Graph-based people segmentation using a genetically optimized combination of classifiers

Cyril Meurie, Olivier Lézoray, Christophe Coniglio, Marion Berbineau

Graph-based people segmentation using a genetically optimized combination of classifiers

Cyril Meurie,[a,*]**, Olivier Lézoray**[b]**, Christophe Coniglio**[a]**, Marion Berbineau**[a]
[a]Univ Lille Nord de France, F-59000 Lille, IFSTTAR, COSYS, LEOST, F-59650, Villeneuve d'Ascq
[b]Normandie Univ., UNICAEN, ENSICAEN, GREYC UMR CNRS 6072, Caen

Abstract. Many approaches for background subtraction and people detection have been developed so far. However, the best state-of-the-art methods do not give yet satisfactory results in real transportation environments. Indeed, these latter configurations imply several difficulties such as fast brightness changes, noise, shadows, scrolling background, *etc.*, and a single approach cannot deal with all these. In this paper, we propose a new approach for people segmentation and tracking in videos that is suited for real-world conditions. Our strategy combines several state-of-the-art methods for people detection, silhouette appearance modeling and tracking. Each process also uses its own frame pre-processing pipeline. The optimal combination of the people classifiers used, as well as the optimal parameters of each of the combined methods, being too difficult to be determined altogether, a genetic algorithm is used to determine the optimal classifier parameters and their combination weights. The output of the latter is used as an initialization for a multi-frame graph-cut operating on superpixel graphs. Our proposed approach is evaluated on the BOSS European project database that was acquired in moving trains and that contains typical scientific locks encountered in real transportation systems.

Keywords: Classifier combination, people detection, people tracking, superpixel, graph-cut, segmentation, transportation environment.

* Cyril Meurie, cyril.meurie@ifsttar.fr

1 Introduction

Image processing applications in video surveillance are becoming more and more complex. Today, counting people is not enough and extracting people's silhouettes is now necessary to enable more complex applications such as people or action recognition. These applications are not only useful to improve security but also to prevent accidents on infrastructure or people. Indeed, automatic segmentation of people's silhouettes enables to generate large amounts of data that can be used to make complex statistics. However, extracting people from camera recordings is complex because each camera has its own characteristics (angle, resolution, position), especially in the case of transportation environments. In the literature, people extraction is usually directly preceded with a motion-based background subtraction strategy (see[1] for a review). This consists in examining the evolution of pixel colors between successive frames in order to detect fixed background versus moving objects in the foreground. Several features (such as color or texture) and techniques (such as Gaussian Mixture Model,[2] Fuzzy logic,[3,4] or Neural networks[5]) have been developed to represent the temporal evolution pixels. These methods give good results in the case of relatively controlled environments, but in complex situations such as transportation environments, the results of these methods tend to degrade considerably. Indeed, state-of-the-art methods cannot cope simultaneously with the appearance of several scientific locks (fast brightness changes, noise, shadows, scrolling background, *etc.*). Deep convolutional neural networks methods have also recently been used to tackle this problem[6,7] but they can not be considered for our application due to the lack of labeled data that we dispose. Moreover, transfer learning or data augmentation based methods are also not relevant due to the high variations of acquisition that are difficult to reproduce or simulate.

As it has already been done in the context of classifier fusion,[8–10] we propose in this paper to build upon state-of-the-art methods for people detection, silhouette appearance modeling and tracking, and to make the most of several of them by an efficient combination optimally determined by a genetic algorithm.[11]

This paper is organized as follows: In Sec. 2, the objective of our work that consists in extracting people as accurately as possible and its new contributions in comparison with previous works are introduced. The synopsis of the proposed approach including both the principle of classifiers' combination and genetic optimization are also described. Sections 3, 4, 5 respectively detail the detection, appearance and tracking based people classifiers and the parameters of the genetic algorithm. In Sec. 6, the combination of classifiers and the temporal graph cut clustering are developed. Before concluding this work, we present the experimental database, the optimal parameters and methods of the proposed strategy automatically determined by the genetic algorithm, and the people extraction results.

2 Proposed method

2.1 Motivations

As mentioned in the introduction, performing people silhouette segmentation in real-world conditions of transportation environments is very challenging. Existing state-of-the-art methods rely on different assumptions to extract people in video sequences that can be roughly grouped into motion-based methods (*e.g.*, background subtraction), detection-based methods (*e.g.*, HOG: Histogram of Oriented Gradients,[12] DPM: Deformable Part Model[13]), appearance-based methods (*e.g.*, Gaussian Mixtures) and tracking-based methods (*e.g.*, Lucas-Kanade). Unfortunately, with real-world conditions, since each method is designed from specific assumptions, they cannot be efficient for all the many different configurations that can be encountered (shadows, moving lights, fast brightness changes, noise, scrolling background, *etc.*). Fortunately, we can make the most of the information provided by several different people classifiers to build, by combination, a more efficient global classifier. This is the course we have been steering in this paper. Our previous work focused on people extraction based on background subtraction method associated to a graph cut clustering initialized by a color distribution model.[14] In this paper, five major contributions are added:

- The proposed approach is based on a temporal graph cut clustering operating on a superpixel segmentation;
- A combination of classifiers (detection, appearance and tracking based classifiers) is used to give the most appropriate information to initialize similarity and capacity of our temporal graph cut;
- Several state-of-the-art methods are used altogether to tackle specific locks appearing in transport environments such as shadows, moving lights, fast brightness changes, scrolling background behind windows;
- Experimental results are enhanced by testing two other sequences of the BOSS European database;

- A ground truth (3252 images) of three video sequences dataset has been handmade and is available for the community on our website[15] to ease reproducible research.

2.2 Synopsis of the approach

To segment the silhouette of people in video sequences, we rely on known state-of-the-art methods that are based on three different cues. The first cue is a detection-based cue that benefits from motion-based background subtraction and bounding box detection from Histogram of Gradients features. The second cue is an appearance-based cue that exploits color appearance modeling with color histograms and Gaussian Mixture Models. The third cue is a tracking-based cue that exploits tracking methods at different levels (pixel, superpixel and silhouette). Each of these cues is used to produce one or several probability maps by classification and estimates the position of people's silhouettes in video frames. However, pre-processing video frames can help in the classification of each cue. Indeed, this enables us to reduce the effects of fast brightness changes and noise. So, each cue classification is preceded by several preprocessing treatments. All these cues are then associated altogether with a weighted combination giving rise to a final unique probability map. This provides the class memberships of each pixel for the two classes people/background to discriminate. The latter probability map is then considered as an input for a multi-frame graph-cut superpixel clustering that enables us to delineate precisely the silhouette. The complete synopsis of the proposed approach is illustrated in Figure 1 and the optimal parameters of each block are given in Table 1 of Sec. 7. As can be seen, several different preprocessings can be used, each having different parameters. In addition, each cue-based people classifier has its own parameters. Given the large choice of parameters that we have to face with, we prefer to optimally determine them by exploring the space of possible solutions obtained with different parameters tuning. This search space being much too large to be fully explored, a genetic algorithm is then used to find the best configuration of parameters on training video sequences.

2.3 People Classifiers Inputs

Before entering into details of cue-based classification, we present common items used to pre-process video frames.

2.3.1 Superpixel segmented frames

The first step of our proposed method is to segment video frames into superpixels. This segmentation will be used during all the following steps both for cue-based classification and graph-cut clustering. This means that we will work at the superpixel level and not at the standard pixel level. This allows us, on the one hand, to reduce the processing time since the number of superpixels is much lower than the number of pixels, and on the other hand, to facilitate the extraction of the people by considering superpixel boundaries close to the border of people's silhouettes. To do this, we have chosen the SLIC[16] superpixel method because it allows us to obtain homogeneous regions without loosing the edge information (see Figure 2(b)). Given a frame I^t and its superpixel segmentation, a region adjacency graph $G^t = (R^t, E^t)$ is constructed where R^t is the set of nodes (regions of the segmented image, represented by the mean color of the region) and E^t is the set of undirected edges (connections between adjacent regions). Pixels of I^t will be denoted p_i^t. The number of pixels of a region will be denoted by $|R_i|$. When there is no ambiguity, super index t will be dropped, and for the sake of clarity we will use R_i instead of R_i^t to denote a given region

