 42]. Even with strong prior assumptions and no computational limitations, often it is difficult to count efficiently people in a crowd from a single simple camera view. To overcome this drawback, several research works [1, 8, 12, 24] propose to use cameras which look straight down. These cameras are fixed to the ceiling. However, the application is mostly limited to indoor environments. Also stereo cameras have been considered, in order to exploit depth information to project moving people to the ground plane, producing an occupancy map and reducing occlusions [2, 15, 33, 36, 40]. Some research works proposed to use multi-camera system. The use of multiple cameras reveals as fundamental for localizing and counting people in crowded environments [13, 22, 26, 29, 30]. In computer vision community, the use of multi-camera takes a lot of scopes. Indeed, motivations are multiple and concern various fields as monitoring and surveillance of significant protected sites, control and estimation of flows (car parks, airports, ports, and motorways). Because of the fast growing of data processing, communications and instrumentation, such applications become possible. These kind of systems require more cameras to cover overall field-of-view. They reduce the effects of objects dynamic occlusion. Using several views of the same scene (multi-view) can allow to recover the information that could have been hidden in a specific view. The people counting systems have been subdivided into individual-centric methods, based on the detection, tracking, and counting the number of tracks, and crowd-centric methods, based on the analysis of global low-level features extracted from crowd imagery to produce accurate counts [4]. In this paper, we investigate individual-centric methods because these methods do not require a special learning about people from the scene to find their localizations and the counting is only based on people detection. Traditionally, counting involves first locating all the individual objects. It is because estimating the number of people depends on detecting individuals in order to count in crowded settings. However, locating all the objects is a demanding task because objects often look a like or occlude each other, making data association difficult.

According to Xu et al. [39], for multi-view object localization, existing multi-camera surveillance algorithms may be classified into three categories.

- The system in the first category fuses low-level information. In this category, multi camera surveillance systems detect and/or track in a single camera view. They switch to another camera when the systems predict that the current camera will not have a good view of the scene [3, 18]. These methods are vulnerable because the foreground information is extracted from individual camera views.
- In the second one, system extracts features and/or even tracks targets in each individual camera. After this, we integrate all features and tracks in order to obtain a global estimate.These systems are of intermediate-level information fusion [17, 21, 38]. These methods are still vulnerable because the features are extracted from individual camera views.
- The system in the third category fuses high-level information. In these systems, individual camera doesn't extract features but provide foreground bitmap information to the fusion center. Detection and/or tracking are performed by a fusion center [9, 19, 20, 39, 41].

This paper points out on the approaches in the third category because these algorithms are robust against dynamic objects occlusion. In this category some algorithms have been proposed. Khan and Shah proposed to use a planar homographic occupancy constraint to combine foreground likelihood images from different views [19]. It resolves occlusions and determines regions on the ground plane that are occupied by people. They also extended the ground plane to a set of planes parallel to it, but at some heights off the ground plane to reduce false positives and missing detections [20]. The foreground intensity bitmaps from each individual camera are warped to the reference image by Eshel and Moses [9]. The set of scene planes is at the height of people heads. The head tops are detected by applying intensity correlation to align frames from different cameras. This work is able to handle highly crowded scenes. Yang et al. detect objects by finding visual hulls of the binary foreground images from multiple cameras [41]. These methods use the visual cues from multiple cameras and are robust in coping with occlusion. However the pixel-wise homographic transformation at image level slows down the processing speed. To overcome this drawback, Xu et al. proposed an object detection approach via homography mapping of foreground polygons from multiple cameras [39]. They approximate the contour of each foreground region with a polygon and only transmit and project the vertices of the polygons. The foreground regions are detected by using Gaussian mixture model. These polygons are then rebuilt and fused in the reference image. They prove that their approach is 40 times faster than state of art fusion algorithm.

In this paper, the people counting system that we proposed suggest a new strategy based on reducing the complexity of polygons fusion for object localization in order to perform the data association between the foreground object detected in each single view. The foreground pixels of each camera view are detected by using codebook model. These pixels are grouped into polygon. For each person, our strategy is to detect the polygon which has the best representation of each person present on the scene. In each camera view, a foreground polygon is obtained by finding the convex hull of foreground region. The selection of the best polygon is done by incorporates geometric properties of the scene and the quality of each single view detection. We also introduce a multi-view tracking strategy to estimate the crowd size dependently on each frame. The number of people at each frame can be calculated by counting the number of polygons resulting from the fusion and the estimation

method associate to the tracking strategy gives the number of distinct people who spent at the scene.

The paper is organized as follows. Section 2 describes the proposed people counting approach. Experiments on different datasets and the performance evaluation are presented in Section 3. The conclusion and further works are presented in Section 4.

2 People counting approach

In this section, we present our proposed approach for people counting using multiple cameras. This method is divided into four modules :

- single foreground pixels detection : this module identifies the foreground pixels of each camera view. In this work we adopt a motion detection algorithm which is based on codebook model;
- foreground information fusion : this module merges all foreground pixels obtained using the first module to get a global information of the scene;
- tracking : this module is adopted to support people counting, by keeping track of people movements and of their identities along time;
- counting : the role of this module is to count people.

These modules are described below.

2.1 Motion detection using Codebook model

Foreground detection in each single scene view is the basic building block of our proposed people counting system and its accuracy is crucial for the entire process. Therefore, we adopt here the codebook background model for video sequences presented by Kim et al. [25], whose high accuracy and robustness to well known moving object detection challenges has already been proved. These results are confirmed in our past research work [32], in which we test the algorithm on several new sequences which presenting more challenging situations. This method detects in real-time object in dynamic background. Figure 1 represents the flow diagram of the codebook based algorithm.

In this method, each pixel p_t is represented by a codebook $\mathcal{C} = \{c_1, c_2, ..., c_L\}$ and each codeword c_i, $i = 1, ..., L$ by a RGB vector v_i and a 6-tuples $aux_i = \{\check{I}_i, \hat{I}_i, f_i, p_i, \lambda_i, q_i\}$ where $\check{I}$ and $\hat{I}$ are the minimum and maximum brightness of all pixels assigned to this codeword c_i, f_i is the frequency at which the codeword has occurred, λ_i is the maximum negative run length defined as the longest interval during the training period that the codeword has not recurred, p_i and q_i are the first and last access times, respectively, that the codeword has occurred. The codebook model is created or updated using two criteria. The first criterion is based on color distortion (1) whereas the second is based on brightness distortion (2).

$$\sqrt{||p_t||^2 - C_p^2} <= \varepsilon_1 \tag{1}$$

$$I_{low} \leq I \leq I_{hi} \tag{2}$$

In (1), the autocorrelation value C_p^2 is given by equation (3) and $||p_t||^2$ is given by equation (4).

$$C_p^2 = \frac{(R_i R + G_i G + B_i B)^2}{R_i^2 + G_i^2 + B_i^2} \tag{3}$$

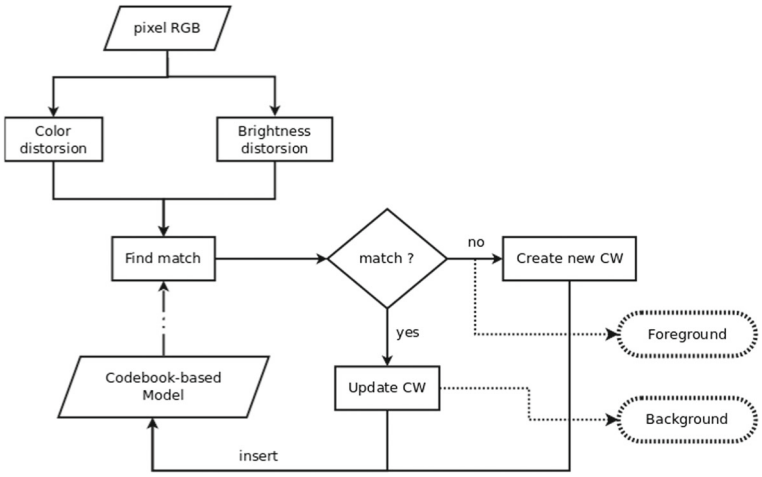

Fig. 1 Flow diagram of the Codebook based algorithm. (the *solid lines* correspond to the learning phase; *dashed lines* correspond to the phase of moving objects detection)

$$||p_t||^2 = R^2 + G^2 + B^2 \tag{4}$$

In relation (2), $I_{low} = \alpha \hat{I}_i$, $I_{hi} = min\left\{\beta \hat{I}, \frac{\check{I}}{\alpha}\right\}$ and $I = \sqrt{R^2 + G^2 + B^2}$.

After the training period, if an incoming pixel matches with a codeword in the codebook, then this codeword will be updated and this pixel will be treated as a background pixel. If the pixel doesn't match, its information will be put in cache word and this pixel will be treated as a foreground pixel.

2.2 Fusion strategy

In this section, we present our fusion approach for moving people counting in a multi camera system. Kuncheva et al. distinguish two information fusion approaches: decision fusion and source selection [28]. For them, decision fusion consists in combining the information from multiple sources to reach a consensus [7, 27] whereas source selection consists to choose dynamically the best source among the sources [14, 23]. In this work, our approach is based on source selection. The main idea of our fusion method is to find the camera which has the best view of the localization of each person present on the scene.

After the foreground pixels in each view are detected, these pixels need to be grouped into foreground regions. Each region can be approximated by a polygon. The polygon is obtained by finding the convex hull of all contours detected in threshold image. The convex hull or convex envelope of a set X of points in the Euclidean plane or Euclidean space is the smallest convex set that contains X. For instance, when X is a bounded subset of the plane, the convex hull may be visualized as the shape enclosed by a rubber band stretched around X. To find the convex hull, we search for any point in X that enters the minimal convex hull for sure. We choose the one with the least x-coordinate (the left most one in X). We create a P list in which we store the numbers of the points (their position in X array). After that, we sort a set of points in increasing order (except for P[0]) as for their left position with regard to the starting $R = XP[0]$ point. We consider that $B < C$ if C point is on the left from RB vector. Then we apply a sort algorithm based on pair-wise comparison