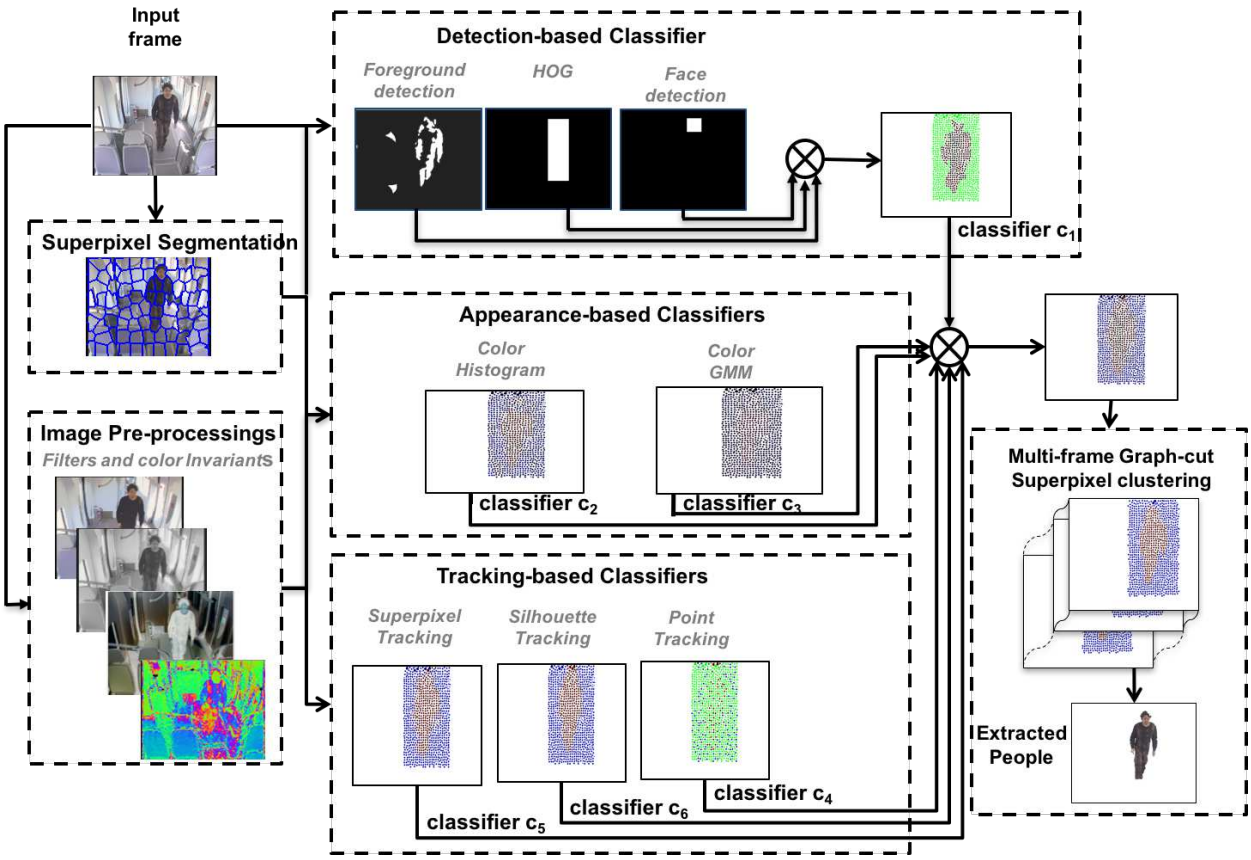


Fig 1 Synopsis of the proposed method of people extraction (where $\otimes$ corresponds to a weighted combination and where the people classifier outputs are marked in three colours : blue=background, red=foreground and green=undetermined).

of a frame. The barycentre of a region R_i^t will be denoted by b_i^t. The k-hop neighborhood of a region R_i is the set of regions that are reachable from R_i in k hops or fewer (that is following a path with k edges or fewer). The notation $R_i \sim_k R_j$ will be used to denote two regions that are k-hop connected.

2.3.2 Pre-processed frames

Cue-based classification can be eased by pre-processing original video frames. Indeed, frames can present some defects (due to recording in transportation environments) such as fast brightness changes or noise. The first can be reduced with the use of color invariant pre-processing and the second one with the use of filters. In our approach, we have considered different filters (blur, gaussian blur, median blur, and bilateral) and color invariants[17] (greyworld, reduced coordinates, $l_1l_2l_3$, $m_1m_2m_3$, affine normalization, and RGB rank). Pre-processing needs to be different from one cue-based people classifier to another (since the considered cue is different), and each cue-based classifier will have its own pre-processing step of the original frame under consideration. Determining which pre-processing, as well as which associated parameters, provides the best results for a given cue-classification is difficult and will be automatically determined by genetic optimization.

2.4 People Classifier Outputs

All considered cue-based people classifiers will take as input both pre-processed frames and associated superpixel segmentation. From these two inputs, they will perform a classification and estimate, for each region of the superpixel region adjacency graph, its probability of belonging either to the background or to the moving foreground object. Given a people classifier c_i, $P_{c_i}^{obj}(R_i^t)$ will denote the estimated probability of a region R_i^t to be a moving foreground object, and $P_{c_i}^{bg}(R_i^t)$ the probability of a region to belong to background. Given a set of k people classifiers, a weighted combination will be performed to obtain the global final probability estimation for each region: $P^*(R_i^t) = \sum_k \alpha_{c_i}^* P_{c_i}^*(R_i^t)$. This latter probability map brings node capacity information on each region and is used to initialize a graph-cut clustering method. Figure 1 presents people classifier outputs on a sample image: blue is for background, red is for foreground and green means undetermined. Details on computation of each people classifier probability map, their combination and use as capacities for graph-cut clustering will be provided in following sections.

2.5 Genetic Optimization

For each of the methods we consider, many different choices are possible. For instance, for the foreground detection used by the detection-based people classifier, up to twenty state-of-the-art algorithms can be considered. In addition, for pre-processing, different filters can be chosen and each filter has several parameters. Having such a large number of different configurations strongly motivates the use of an optimization strategy to determine the best setting. Keeping this in mind, it is now easy to understand that our proposed method has too many parameters to have them tuned by hand. This is especially true since we consider real and complex images, and a given set of parameters that performs well in a given situation, will not necessarily be efficient in other situations. Indeed, we will see later in this paper (see Table 2) that the best methods and parameters are not always the same as we consider different sequences of the BOSS project database. The search of the values of those parameters is called model selection in machine learning. This problem is very difficult to solve since the set θ of parameters to be tuned is very large and it is very hard to determine the set θ^* that optimizes a given quality criterion. This problem being not tractable, we have chosen to consider a meta-heuristic with the use of a genetic algorithm. In this paper, the genetic algorithm will be used at three levels:

- To determine the best parameters of the method involved in the proposed approach (for example, temporal graph-cut, silhouette tracking, pixel tracking, etc.);
- To optimize the choice of the state-of-the-art approach when several of them are considered and the choice of two preprocessing methods (filter and colorimetric invariant). For instance, in the case of detection-based people classifier, a filter (median), a colorimetric invariant (RGBrank) and a foreground detection method (MultiLayerBGS) are automatically chosen among many different choices;
- To optimize the combination weights of several optimized methods according to two cases. The first case concerns the combination of different methods of detection such as foreground detection, histogram of oriented gradients and face detection which is realized in a same classifier (Eq. 1). The second case deals with the combination of results of several people classifiers (Eq. 11): a detection-based classifier, two appearance-based classifiers and three tracking-based classifiers. More details on the genetic optimization will be given in Sec. 7.

In the sequel, we detail cue-based people classifiers as well as the possible choices of state-of-the-art methods.

3 Detection-based People Classifier

The first cue used to construct a people classifier is a detection-based one. We have considered a combination of foreground detection methods, learned statistical methods for people localization (HOG) and face detection. Although HOG-based methods[12] have highly developed in recent years, they are not the best suited to our problem. Indeed, the tested image sequences contain different angles with more or less important perspective effects that give unsatisfactory results. In addition, the learning phase of these methods on our sequences is difficult to efficiently implement due to the low number of images available for each sequence. In contrast, methods based on foreground detection appear to be better adapted to our sequences. We have nevertheless considered the two types of methods and we perform a weighted combination of three standard methods for people detection (foreground detection, HOG features classification, and face detection) in a single mask. The genetic algorithm chooses the best possible combination. Once this person detection is performed, two post-treatments are applied: a shadow detection to remove false positives generated by the shadows of people and a mathematical morphology step to remove small detection errors.

3.1 Foreground detection

Foreground detection is a well-established method in the literature. The state-of-the-art is very large[1] and it is very difficult to choose one method over others since each method is not always efficient for all situations. Therefore, we consider several possible methods and the best one will be chosen by the genetic optimization. The approaches we have retained are: neural networks,[5] fuzzy-based methods,[3,4] Gaussian mixture model methods,[2] statistical methods using both color and texture features[18] and a non parametric method.[19] All these methods come from the BGS library[20,21] and provide a result in the form of a binary mask.

3.2 HOG-based people detection

We use the state-of-the-art approach described by DALAL ET AL.[12] Histograms of Oriented Gradients (HOG) features are computed within pixel blocks that are classified by a Support Vector Machine classifier (SVM) using a linear kernel.[22] This method is robust to noise and light variations. To better detect people, the feature used is based on a concatenation of localized HOG extracted from a bounding box. HOG descriptor is computed on local windows divided into blocks and each one is divided into cells. This descriptor is then classified by an SVM with cross-validation. Settings of HOG and SVM are those suggested by DALAL ET AL.[12] People detection is performed on the initial image but uses overlapping sliding bounding boxes. This implies a set of possible people locations in the form of bounding boxes that we convert into a binary mask.

3.3 Face detection

In embedded environments the face is not so easily detected by the two previous people detection methods, so we also consider a state-of-the-art method for face detection based on facial landmark detection.[23] This method provides a set of possible face locations in the form of bounding boxes that we convert into a binary mask.