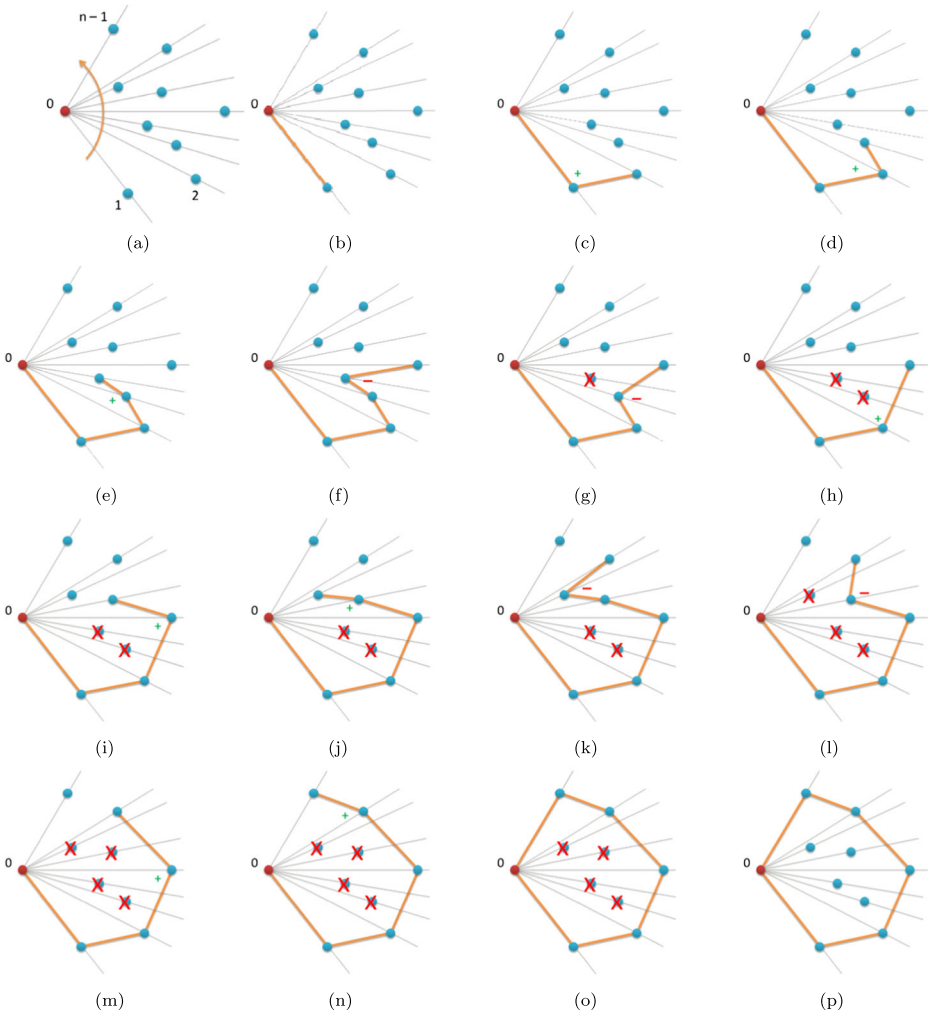

Fig. 2 Example of detection of the convex hull of a set of points

of these elements. This allows us to have a starting point for our polygon and a potential order in the succession of the vertices of the polygon (confers sub-figure (a) of Fig. 2 : the starting point is colored red and the potential order is 0, 1, 2, ..., $n - 1$). The last step is to cut angles. In order to do that, we create a list S and place the first two vertices into it ($S = [P[0], P[1]]$). Then look through all other vertices, we keep track of recent three points, and we find the angle formed by them. If orientation of these points is not counter-clockwise, we can cut the angle by removing the last vertex from S. As soon as orientation is clockwise, it is no longer necessary to cut angles, so we will place the current vertex into S. This last step is illustrated by the sub-figures (b), (c),.., (p) of Fig. 2. All region can be approximated by a polygon and each polygon is convex. Figure 3 presents some results of the implementation of the moving objects detection algorithm and the approximation by a polygon strategy. The approximation by a polygon reduces the amount of data which will

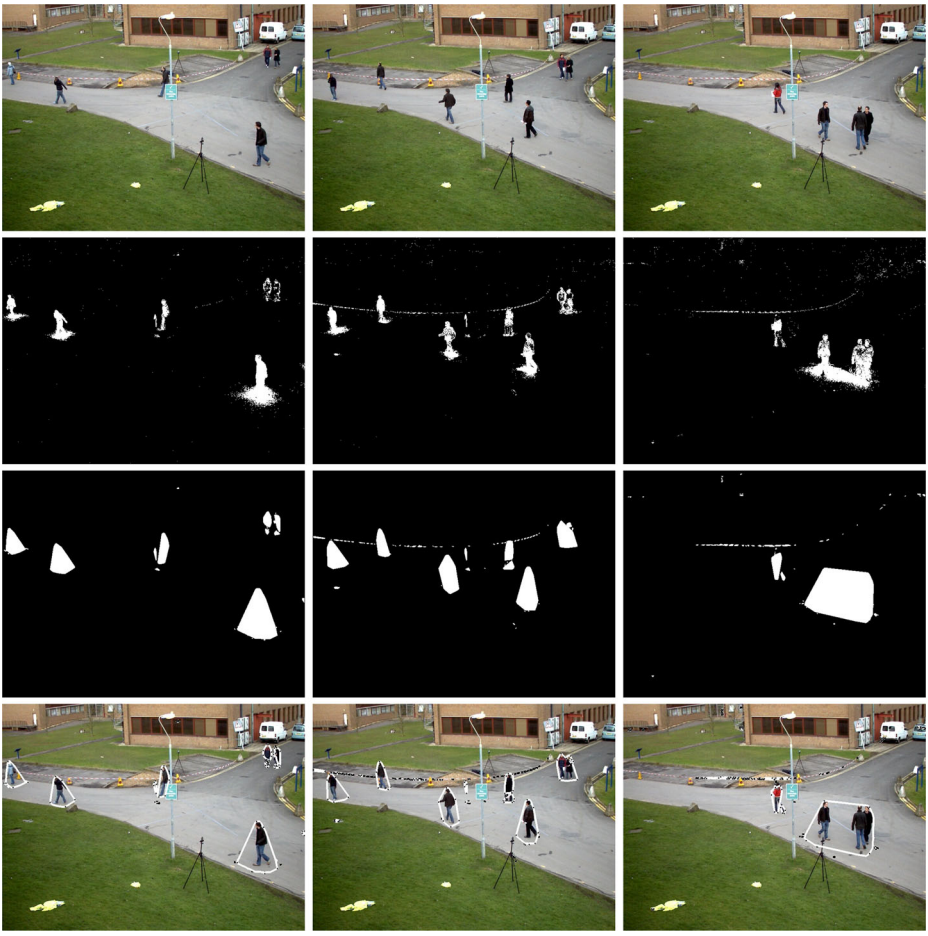

Fig. 3 These results are obtained by using sequence from PETS 2010 dataset. The first row shows the original images. The second row shows the foreground pixels obtained by using Codebook based algorithm. The third row shows the foreground regions and the last row shows the detected polygons. *Black points* on polygon represent the vertices of the polygon

be processed. This approximation also allows us to consider all the holes or discontinuous blobs as foreground pixels. Because all pixels which belong to the polygon are considered to be a foreground pixels. To use the polygon vertices in the information fusion module, we have decided to assign an unique identifier id to each polygon. With the detection of each foreground regions, we need to fuse these regions to get a multi-view information. We find the projection of each polygon in the ground plane by considering the projection of each of its vertices in this plan. The projection is done by using planar homography mapping.

Homographies are usually estimated between a pair of images by finding feature correspondence in these images. The most commonly used feature is corresponded points in different images, though other features such as lines or conics in the individual images may be used. These features are selected and matched manually or automatically from 2D images to compute the homography between two camera views or the homography between one camera view and the top view. Thus, a calibration of the stage must be carried out for

obtaining the projection matrix. The homography transformation is a special variation of the projective transformation. Let us consider the point $x = (x_s, y_s, 1)$ in the image without distortion and the point $X = (X_w, Y_w, Z_w, 1)$ in the 3D world. The projection transformed from X to x is given by equation (5).

$$\begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \begin{bmatrix} f/s_x & s & C_x & 0 \\ 0 & f/s_y & C_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 0 \\ 1 \end{bmatrix} \tag{5}$$

If X is limited on the ground plane, therefore Z_w will be 0 and the projection transformed from X to x becomes:

$$\begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \begin{bmatrix} f/s_x & s & C_x \\ 0 & f/s_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} \tag{7}$$

Planar homography mapping consists in finding 3×3 matrix which correspond a point $pt(x, y)$ to an other point $pt'(x', y')$ on the ground plane in two different views. In case of non-planar scenes, a spatial model would be needed and has to be integrated. This spatial model should include the detection of the planar sub-regions in the scene that do not conform to the planar hypothesis. A homography can be performed for each subregion because these subregions are planar. The images are then combined by considering the spatial model (Fig. 4).