3.4 Detection combination

Given the three results of foreground detection, HOG-based people detection and face detection, we combine the binary masks they provide into a single one using a weighted combination. This combination is performed at the superpixel level and the probability map is defined as:

$$P_{c_1}^{obj}(R_i) = \frac{1}{|R_i|} \sum_{p_i \in R_i} \sum_k \beta_k^1 P_k^{obj}(p_i) \tag{1}$$

and similarly for $P_{c_1}^{bg}(R_i)$. The index k is the considered method among foreground detection, HOG-based people detection and face detection, and $\sum_k \beta_k^1 = 1$. Weights β_k^1 will be determined by the genetic optimization.

3.5 Shadow removal

Shadows are a classic problem in people detection mainly with methods of foreground extraction. Shadows are often detected as foreground because they have similar shapes and moves to people. The state of the art is large.[24] As for foreground detection, we have considered several possible methods and the best one will be chosen by genetic optimization. The approaches we have retained are: chromaticity based method,[25] physical method,[26] geometry based method,[27] and texture based method.[28] All these methods come from the library of SANIN ET AL.[24] To improve this shadow removal, we use it in conjunction with background learning that uses the same methods than for foreground detection. The obtained shadow mask is removed from the detection combination. After shadow removal, some small detection errors still remain and they are filtered by several mathematical morphology closing operations.

4 Appearance-based People Classifier

The second cue used to construct a people classifier is an appearance-based one. Here we construct two distinct classifiers that are both based on the color distribution estimation of two classes (people and background). We use two methods of color distribution estimation: color histograms and Gaussian mixture models. Each method provides its own probability map. Since video sequences are considered, the update of the colors models is performed with the result of the people silhouette segmentation provided by the detection-based classifier. The background is not modeled on the whole frame but on a bounding box around the silhouette of the previous frame. If a person appears for the first time in a frame, then the initialization of the models is performed from the result of foreground detection. We recall that each classifier takes frames pre-processed by one of the methods enumerated in Sec. 2.3.2.

4.1 Color histograms

The color histogram appearance models are composed of two color histograms (one for each class - people object and background) of 3×256 bins each (one per color channel). Using the color histogram models per channel of detected people, the belonging of each pixel of the frame to

people and background classes can be estimated (denoted as $P_k^{obj}(p_i)$ with k the color channel) and the superpixel probabilities deduced from:

$$P_{c_2}^{obj}(R_i) = \frac{1}{|R_i|} \sum_{p_i \in R_i} \sum_{k=1}^{3} \beta_k^2 P_k^{obj}(p_i) \tag{2}$$

and similarly for $P_{c_2}^{bg}(R_i)$. Parameters β_k^2 are coefficients affected to each histogram channel according to their number of pixels.

4.2 Gaussian mixture models

Two independent Gaussian mixture models (GMM)[29] using the classic EM algorithm are considered: one for each class: people object and background. Each class is modeled by five Gaussians. The initialization is obtained from a k-means. Using the GMMs, the belonging of a pixel to people and background classes can be estimated (denoted as $P_k^{obj}(p_i)$ with k each Gaussian) and the superpixel probabilities deduced from:

$$P_{c_3}^{obj}(R_i) = \frac{1}{|R_i|} \sum_{p_i \in R_i} \sum_{k=1}^{5} \beta_k^3 P_k^{obj}(p_i) \tag{3}$$

and similarly for $P_{c_3}^{bg}(R_i)$. Parameters β_k^3 are coefficients affected to each Gaussian according to the number of pixels of each Gaussian.

5 Tracking-based People Classifiers

The third cue used to construct a people classifier is a tracking-based one. Indeed, since we are dealing with video sequences, we can make the most of the temporal information to predict the position of people between frames using tracking information.[30] Three different tracking-based people classifiers are considered that work at different scales: pixel, superpixel and silhouette. They produce three probability maps: $P_{c_j}^{obj}(R_i)$ with $j \in \{4, 5, 6\}$. Each tracking method uses the result of the people silhouette segmentation on the previous frame. To initialize each $P_{c_j}^{obj}(R_i)$ when the tracking is performed for the first time, the result of the detection-based people classifier $P_{c_1}^{obj}(R_i)$ on the previous frame is used.

5.1 Pixel tracking

To perform pixel tracking between frames, we follow the approach of SHI AND AL.[31] In a video frame I_t at time t, points of interest q_i^t are extracted and their matching with the previous frame is performed with the classic Lucas-Kanade optical flow. To enhance the performance of the tracking, we set two parameters corresponding to the minimal accepted quality of interest points and the minimum possible Euclidean distance between matched points. These parameters will be tuned with genetic optimization. A matching function is then defined:

$$f(q_i^t) = \begin{cases} q_j^{t-1} \\ \emptyset \end{cases} \tag{4}$$

that associates, if possible, points from the actual frame to the previous one. Then, a matching between regions is performed from matching points:

$$h(R_i^t) = \{R_j^{t-1} | \exists (q_k^t \in R_i^t, q_l^{t-1} \in R_j^{t-1}), f(q_k^t) = q_l^{t-1}\} \tag{5}$$

This associates a set of regions, if possible, from the actual frame to the previous one. The probability of a region to belong to people and background classes can then be estimated from the matching regions. Since one region can have no match or several matches, an average is performed:

$$P_{c_4}^{obj}(R_i^t) = \begin{cases} \frac{1}{|h(R_i^t)|} \sum\limits_{R_j^{t-1} \in h(R_i^t)} P_{c_4}^{obj}(R_j^{t-1}) & \text{if } h(R_i^t) \neq \emptyset \\ 0 & \text{if } h(R_i^t) = \emptyset \end{cases} \tag{6}$$

and similarly for $P_{c_4}^{bg}(R_i^t)$.

5.2 Superpixel tracking

To perform superpixel tracking between frames, we use their barycentres position within consecutive frames. Given a frame I^t, $b(R_i^t)$ denotes the region on the previous frame that contains the barycentre of R_i^t, so one has:

$$b(R_i^t) = \{R_j^{t-1} | b_i^t \in R_i^{t-1}\}. \tag{7}$$

This matching being only spatial, it cannot be correct but this provides an approximate position useful in obtaining the correct matching. To determine which region of the previous frame located around $b(R_i^t)$ can be considered as the photometrically closest to R_i^t, we explore the k-hop around $b(R_i^t)$. This best matching region R_{l*}^{t-1} is obtained by:

$$R_{l*}^{t-1} = \underset{R_l^{t-1} \sim_k b(R_i^t)}{argmin} \|R_l^{t-1} - R_i^t\|_2 \tag{8}$$

where $\|R_l^{t-1} - R_i^t\|_2$ is the l_2 norm between the mean color of the two regions. The k-hop is 2 in order to limit processing time. Then, the probability of a region to belong to people and background classes can be estimated from the matching regions and we set:

$$P_{c_5}^{obj}(R_i^t) = \begin{cases} P_{c_5}^{obj}(R_{l*}^{t-1}) & \text{if } P_{c_5}^{obj}(R_{l*}^{t-1}) > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

and similarly for $P_{c_5}^{bg}(R_i^t)$.

5.3 Silhouette tracking

Silhouette tracking is used to predict the shape of the silhouette across frames. A silhouette is represented by a set of n key points $\{q_1^t, \cdots, q_n^t\}$ plus its barycentre q_0^t. These key points are used to draw the contour of the silhouette. The barycentre q_0^t of the silhouette is used as a stable reference across frames and key points q_i^t are obtained with a regular radial sampling of angle θ on the silhouette. The position of key points and barycentre are predicted with an Extended Kalman Filter (EKF).[32] The position of the barycentre is obtained by prediction on X and Y axes. The position of key points is obtained by prediction of the distance between the barycentre and the silhouette contour. Once points have been predicted, the silhouette S is reconstructed by linking key points with lines. All pixels inside the silhouette are considered as being an object. Since the estimation of class probabilities is performed at the superpixel level, a superpixel is considered as being part of the people silhouette if at least 60% of its area overlaps the silhouette. Then, the probability of a region to belong to people and background classes can then be estimated from the matching regions and we set:

$$P_{c_6}^{obj}(R_i^t) = \begin{cases} 1 & \text{if } \frac{|p_i^t \in (R_i^t \wedge S)|}{|R_i^t|} > 0.6 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

and similarly for $P_{c_6}^{bg}(R_i^t)$. The number of key points n and all the parameters of the EKF will be optimized with the genetic algorithm.

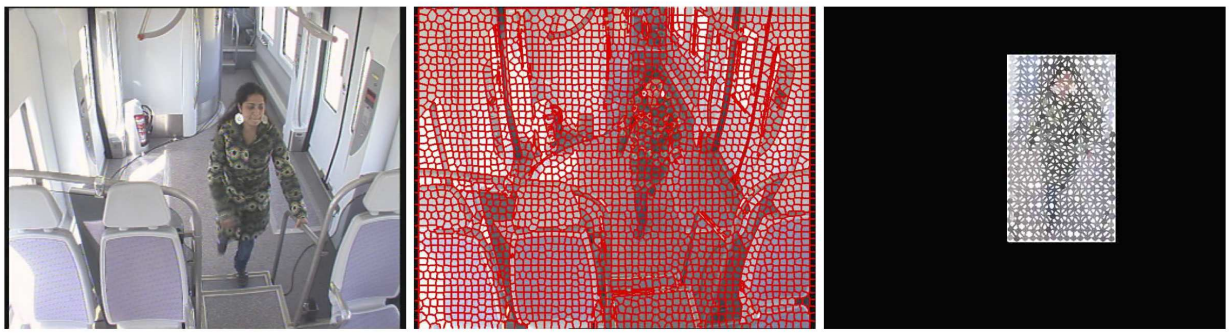
6 Multi-frame graph-cut from combined classifiers

6.1 Classifier Combination

Now that we have presented the six people classifiers we have considered, we detail the step of their combination. Given one people classifier c_k, $P_{c_k}^{obj}(R_i^t)$ provides the estimated probability of a region R_i^t to be a moving foreground object, and $P_{c_k}^{bg}(R_i^t)$ the probability of a region to belong to background. Given the set of 6 people classifiers we have introduced, a weighted combination is performed to obtain the global final probability estimation for each region:

$$P^{obj}(R_i^t) = \sum_{k=1}^{6} \gamma_{c_k}^{obj}(t) P_{c_k}^{obj}(R_i^t) \tag{11}$$

and respectively for $P^{bg}(R_i^t)$. The combination weights affected to each classifier are defined such that $\sum_{k=1}^{6} \gamma_{c_k}^{obj} = 1$ and similarly for $\gamma_{c_k}^{bg}$. The best combination weights will be determined by the genetic algorithm. The combination also uses an additional time-dependent function γ applied on weighting coefficients. Since we consider fixed cameras, during the first frames that a person appears, all people detection classifiers do not always enable good detection (this is especially the case for point tracking for instance). But as time goes by, detection becomes better and better until the person goes out of the camera field of view. The goal of using time-dependent function γ_{c_k} is therefore to adapt the classifiers' coefficients in function of their significance during the detection for a specific camera position. The function γ_{c_k} is defined in the form of a non-linear function approximated by three line segments. The coefficients of this learning function will also be tuned by the genetic algorithm. This enables us to verify the significance of the use of such a function.



(a) (b) (c)

Fig 2 Example of superpixel segmentation and graph: (a) original image, (b) SLIC superpixel segmentation superimposed in red and (c) reduced superpixel graph superimposed $\mathcal{G}^t$ (each region is shown with its mean color).

6.2 Connected-frame graph

After classifier combination, we use of a probability map for a given frame, estimated at the superpixel level. This information not being a segmentation, we use it to initialize a graph-cut clustering that operates on a specific graph. The latter is a superpixel graph that connects adjacent frames to make the most of temporal information.

To speed up the processing time of the graph cut clustering algorithm, we use a bounding box around the silhouette that was detected on the previous frame. For a given frame I^t, its superpixel graph $G^t = (R^t, E^t)$ is therefore reduced to $\mathcal{G}^t = (\mathcal{R}^t, \mathcal{E}^t)$ with $\mathcal{R}^t \subset R^t$ and $\mathcal{E}^t \subset E^t$. These subsets are obtained by retaining only regions that are within the bounding box, and the edges that connect them. Figure 2(c) shows such an example.

Finally a connected-frame graph $\mathcal{G}^* = (\mathcal{R}^*, \mathcal{E}^*)$ is created by connecting adjacent frames I_t to I_{t-n}. The interest in this construction is based on the fact that the segmentation obtained on the previous frames can ease the segmentation on the actual frame. We explain how is constructed the connection between two adjacent graphs of frames $\mathcal{G}^t$ and $\mathcal{G}^{t-1}$, the principal being the same to connect $\mathcal{G}^{t-i}$ and $\mathcal{G}^{t-i-1}$. To do so, the same procedure that was used in Sec. 5.2 is performed. A matching of each region $\mathcal{R}^t$ with regions of $\mathcal{G}^{t-1}$ is obtained. The matching region $\mathcal{R}_l^{t-1}$ of a region $\mathcal{R}_i^t$ is then connected to $\mathcal{R}_i^t$ and all the regions within a k-hop neighborhood around $\mathcal{R}_l^{t-1}$ are connected to $\mathcal{R}_i^t$. The optimal size of the k-hop neighborhood will be determined by the genetic algorithm and remains the same for all adjacent frames to be connected. Since we connect adjacent frames I_t to I_{t-n} with n defined by the genetic algorithm, one has therefore:

$$\mathcal{R}^* = \mathcal{R}^t \cup \mathcal{R}^{t-1} \cup \cdots \cup \mathcal{R}^{t-n} \tag{12}$$

and

$$\begin{aligned} \mathcal{E}^* = \; & \mathcal{E}^t \cup \mathcal{E}^{t-1} \cup \{\mathcal{R}_i^t \sim_k \mathcal{R}_l^{t-1}\} \cup \cdots \\ & \cup \mathcal{E}^{t-n+1} \cup \mathcal{E}^{t-n} \cup \{\mathcal{R}_i^{t-n+1} \sim_k \mathcal{R}_l^{t-n}\}. \end{aligned} \tag{13}$$

6.3 Graph-cut clustering

Graph cuts[33] are a powerful segmentation algorithm that enables binary clustering of a graph. It consists in formulating the clustering problem as an energy minimization in the form of a labeling problem.[34] In this paper, we have used the min-cut/max-flow implementation.[35] As a result, the superpixel connected frame graph $\mathcal{G}^* = (\mathcal{R}^*, \mathcal{E}^*)$ is classified into two classes starting with an initialization of vertices labels in sources (*i.e.*, foreground object) and sinks (*i.e.*, background), given the result of the clustering on the previous frame. This means that each node $\mathcal{R}_i^* \in \mathcal{R}^*$ is assigned a binary label $l_i \in \{obj, bg\}$. To perform the graph-cut clustering, we assign a capacity to each node of $\mathcal{R}^*$ for these two classes and a similarity for each edge of $\mathcal{E}^*$. Given both, the minimum $\hat{\imath}$ of the energy shown below corresponds to the best segmentation among the set F of all possible labeling solutions:

$$\hat{\imath} = \arg\min_{l \in F} \left(\sum_{\mathcal{R}_i \in \mathcal{R}^*} W^{l_i}(\mathcal{R}_i) + \sum_{\mathcal{R}_i \in \mathcal{R}^*} \sum_{\mathcal{R}_i \sim_1 \mathcal{R}_j} S(\mathcal{R}_i, \mathcal{R}_j) \cdot \delta_{l_i \neq l_j} \right) \tag{14}$$

where $S(\mathcal{R}_i, \mathcal{R}_j)$ is the similarity between two superpixel regions, $W^{l_i}(\mathcal{R}_i)$ is the capacity of a node, and the term $\delta_{l_i \neq l_j}$ in the second sum is the Potts prior that encourages piecewise-constant labelling. Each label l_i corresponds to either background or object. The capacities are obtained directly from the probabilities obtained from the step of classifier combination:

$$W^{obj}(R_i) = -\log(P^{obj}(R_i)) \tag{15}$$

and similarly for $W^{bg}(R_i)$. The similarity $S(\mathcal{R}_i, \mathcal{R}_j)$ between two regions $\mathcal{R}_i$ and $\mathcal{R}_j$ is given by:

$$S(\mathcal{R}_i, \mathcal{R}_j) = \frac{\exp\left(-\frac{\|\mathcal{R}_i - \mathcal{R}_j\|_2}{2\theta^2}\right)}{\|b_i - b_j\|_2} \tag{16}$$

where $\|.\|_2$ is the Euclidean distance between the barycentres of the regions. When comparing regions, mean colors are used. The θ coefficient is a bandwidth similarity parameter that will be fixed by the genetical algorithm. To perform the minimization, the min-cut/max-flow implementation[35] is used.