After projection, we propose an strategy to fuse the polygons. Our fusion approach is based on geometric properties of the scene and on the quality of each camera detection. Let us consider a scene being observed by cameras with overlapping views as shown in Fig. 5. In this figure, the scene is observed by two cameras. Each camera observes the scene differently and it is then necessary to make a mapping of the scene to know the fields of view of each camera. This information is important in order to do an efficient fusion because it identifies the number of camera which covers each point of the scene. Using the projected polygons obtained from each single single view and considering two views, we identify three cases:

c1 : A polygon from the first view is not associated with any polygon from the second view. If the polygon is detected in an area cover by one camera we assume that this camera has the best possible view of the object. Else If the polygon is detected in an area cover by the two cameras, we assume that this polygon is an false detection. Then we ignore this projected polygon. This case allows us to reduce the counting errors due of the presence of the false positive pixels in each thresholded view.

c2 : A polygon from the first view is associated with only one polygon from the second view. For the selection of the best view, we prioritize the camera which detects the lowest point of the projected polygons (projection in the ground plane) associated with the person. For each polygon the lowest point is the point nearest to the ground. It is the vertex which has the largest value on y-axis. Using the example presented in Fig. 6, our fusion strategy will select the polygon which comes from the second view (polygon in color red) because it detects the lowest point of the polygon which is obtained from the second view. If this criterion does not permit to choose a polygon

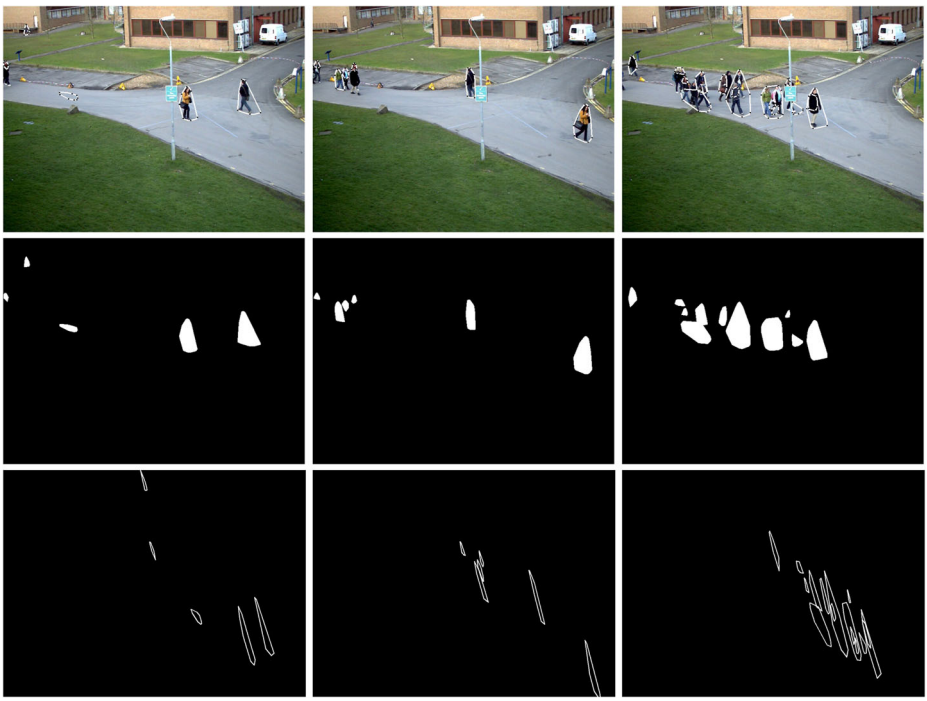

Fig. 4 These results are obtained by using sequence from PETS 2010 dataset. The first row shows the original images with detected polygons. The second row shows the foreground mask and the third row presents the projected polygons. The *ground plane* image is obtained using Google Maps view of the scene provided with the dataset

as in the case of Fig. 7, the selection will be made with respect to the position of the object relative to the camera. Indeed, this position has an influence on the rendering in the homographic plan. For performing this criterion, for each camera we calculate

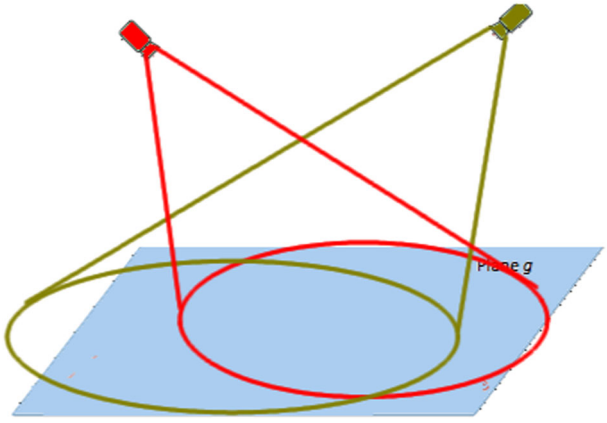

Fig. 5 Illustration of scene observed by two cameras

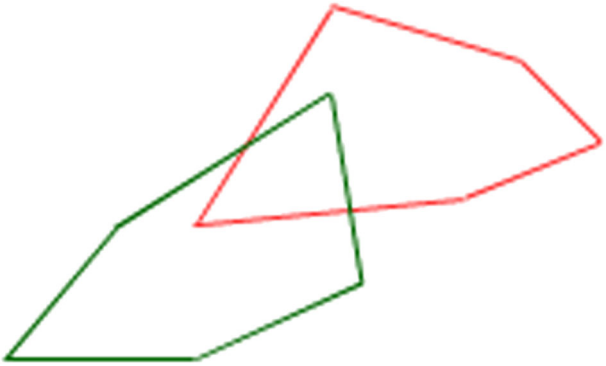

Fig. 6 Polygons obtained after projection in a ground plane. In this plane, the x-axis is oriented from right to left and the y-axis is oriented from bottom to up. The *polygon in color green* is from the first view and the polygon in color red is from the second view

the distance between the highest vertex of the projected polygon (vertex which has the smallest value on y-axis) and the projection of the lowest vertex of the same polygon. The best polygon is the polygon which has the smallest distance. In the case of Fig. 7, the best polygon is the polygon that has the color green.

c3 : A polygon from the first view is associated with more than one polygon from the second view. This third case is illustrated by Fig. 8. In this case, we conclude that this is a dynamic occlusion between objects. According to this, the best camera is the camera in which has the largest number of associated polygon. These polygons are then selected. If we apply this principle to Fig. 8, the green polygons will be selected as result of the fusion process.