7 Experimental results

7.1 BOSS European Dataset

To test the performance of our proposed approach, we have considered a video database that has been shot in real transportation environments. In literature, to the best of our knowledge, no video database in real transportation environments (inside a vehicle such as train, tramway, bus, *etc.*) currently exists with an associated ground truth mandatory to perform both training for machine learning and segmentation quality evaluation. We have therefore considered video sequences from the BOSS European project database.[36] Video sequences were recorded inside a train in motion. They contain many difficulties for people silhouette segmentation. First, video sequences were shot during a sunny afternoon and according to the position of the camera one can see only inside the train or also outside through windows. Since the train is moving, many moving elements can be seen through windows with a speed effect when the train runs fast. Second, because of the sun, the

(a) (b) (c)

Fig 3 Sample video frames of the BOSS project database:[36] (a) sequence 1, (b) sequence 2 and (c) sequence 3.

good weather brings a lot of shadows: shadows projected through windows of objects outside the train, and shadows of objects inside the train. Third, the train speed causes a lot of fast brightness changes. Fourth, several people move inside the train. They wear different clothes (dress, shirt, jean, pants) with different textures (united to complex patterns), and some people carry accessories (*e.g.*, bags). During their displacement in the train, people can have special behaviors such as running, turning round or making large movements with their arms. Figure 3 illustrates three video sequences of the BOSS European project. All these elements render video sequences of the BOSS project database very challenging for people silhouette segmentation. Anyway, there is no associated ground truth where on each frame the people' silhouettes are delineated. The selected video dataset contains a total of 12920 frames (720×576 pixels) and is divided into three sequences that contain the scientific locks we have just mentioned: 4257 frames in sequence 1, 4585 frames in sequence 2, 4078 frames in sequence 3. The first sequence contains 12 people displacements with front camera shooting. The second one contains 11 people displacements with perspective camera shooting. The third sequence contains 11 people displacements with side camera shooting equipped with fish-eye lens. In each sequence, only some frames contain moving people, the other ones being empty. We have manually created 3252 reference segmentations corresponding to the crossing of the 12 persons in the camera's field of view: 1439 frames in sequence 1; 1117 frames in sequence 2; 996 frames in sequence 3. Whether for the training, testing or evaluation phase, we only consider frames containing people. As it will be discussed in details in Sec. 7.3, the dataset in not shared for training and testing since we use a specific Leave-One-Out (LOO) procedure that leaves one person out of the sequence for training and tests on the remaining ones. This database and the ground truth of the video sequences are available at our website.[15] Providing such a database is also one strong contribution of this paper and future research can benefit from it as well as comparing with the results we have obtained.

7.2 Genetic Optimization

As we have previously mentioned it, the whole strategy we propose involves a lot of different possibilities of methods and associated parameters, and combination weights. A genetic optimization is used to automatically determine the best configurations and optimize our proposed approach. It is performed with a population of 24 chromosomes that encode possible solutions. Each chromosome corresponds to a complete setting of our proposed people extraction method. A chromosome

is divided into several blocks that are composed of one or several genes. For instance, for the filtering preprocessing, each gene encodes either the use of a method (binary gene) or the value of a parameter (quantized possible values). Figure 4 illustrates a four genes encoding of the use of a filter block in the proposed approach as well as the two steps of crossover and mutation. In this figure, the first gene corresponding to the chosen filter method is illustrated in green color, the associated parameters of the filter are illustrated in yellow and those that are not used are marked in grey. The genetic algorithm uses a standard configuration and begins with an initialization step and iteratively processes steps of selection, crossover and mutation steps until the population is stable.

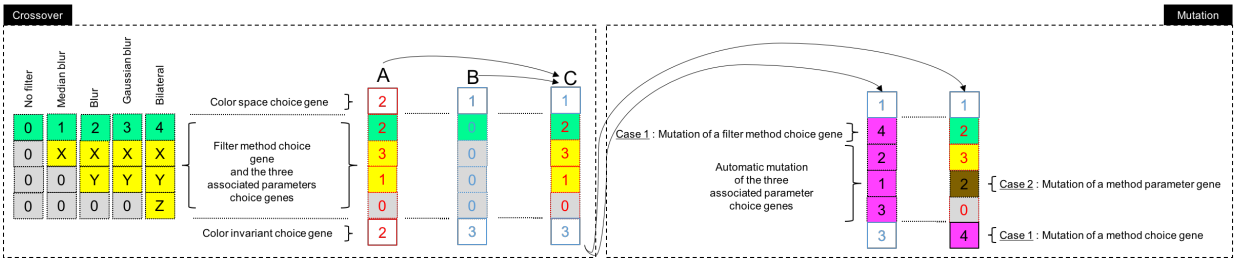


Fig 4 Example of crossover and mutation steps of our genetic algorithm.

We detail these steps in the sequel:

- The initialization step constructs a list of candidate solutions (called population). The initialization of the chromosome is made by block. Thus, the genes corresponding to a choice of method are first randomly initialized. Then, the genes corresponding to methods parameters are in turn randomly initialized. Finally, if certain genes are not used, they are set to zero. As an example, let us consider the blur filter illustrated in Figure 4, the first gene of the parent A corresponds to the filter choice ($value = 2$) and is marked in green, the next two genes correspond to the associated parameters ($size_X = 3$ and $size_Y = 1$) and are illustrated in yellow, and the last gene which is not used for this type of filter is set to 0 and marked in grey.

- The crossover step aims at generating new candidate solutions from existing ones in the population. A child is produced from mixing two chromosomes randomly chosen. Figure 4 illustrates an example of the crossover step in our genetical algorithm. The child C is obtained by successively copying the block of one of the two parents A or B. A random sampling is used to determine the block to copy. In our example, the child C obtains the color space choice and the color invariant choice genes of the parents B marked in blue and the filter method and the associated parameters choice genes (corresponding to the blur filter) of the parents A marked in red.

- The mutation step consists in slightly modifying a part of the new chromosome generated in the previous crossover step. It is motived to obtain new chromosomes different from parents in order to cover a wide range of solutions. To increase this phenomenon, we have chosen the following configurations: i) in the case of a mutation of a gene corresponding to the choice of a method (case 1 in Figure 4), the selected gene (which is marked in pink) is randomly modified with a mutation rate of 25% and the others genes of the considered block are re-initialized (as described in the initialization step); ii) in the case of a mutation of a gene

corresponding to a parameter of a method (case 2 in Figure 4), the selected gene (which is marked in brown) is randomly modified with a mutation rate of 25% or a small value is added within an interval that is plus or minus 10% of the actual value with a mutation rate of 50%. Since several parameters of our proposed method are linked together (*e.g.*, a filtering method and its parameters) the steps of crossover and mutation are specially designed to combine them correctly. These parameters values are selected in order to enable a fast convergence of the genetic algorithm.

- For the selection step, selected candidates (the first half of the generated chromosomes) are kept in order to obtain a stable size of population for each generation. The other ones (the second half) are rejected. The fitness measure used to sort the population corresponds to an average of all F-Measure scores of the sequence determined for each image by comparing the segmentation result of the initial image with its handmade ground truth. In the state-of-the-art, the F-Measure score is usually used to evaluate results of people segmentation on one image. But, in our application, we must adapt this criterion to the use of videos composed of several sets of images, where each one characterizes 11 different people displacements in the train. To that aim, the displacement of one people is evaluated by an average of the F-Measure computed on each image. Then, the video sequence (where many people appears) is evaluated by an average of the F-Measure of each people. The fitness function uses this strategy. The genetic algorithm is stopped when the best candidate of the population has not changed during ten generations.

$$F - Measure = 2 \times \frac{precision \times recall}{precision + recall} \tag{17}$$

$$recall = \frac{true\ positives}{true\ positives + false\ negatives} \tag{18}$$

$$precision = \frac{true\ positives}{true\ positives + false\ positives} \tag{19}$$

7.3 Extraction of right candidates

Each video sequence corresponds to a specific camera position in the train, and therefore we might expect that having a common strategy of people silhouette segmentation for all silhouettes will be very difficult. Indeed, given a camera position, some specific processings might be necessary (*e.g.*, by applying a colorimetric invariant) because of light changes whereas in other configurations, this is not the case. In addition, camera shooting is very different from one sequence to another and this complicates considerably the conception of a common segmentation strategy.

To assess this assumption, we first have conceived a segmentation strategy per video sequence (*i.e.*, per camera position). This means that we perform a genetic optimization for each sequence. However, the number of people in each sequence is not so high (around eleven) with around hundred frames for each person. To better evaluate the performance of our proposal, we use a specific Leave-One-Out (LOO) procedure that leaves one person out of the sequence for training and tests on the remaining ones. Given a video sequence that contains k different persons, we take all the

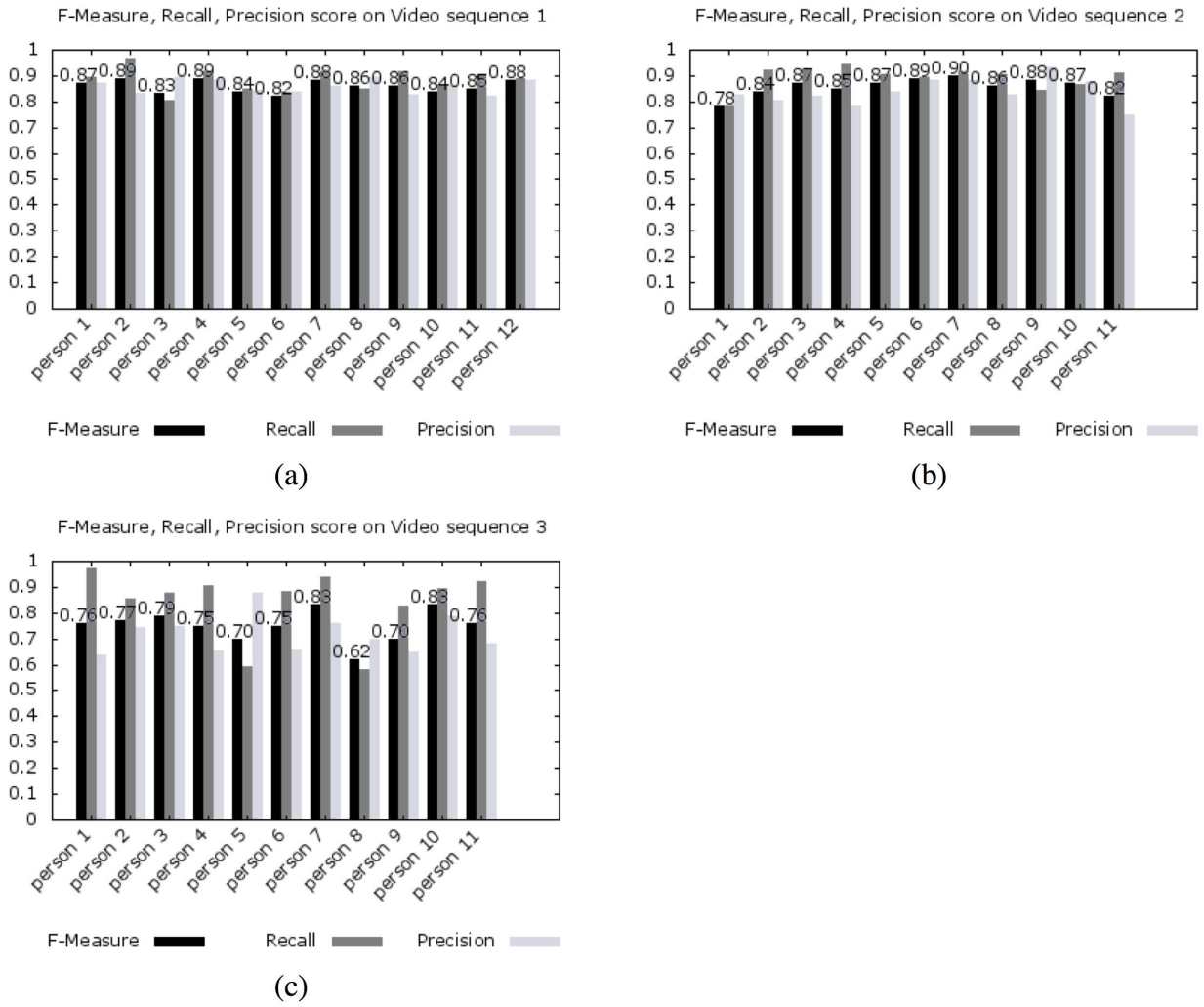


Fig 5 F-Measure scores obtained with People LOO cross validation on the three sequences (a), (b) and (c) of the BOSS project database.

frames that contain a given person P_i and perform the genetic optimization of our strategy to obtain the best configuration. This configuration is tested on the remaining frames that contain other people P_j (with $j \in \{1, .., k\} \setminus \{i\}$) in the video sequence and an F-Measure is obtained for each person. This is performed for all the possible values of $i \in \{1, .., k\}$ and an average F-Measure for each person is obtained from all the results. The genetic optimization was realized on a cluster of 96 cores (2 Ghz - 768Go).

Figure 5 shows the LOO results obtained on the three sequences for each person. Results on two first sequences (a) and (b) are very good with an F-Measure close to 0.9. Moreover, recall and precision are also very good and the gap between them is small to assess the reproducibility of the approach. Results on the third (c) sequence are more mitigated. Indeed the results are less similar and 8 scores out of 11 are above an F-Measure of 0.75. For three persons (5, 8 and 9) the F-Measure is below 0.70. These less good performances can be explained by the difference between the people used for genetic optimization and used for training. However, a good configuration

of the strategy exists since, with a genetic optimization performed on person 7, an F-Measure of 0.83 is obtained. This shows that the genetic optimization is an essential step of our approach and this enables us to find good configurations of the proposed strategy. Figure 9 illustrates the people extraction results obtained with our proposed method. We can notice that results of people extraction are very satisfactory, and robust, whatever the angle of view used. These fine results could be used in our future works: people re-identification.

7.4 Summary of right candidates setting

The optimization strategy that we perform has generated and evaluated several thousands of settings. Figure 6 illustrates the best settings for a given time processing. The evaluation function used to place the points in this figure is the same as that used for the genetic optimization step. One can notice that the processing time is high but the fitness function (and the implementation code) was not thought to reduce the computation time in order to achieve to a real time application but more to define the optimal parameters and methods of the proposed approach. Thus, we have kept the swiftest optimizations (marked by black points in Figure 6) which obtain an F-Measure upper to a manually set threshold (0.80 for the two first sequences (a) and (b) and 0.70 for the third sequence (c)).

Table 1 presents the main parameters and methods. We can see that optimums kept on a given sequence does not share much common configuration items with another. This shows that our assumption was true and it is preferable to have one optimal configuration per camera instead of a common one.

Figure 7 shows the evolution of the time coefficient for the combination step. For the two first sequences, the people-based detection classifier always has the highest combination weights. In the sequence (a) where the camera shoots front, the evolution of the coefficient in time is not so high, which is normal since there is little evolution and all classifiers perform similarly. In the sequence (b), the evolution is visible and coefficients grow during the displacement of the person. Again, this was expected since there is a perspective shooting effect and both shape and appearance of the person become more reliable as the person moves forward in the train. In contrast, for the third sequence, the appearance-based Histogram classifier always has the best combination weights. One can see that, for the third sequence, we have an evolution of the coefficients that is in between sequences (a) and (b). In addition, the optimal combination is very different from that of other sequences. Indeed, for the third sequence, this optimal configuration undoubtedly shows that an accurate classifier for one given camera position is not accurate for another camera position anymore. This justifies our strategy towards a camera position-dependent approach.

7.5 Best candidate comparison between sequences and well-known methods

We begin by choosing one setting for each sequence which has been trained during the LOO procedure. Each sequence contains k persons, and we obtain k different optimal configurations θ_k. To select the best one θ^*, we choose the configuration that performs the best on the whole sequence with the k persons. This gives us the optimal configuration for each sequence. We have determined that: the optimum trained with person 5 of sequence 1 is the best with an F-Measure score of 0.85, the optimum trained with person 2 of sequence 2 is the best with an F-Measure score of 0.86, and the optimum trained on person 7 of sequence 3 is the best with an F-Measure score of 0.80.

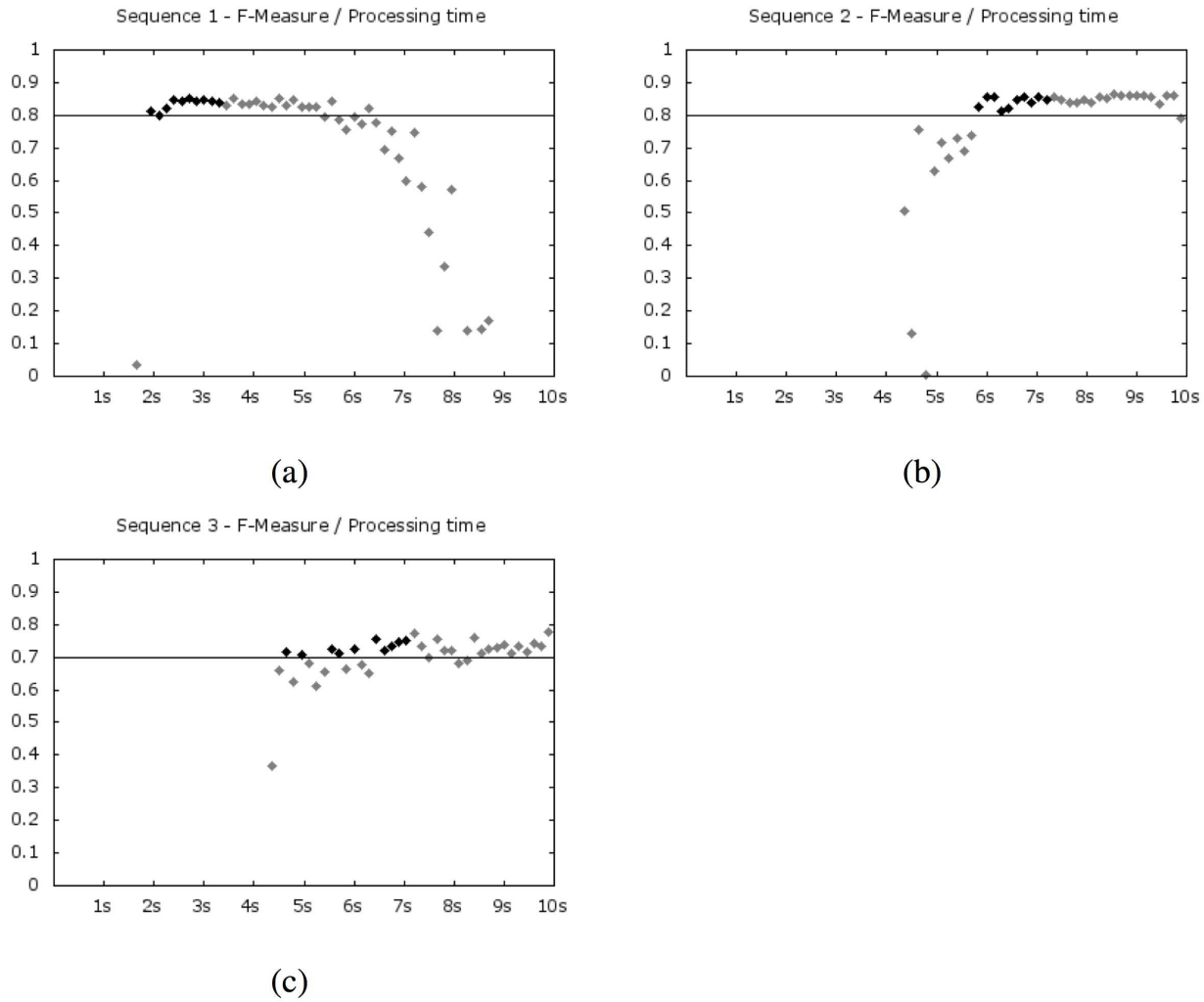


Fig 6 Processing time by frame for each setting generated with the genetic algorithm for sequences (a), (b) and (c) of the BOSS project database.