In **c1**, **c2** and **c3**, a polygon $P1$ from a view will be considered associated with a polygon $P2$ which is from another view, if the projection into the ground plane of one of the vertex of $P1$ belongs to the projected polygon obtained by projecting the vertices of $P2$ into the ground plane. The ray casting algorithm proposed by Sutherland et al. in [35] has been used in order to resolve point-in-polygon problem. In its, the number of times that a ray starting

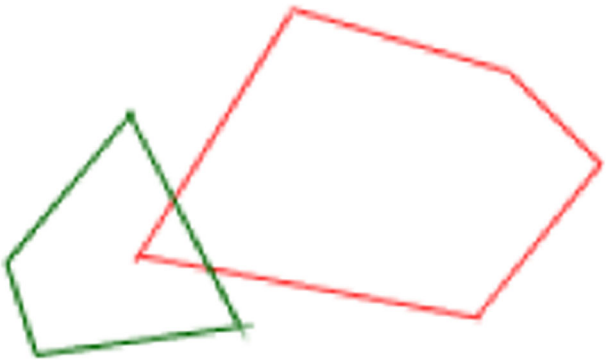

Fig. 7 Polygons obtained after projection in a ground plane. In this plane, the x-axis is oriented from right to left and the y-axis is oriented from bottom to up. The *polygon in color green* is from the first view and the polygon in color red is from the second view

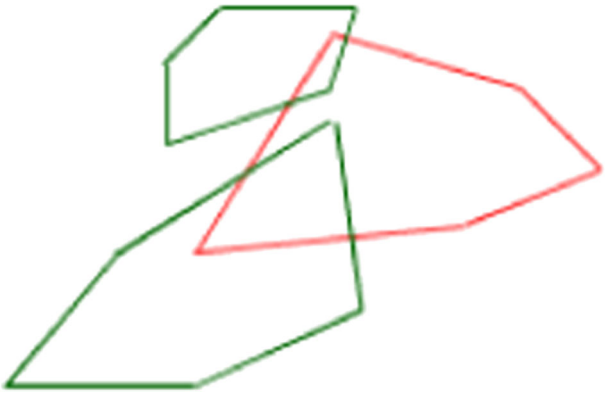

Fig. 8 Polygons obtained after projection in a ground plane. In this plane, the x-axis is oriented from right to left and the y-axis is oriented from bottom to up. The *polygon in color green* is from the first view and the polygon in color red is from the second view

from the given point intersects the edges of the polygon is counted. If the point in question is not on the boundary of the polygon, it is outside if the number of intersections is an even number; it is inside if this number is odd. This algorithm can be applied to either convex or concave polygons.

For each of the remaining cameras (if any remains), the fusion process described above is repeated by considering the polygons resulting from the previous fusion and those of the new view.

2.3 Tracking strategy

Our tracking strategy and people counting method are presented in this section. In the proposed system the tracking module is adopted to support people counting, by keeping track of people movements and of their identities along time. Our tracking module is an extension of tracking strategy proposed by Motamed [31]. Indeed, he proposed a simple and effective strategy for tracking objects in a mono-camera video surveillance system. Our tracking strategy incorporates in algorithm proposed by [31], the requirements of a multi camera system. The proposed strategy track the object on the ground plane after the fusion. Then, after fusion and in order to take into account approximate object behavior first order position prediction in a ground plane of each tracked object is used. For each detected region, some visual features are computed: centre of gravity, bounding box, and color histograms of the object in each view. These histograms are reduced to 32 bins per color channel in order to reduce the information quantity. For each object in order to reduce region candidates, a spatial validation gate is defined. The gate permits to incorporate cinematic limitation for all objects. For computational efficiency, the dimension of this gate is defined empirically to twice of the object bounding box and is located around the predicted object position. The matching is based on the spatial proximity of regions and their visual compatibilities. The algorithm evaluates explicitly the quality of each association. This information is summarized by two indicators: Consistency and Identity indicators. These indicators are recursively updated and stored during the tracking step. The consistency indicator of tracked objects firstly permits effective new objects to be validated after consecutive observations. Secondly, it permits to tolerate some temporary loss of the objects having a reliable track. It

Table 1 Object information

Object information
Object number
View 1 color histogram (k), view 2 color histogram (k),...,view n color histogram (k)
Position in ground plane (k), Bounding box in ground plane (k)
Consistency indicator (k)
Speed vector (k)
Identity indicator (k)

reacts as a robust filter at the object level. The indicator increases when no significant variation in the object features (color histogram of associated object in each view, and bounding box size of the resulting projected polygon on the ground plane and the speed vector of the projected polygon) is perceived. Otherwise or in extreme situations when the target is lost, indicator is decreased. The dissimilarity between the object color histograms is performed by the Bhattacharya distance. The updating process of the consistency indicator of each tracked object is controlled in terms of time delay defined by the human expert as the stability indicator. The track termination is decided for lost objects after a period of consecutive zero value consistency indicator. This delay has been fixed typically at 3 s. For each tracked object, a set of information is stored (Table 1). In Table 1 the attribute "Object number" is a unique identifier for each object. We adopt the merging procedure (or respectively splitting procedure) presented in [31] when we are in merging situation (or respectively splitting situation). The splitting situation is detected once a new object is detected close to a temporary group region whereas the merging situation is detected whereas the merging situation is detected one object is detected close to more than one object.

2.4 Counting strategy

Our method of people counting is closely associated with the tracking strategy. When an effective new object is detected by the tracker the number of the people is is incremented by one. The tracker detects a new object when it creates an object with new object number. Proceeding as this, and based on the assumption that all objects are tracked with the tracker, we are sure to count all the people who passed through the scene. For counting the people on the frame we firstly count all objects detect by the tracker. We add to this, the number of items which are considered temporarily lost by according to their consistency indicators. This strategy allows us to perform an efficient counting while tolerating some temporary loss.

3 Experimental results and performance evaluation

3.1 Experimental environment

In this section we present the experimental environment in order to evaluate the performance of the proposed multi-view people counting method based on our fusion algorithm. This method has been tested on several multi-view sequences. These sequences are publicly available and adopted by other existing methods. These sequences belong to EPFL [11] and

PETS 2010 [10] datasets. They were adopted in the experiments of many research works. Consequently, it is fair to compare the performance between our proposed algorithm and some past research works. Sequence **Terrace1** ($view_0$ through $view_3$) from the EPFL dataset is a sequence of about 3 and 1/2 min, where up to 7 people walk around a terrace. These sequences present well known issues for people detection, tracking, and counting, including lighting variations and shadows, distance of the cameras from the scene, and frequent occlusions due to crowd. Sequences **Time_12-34_S0** ($view_1$ through $view_8$), **Time_12-34_S2** ($view_1$ through $view_8$), **Time_13-57_S1** ($view_1$ through $view_8$), and **Time_13-59_S1** ($view_1$ through $view_4$) from the PETS 2009 dataset are short sequences where up to 40 individuals walk around in a $10m^2$ area. For the tracking module, we consider that the consistency indicator reaches its maximum value after a delay of 1.5 s of good associations. All experiments are performed by using a laptop which has an Intel Core i7 CPU L 640 @ $2.13GHz \times 4$ processor with 4GB memory and the programming language is C++ through the OpenCv Library.