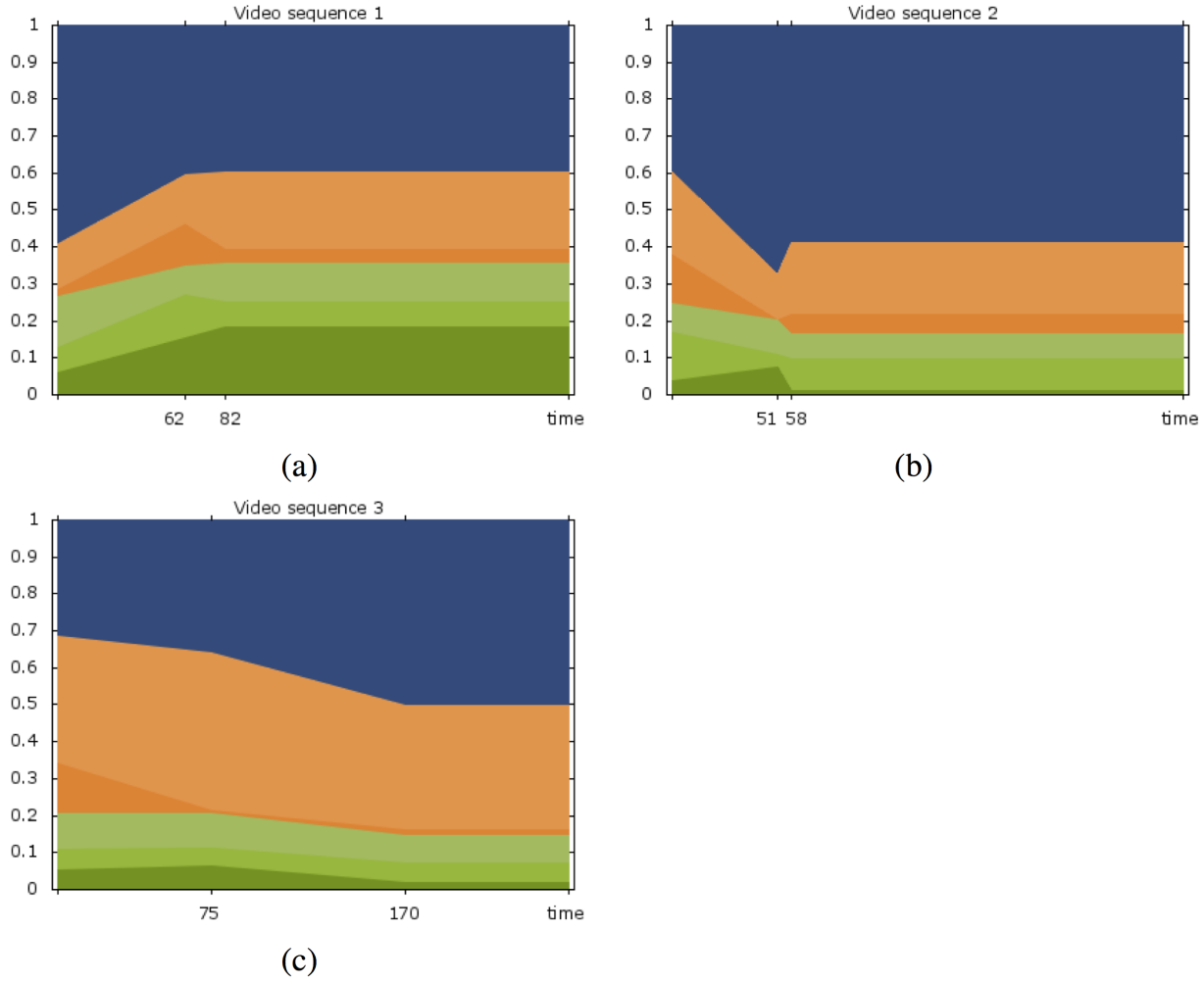


Fig 7 Time-Dependent coefficient for people classifier combination. Curves show the evolution in time of $\gamma_{c_i}(t, \alpha_{c_i}^{obj})$ with $i \in \{1-6\}$. Each color corresponds to a people classifier (blue: c_1 – detection-based classifier, light orange: c_2 – Appearance-based Histogram classifier, dark orange: c_3 – appearance-based GMM classifier, green: c_4 – Pixel Tracking-based People Classifier, light green: c_5 – Superpixel Tracking-based People Classifier, dark green: c_6 – Silhouette Tracking-based People Classifier).

Table 1 Best methods and parameters of the proposed strategy obtained with genetic optimization for the three sequences (a), (b) and (c) of the BOSS project database.

Detection-based People Classifier		**Sequence 1 (a)**	**Sequence 2 (b)**	**Sequence 3 (c)**
Foreground detection	Filter	Gaussian	Median	Median
	Invariant	Grey World	RGBRank	Ø
	Method	MultiLayerBGS[18]	MultiLayerBGS[18]	MultiLayerBGS[18]
Background learning	Filter	Blur	Gaussian	Median
	Invariant	Grey World	Reduced Coords	Grey World
	Method	FuzzyChoquetIntegral[5]	LBMixtureofGaussian[2]	FuzzyChoquetIntegral[5]
Shadow Removal	Filter	Gaussian	Gaussian	Ø
	Invariant	$m_1 m_2 m_3$	RGBRank	$m_1 m_2 m_3$
	Method	Chromacity[25]	LrTexture[28]	Chromacity[25]
Appearance-based People Classifier		**Sequence 1**	**Sequence 2**	**Sequence 3**
Color Histograms	Filter	Bilateral	Gaussian	Bilateral
	Invariant	GreyWorld	RGB Rank	RGB Rank
Gaussian Mixture Models	Filter	Median	Bilateral	Bilateral
	Invariant	Affine normalization	RGB Rank	RGB Rank
Tracking-based People Classifier		**Sequence 1**	**Sequence 2**	**Sequence 3**
Superpixel tracking	Filter	Gaussian	Gaussian	Blur
	Invariant	Affine normalization	Affine normalization	GreyWorld
Silhouette tracking	#Points	584	194	779
	EKF Points	Position	Velocity	Acceleration
	EKF Center	Position	Position	Position
Multi-frame Graph-cut Superpixel clustering		**Sequence 1**	**Sequence 2**	**Sequence 3**
Graph	#Connected-frames	4	3	3
	k-hop	3	3	3

Finally, we have tested the optimal combination and parameters determined for the first sequence on the two others sequences of the BOSS European project (with other types of camera or angle of view). We have also tested our method on two strategies for people segmentation.[37,38] These methods provide a high number of false positives and do not seem to be not adapted for our complex transportation dataset including many locks. This is why, we have chosen to illustrate several settings of the proposed method and some well-known background subtraction methods in Figure 8. For this comparison, we have tested: i) three basic methods: Fuzzy Gaussian[4] (called LBFuzzyGaussian), Fuzzy Sugeno Integral[39] (called FuzzySugenoIntegral) and Fuzzy Choquet Integral[3] (called FuzzyChoquetIntegral); ii) one statistical method using multiple Gaussians: Gaussian Mixture Model[2] (called LBMixtureofGaussians); iii) one statistical method using color and texture features: Multi-Layer BGS[18] (called MultiLayerBGS); iv) one non-parametric method: VuMeter;[19] v) one neural method: Adaptive SOM[5] (called LBAdaptiveSOM). We would like to clarify that the parameters of these state of the art methods, like those of the proposed method, have been genetically optimized. To know the list of optimized parameters of each method, the reader can refer to Table 3 of the SOBRAL AND AL.'s paper.[1] One can notice that the optimum found on a given sequence performs always much better on that sequence than the one found on the other sequences. This can be explained by the difference of camera angle between sequences: in sequence 1, the camera is front shooting, in sequence 2, the camera is three-quarter shooting, and in sequence 3, the camera is side shooting and the lens is particular. This also explains why the optimum found on the second sequence performs the best on the other two sequences. Again, this shows that having an optimal configuration per camera is preferable. In addition, we can conclude that several techniques well-known of background subtraction do not offer good results for this application, and our complete strategy play its full role. Regarding the computation time, once the choice of the optimal parameters has been realized, the proposed approach can be implemented in a smartest way than the generic approach used for parameter tuning. This dedicated implementation has been tested on a laptop equipped with one CPU thread cadenced at 3.4 Ghz, has an average