3.2 Performance evaluation and discussion

We present and analyze the performance of our proposed method at two levels. We evaluate first the performance of our fusion strategy which is a crucial module of our people counting method. After this, we analyze the performance of the counting system.

The aim of the fusion module is to perform the association between people from each camera of the network. We compare our fusion method to other fusion approach [39, 41] proposed in the state of the art. The results show that our approach provides a good moving people association and provides good accuracy in a dynamic occlusion case. It has similar performance to the fusion approach proposed in the state of the art [39, 41]. We also evaluate the processing time of the fusion approach and compare it to fusion approach suggested in [41] and [39]. The results for **Time_13-59_S1** and **Time_12-34_S2** datasets are reported in Fig. 9. This figure confirm the conclusion of works done in [39] (Authors in [39] say that their approach is 40 times faster than state of art fusion algorithm). These results also confirm that our proposed fusion approach run faster than Xu et al.'s method. Thus we can conclude that our fusion algorithm allows to make faster data association between the different observations of each single camera. Also for measure the gained brought of the

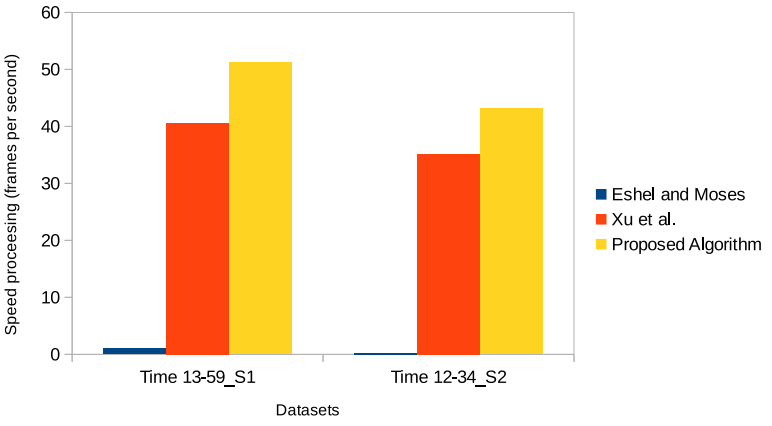

Fig. 9 Speed processing evaluation for **Time_13-59_S1** and **Time_12-34_S2** datasets

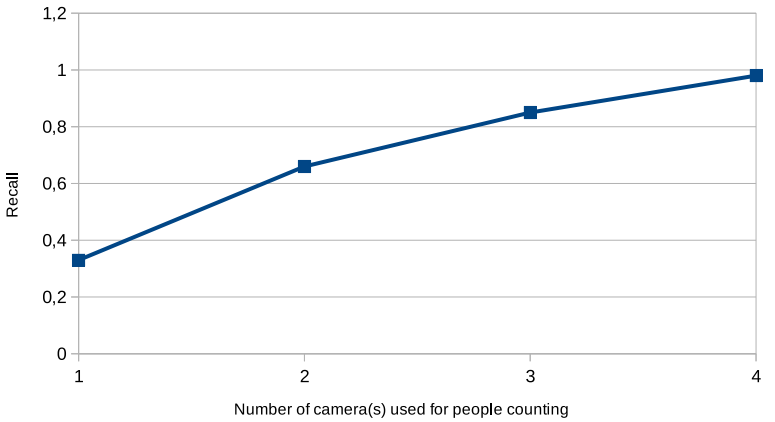

Fig. 10 Evolution of the recall value depending on the number of camera used for people counting (sequence : Terrace 1)

fusion strategy, we study the impact of the number of camera which is used in the counting process. The evaluation of the people counting is performed by using some metrics. These metrics are Recall, Precision and F-Measure. For sequence **Terrace 1**, the evolution of the recall and precision values depending on the number of camera is shown by Figs. 10 and 11. These two figures, demonstrate that the recall and precision values increase with the number of cameras. The fusion strategy therefore combines effectively the information of different views to increase the counting accuracy.

After proving the usefulness of the fusion method, we compare the performance of our people counting approach to other algorithms of the state of the art. Where available, we compare the performance values achieved by the multi-view people counting methods reported in [13, 29] and [30]. The results are reported in Table 2. These results showing the higher accuracy of the proposed approach. Our method outperform than methods proposed by Ge and Collins [13] and Ma et al. [29]. Its performance is closer than method proposed

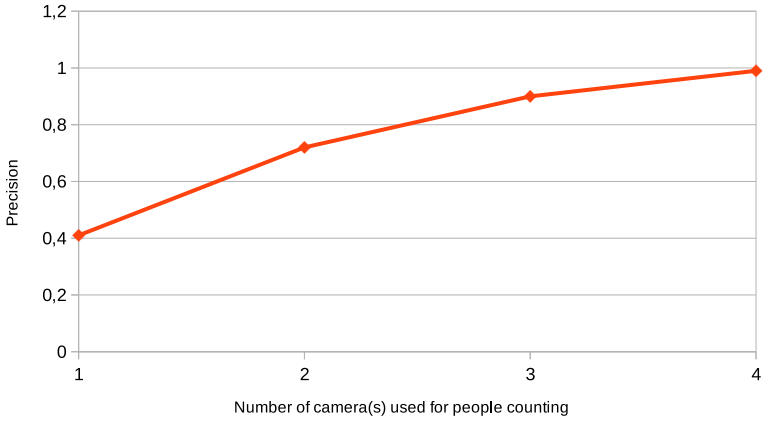

Fig. 11 Evolution of the precision value depending on the number of camera used for people counting (sequence : Terrace 1)

Table 2 People counting performance in sequences **Time_12-34_S0**, **Time_12-34_S2**, **Terrace1**

Sequence	Algorithm	Recall	Precision	F-measure
Time_12-34_S0	Maddalena et al. [30]	0.97	0.99	0.98
	proposed	0.98	0.99	0.98
Time_12-34_S2	Ge and Collins [13]	0.91	0.95	0.93
	Ma et al. [29]	0.92	0.97	0.94
	Maddalena et al. [30]	0.98	0.98	0.98
	proposed	0.98	0.98	0.98
Terrace 1	Ma et al. [29]	0.95	0.92	0.93
	Maddalena et al. [30]	0.96	0.96	0.96
	proposed	0.98	0.99	0.98

by Maddalena et al. [30]. But our system doesn't need a learning phase whereas the system proposed in [30] use an supervised classification approach. This provides much greater flexibility to our system. We also present the results for sequences **Time_13-57_S1** and **Time_13-59_S1** in terms of Average Frame Error. This results are reported in Fig. 12. By observing the Fig. 12, we conclude that our proposed method provides a low average error when we compare it to that proposed by Maddalena et al. [30]. Thus our method gives better result for counting while minimizing errors. Summing up performance results reported in Table 2 and in Fig. 12, we can conclude that the proposed approach achieved good performance in the case of moderate crowd density scenes. Finally, we report on Table 3 the processing rate of our algorithm using a common laptop described in Section 3.1. Using this table we conclude that our method which is implemented on a common laptop, achieves its performance at a speed that varies according to the number of people present on the scene and the number of cameras which is used.

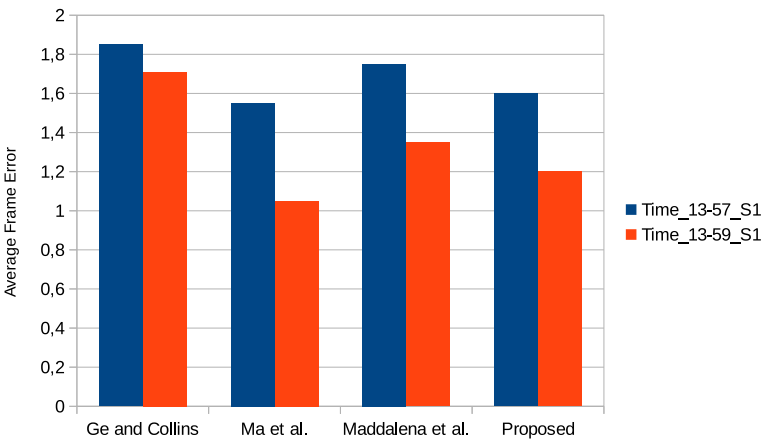

Fig. 12 Average Frame Error for people counting in sequences **Time_13-57_S1** and **Time_13-59_S1**

Table 3 Processing rate

Sequence	Processing rate
Time_12-34_S0	2.02 frames/second
Time_12-34_S2	2.11 frames/second
Time_13-59_S1	5.32 frames/second
Terrace 1	7.97 frames/second

4 Conclusion and future works

In this paper, we propose a multi-view camera algorithm for people counting in video surveillance. The system relies on achievements in multi-view video objects detection and moving object tracking, integrating modules that turned out to be very efficient. We use a codebook based model to detect foreground pixels and we fuse the foreground maps into ground plane. The computational complexity of our data association method is a major advantage for our counting system. Subsequent tracking system is adopted in order to estimate the crowd size dependently on each frame. This system is robust against dynamic occlusion (through the use of the fusion strategy) and the temporary loss of detection of objects (through the use of the tracking strategy). However, as the crowd becomes more larger and denser, individual detection and tracking become hard, and thus people counting tends to be less accurate. An alternative could be a "crowd-centric" approach, based on analyzing global low-level features extracted from crowd imagery to produce accurate crowd counting estimation. The counting method is also highly dependent on each camera foreground pixel extraction. Then the presence of false positive can influence the counting results. Finally counting errors will occur if non-human objects appear in the scene.

In a future, we plan to exploit motion direction trajectories to segment the crowds into sub-parts moving in different directions. This information will be useful in order to propose an algorithm for event recognition based on motion trajectory analysis.

Acknowledgments This work is partially funded by the Association AS2V and Fondation Jacques De Rette, France. Authors are grateful to the Service de Coopération et d'Action Culturelle de l'Ambassade de France au Bénin. We also appreciate the valuable comments provided by the anonymous reviewers as these have improved the manuscript immensely.

References

1. Albiol A, Mora I, Naranjo V (2001) Real-time high density people counter using morphological tools. IEEE Trans Intell Trans Syst 2(4):204–218
2. Beymer D (2000) Person counting using stereo. In: Proceedings of workshop on human motion, pp 127–133
3. Cai Q, Aggarwal J (1998) Automatic tracking of human motion in indoor scenes across multiple synchronized video streams. In: Proceedings of IEEE international conference on computer vision, pp 356–362
4. Chan A, Vasconcelos N (2012) Counting people with low-level features and bayesian regression. IEEE Trans Image Process 21(4):2160–2177
5. Conte D, Foggia P, Percannella G, Tufano F, Vento M (2010) A method for counting moving people in video surveillance videos. EURASIP J Adv Signal Processing:1–10

6. Davies A, Yin JH, Velastin S (1995) Crowd monitoring using image processing. Electron Commun Eng J 7(1):37–47
7. Duin RPW, Tax DMJ (2000) Experiments with classifier combining rules. In: Proceedings of 1st international workshop on multiple classifier systems, pp 16–29
8. Englebienne G, Krose B (2010) Fast bayesian people detection. In: Proceedings of the 22nd benelux conference on artificial intelligence
9. Eshel R, Moses Y (2008) Homography based multiple camera detection and tracking of people in a dense crowd. In: Proceedings of 18th IEEE international conference on computer vision and pattern recognition, pp 1–8
10. Ferryman J, Shahrokni A (2009) An overview of the pets 2009 challenge. In: Proceedings of the 11th IEEE international workshop on performance evaluation of tracking and surveillance, pp 25–30
11. Fleuret F, Berclaz J, Lengagne R (2008) Multi-camera people tracking with a probabilistic occupancy map. IEEE Trans Pattern Anal Mach Intell 30(2):267–282
12. Fu H, Ma H, Xiao H (2014) Scene-adaptive accurate and fast vertical crowd counting via joint using depth and color. Multimed Tools Appl 73(1):273–289
13. Ge W, Collins R (2010) Crowd detection with a multiview sampler. In: Daniilidis K, Maragos P, Paragios N (eds) Computer vision ECCV 2010. Lecture Notes in Computer Science, pp 324–337
14. Giacinto G, Roli F, Fumera G (2000) Selection of classifiers based on multiple classifier behaviour. In: Advances in pattern recognition, pp 87–93
15. Harville M (2004) Stereo person tracking with adaptive plan-view templates of height and occupancy statistics. Image Vis Comput 22(2):127–142
16. Hashemzadeh M, Pan G, Yao M (2014) Counting moving people in crowds using motion statistics of feature-points. Multimed Tools Appl 72(1):453–487
17. Hu W, Hu M, Zhou X, Tan T, Lou J, Maybank S (2006) Principal axis-based correspondence between multiple cameras for people tracking. IEEE Trans Pattern Anal Mach Intell 28(4):663–671
18. Khan S, Shah M (2003) Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. IEEE Trans Pattern Anal Mach Intell 25(10):1355–1360
19. Khan SM, Shah M (2006) A multi-view approach to tracking people in crowded scenes using a planar homography constraint. In: Proceedings of 9th European conference on computer vision, pp 133–146
20. Khan SM, Shah M (2009) Tracking multiple occluding people by localizing on multiple scene planes. IEEE Trans Pattern Anal Mach Intell 31(3):505–519
21. Kang J, Cohen I, Medioni G (2003) Continuous tracking within and across camera streams. In: Proceedings of international conference on computer vision pattern recognition, vol 1, pp 267–272
22. Kim K, Davis LS (2006) Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. In: Proceedings of the 9th European conference on computer vision, pp 98–109
23. Kim J, Seo K, Chung K (1997) A systematic approach to classifier selection on combining multiple classifiers for handwritten digit recognition. In: Proceedings of international conference on document analysis and recognition, vol 2, pp 459–462
24. Kim JW, Choi KS, Park WS, Lee JY, Ko SJ (2002) Robust real-time people tracking system for security
25. Kim K, Chalidabhonse TH, Harwood D, Davis L (2005) Real-time foreground-background segmentation using codebook model. Elsevier Real-Time Imaging 11(3):167–256
26. Krahnstoever N, Yu T, Patwardhan K, Gao D (2009) Multi-camera person tracking in crowded environments. In: Proceedings of 12th IEEE international workshop on performance evaluation of tracking and surveillance, pp 1–7
27. Kuncheva LI, Whitaker CJ, Ship CA, Duin RPW (2000) Is independence good for combining classifiers? In: Proceedings of international conference on pattern recognition, vol 2, pp 168–171
28. Kuncheva LI, Bezdek CJ, Duin RPW (2001) Decision templates for multiple classifier fusion : on experimental comparison. Pattern Recogn:299–314
29. Ma H, Zeng C, Ling CX (2012) A reliable people counting system via multiple cameras. ACM Trans Intell Syst Technol 3(2)
30. Maddalena L, Petrosino A, Russi F (2014) People counting by learning their apperarance in a multi-view camera environment. Pattern Recogn Lett 36:125–134
31. Motamed C (2006) Motion detection and tracking using belief indicators for an automatic visual-surveillance system. Image Vis Comput:1192–1201

32. Mousse MA, Ezin EC, Motamed C (2014) Foreground-background segmentation based on codebook and edge detector. In: 10th international conference on signal-image technology and internet-based systems, pp 119–124
33. Qiuyu Z, Li T, Yiping J, Wei Jun D (2010) A novel approach of counting people based on stereovision and dsp. In: Proceedings of The 2nd international conference on computer and automation engineering, vol 1, pp 81–84
34. Subburaman V, Descamps A, Carincotte C (2012) Counting people in the crowd using a generic head detector. In: Proceedings of IEEE 9th international conference on advanced video and signal-based surveillance, pp 470–475
35. Sutherland IE, Sproull RF, Schumacker RA (1974) A characterization of ten hidden surface algorithms. In: ACM Computing Surveys (CSUR), pp 1–55
36. van Oosterhout T, Bakkes S, Krse BJA (2011) Head detection in stereo data for people counting and segmentation. In: VISAPP, pp 620–625
37. Wren C, Azarbayejani A, Darrel T, Pentland A (1996) Pfinder: real-time tracking of the human body. In: Proceedings of 2nd international conference on automatic face and gesture recognition, pp 51–56
38. Xu M, Orwell J, Lowey L, Thirde D (2005) Architecture and algorithms for tracking football players with multiple cameras. IEE Proc-Vis Image Signal Process 152(2):232–241
39. Xu M, Ren J, Chen D, Smith J, Wang G (2011) Real-time detection via homography mapping of foreground polygons from multiple. In: Proceedings of 18th IEEE international conference on image processing, pp 3593–3596
40. Yahiaoui T, Khoudour L, Meurie C (2010) Real-time passenger counting in buses using dense stereovision. J Electron Imaging 19(3)
41. Yang DB, Gonzalez-Banos HH, Guibas LJ (2003) Counting people in crowds with a real-time network of simple image sensors. In: Proceedings of 9th IEEE international conference on computer vision, vol 1, pp 122–129
42. Zhao T, Nevatia R (2003) Counting people in crowds with a real-time network of simple image sensors. In: Proceedings of IEEE international conference on computer vision and pattern recognition, vol 1, pp 122–129

Mikaël A. Mousse received a Bachelor Engineering degree from Ecole Nationale d'Economie Appliquée et de Management of Université d'Abomey-Calavi, Bénin in 2008 and Master degree in Computer Engineering and Applied Sciences in 2012 from Institut de Mathématiques et de Sciences Physiques of Université d'Abomey-Calavi, Bénin. He is currently completing his Ph.D degree in Computer Engineering at Institut de Mathématiques et de Sciences Physiques of Université d'Abomey-Calavi, Bénin and at Université du Littoral Côte d'Opale, France. His research interests include signal processing, image processing, video processing, machine learning and pattern analysis and recognition.

Cina Motamed is associate professor in Computer Science in the University of Littoral Cote d'Opale, Calais, France. He received his B.Sc. in mathematics, and M.Sc in Electrical Engineering and Computer Science from the University of Caen, France and the PhD degree in Computer Science from the University of Compiegne, France, in 1987, 1989, and 1992, respectively. Current research is concerned with the automatic visual surveillance of wide area scenes using computational vision. His research interests focus on the design of multicamera system for real-time multiobject tracking and human action recognition. He is recently focusing on the uncertainty management over the vision system by using graphical models, and beliefs propagation. He is also interested by unsupervised learning approaches for human activity recognition.

Eugène C. Ezin, IEEE member in computer society, received his PhD degree in 2001 with highest level of distinction after research works carried our on neural networks and fuzzy systems for speech applications at the International Institute for Advanced Scientific Studies in Italy. Since July 2012, he is associate professor in computer science in the field of artificial intelligence. He supervised many master theses and some works are ongoing for PhD theses. He is a reviewer for Mexican International Conference on Artificial Intelligence and other journals. His research interests include machine learning, neural networks and fuzzy systems, signal and image processing, cryptography, information system and network security. He is also interested by human activities recognition through multi sensor systems.

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH ("Springer Nature").

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users ("Users"), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use ("Terms"). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com