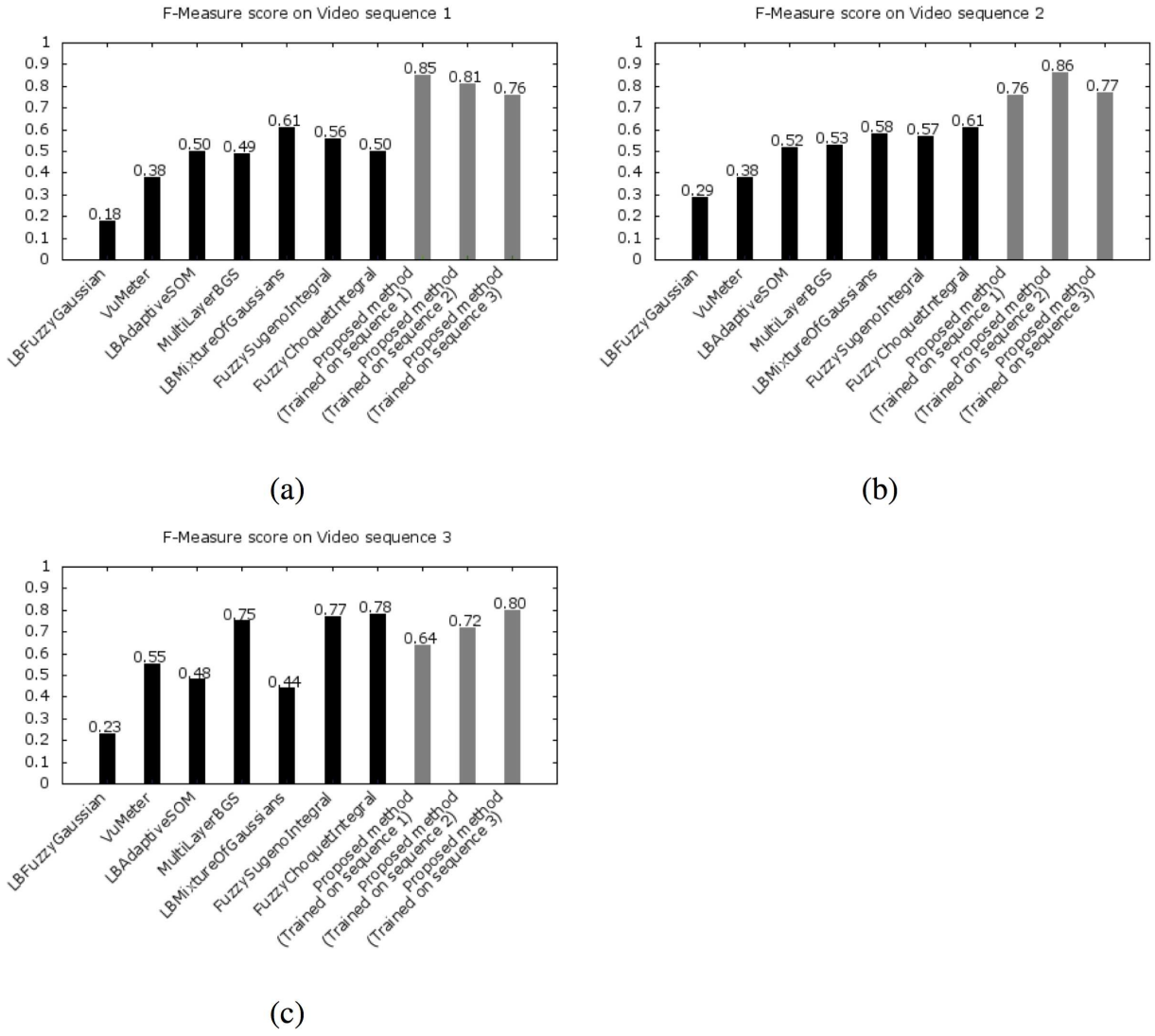


Fig 8 Comparison of F-Measure scores obtained with our proposed genetically optimized method and best method of foreground detection of the state-of-the-art on three sequences (a), (b) and (c) of the BOSS project database.

computation time of 180ms per image. It is actually longer than the two state of the art approaches of MIGNIOT AND AL.[40] (which does not use any people detector and requires a computation time of 16ms) or YANG AND AL.[41] (which uses only one people detector and requires a computation time of 90ms), but our approach that combines many detectors, offers better segmentation performances. Nevertheless, we think that this computation time could be strongly reduced by using GPGPU implementation in future works.

8 Conclusion

In this paper, we have proposed a strategy that combines several state-of-the-art methods for people segmentation with detection-based, appearance-based and tracking-based approaches. The optimal combination of the people classifiers and the parameters of these used people classifiers being

Fig 9 People extraction image test results with genetic optimization done on the three sequences (a), (b) and (c) of the BOSS project database.

difficult to determine altogether, a genetic algorithm is used to obtain the optimal configuration and combination of the classifiers. A temporal graph-cut based clustering is used to delineate peoples' silhouettes from estimated probabilities of the combined people classifiers. The proposed approach has been tested on a video database shot in real transportation conditions, for which we have manually constructed a silhouette ground-truth available at our website.[15] We have shown that: (i) given different camera configurations, it is preferable to optimize the strategy optimally for each camera, (ii) the obtained optimal strategy always performs better than foreground detection state-of-the-art methods. Some segmentation errors still remain and future works will consist in enhancing the appearance model associated to superpixels: a simple color mean was used and this is not sufficient for some complex situations. The proposed method can be adapted to the setting of hyper parameters of any method, that is why, in future works, it could be interesting to combine deep learning approaches. Another perspective will be to integrate intrinsic parameters of the camera. Moreover, due to the low consumption of time processing, it could be useful to define a real-time strategy including the appearance-based classifier and the processing time in the fitness function of the genetic algorithm.

References

1 A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Computer Vision and Image Understanding* **122**, 4 – 21 (2014).

2 T. Bouwmans, F. E. Baf, and B. Vachon, "Background modeling using mixture of gaussians for foreground detection - a survey," *Recent Patents on Computer Science* **1**(3), 219–237 (2008).

3 F. E. Baf, T. Bouwmans, and B. Vachon, "Fuzzy integral for moving object detection," in *FUZZ-IEEE*, 1729–1736 (2008).

4 M. Sigari, N. Mozayani, and H. Pourreza, "Fuzzy running average and fuzzy background subtraction: Concepts and application," *International Journal of Computer Science and Network Security* **8**(2), 138–143 (2008).

5 L. Maddalena and A. Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications," *IEEE Transactions on Image Processing* **17**(7), 1168–1177 (2008).

6 C. Song, Y. Huang, Z. Wang, *et al.*, "1000fps human segmentation with deep convolutional neural networks," in *3rd IAPR Asian Conference on Pattern Recognition*, (2015).

7 P. Luo, X. Wang, and X. Tang, "Pedestrian parsing via deep decompositional network," in *IEEE International Conference on Computer Vision*, (2013).

8 B. Gabrys and D. Ruta, "Genetic algorithms in classifier fusion," *Applied soft computing* **6**(4), 337–347 (2006).

9 K. Sirlantzis, M. Fairhurst, and M. Hoque, *Genetic algorithms for multi-classifier system configuration: a case study in character recognition*, 99–108. Springer (2001).

10 S. Günter and H. Bunke, "Optimization of weights in a multiple classifier handwritten word recognition system using a genetic algorithm," *Electronic Letters on Computer Vision and Image Analysis* **3**(1), 25–41 (2004).

11 J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, MIT press (1992).

12 N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition*, 886–893 (2005).

13 P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1–8, IEEE (2008).

14 C. Coniglio, C. Meurie, O. Lézoray, *et al.*, "A genetically optimized graph-based people extraction method for embedded transportation systems real conditions," in *17th International Conference on Intelligent Transportation Systems*, 1589–1595 (2014).

15 C. Meurie, O. Lezoray, C. Coniglio, *et al.*, "Ground truth of the boss project video database." http://www.ifsttar.fr/menu-haut/annuaire/fiche-personnelle/produits/personne/meurie-cyril/?no_cache=1#.

16 R. Achanta, A. Shaji, K. Smith, *et al.*, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(11), 2274–2282 (2012).

17 S. D. Hordley, G. D. Finlayson, G. Schaefer, *et al.*, "Illuminant and device invariant colour using histogram equalisation," *Pattern Recognition* **38**, 2005 (2005).

18 J. Yao and J. M. Odobez, "Multi-layer background subtraction based on color and texture," in *Computer Vision and Pattern Recognition, Workshop on Visual Surveillance*, 1–8 (2007).

19 Y. Goyat, T. Chateau, L. Malaterre, *et al.*, "Vehicle trajectories evaluation by static video sensors," in *IEEE International Conference on Intelligent Transportation Systems*, 864–869 (2006).

20 A. Sobral, "BGSLibrary: An opencv c++ background subtraction library," in *IX Workshop de Visao Computacional*, (2013).

21 A. Sobral and T. Bouwmans, *BGS Library: A Library Framework for Algorithm's Evaluation in Foreground/Background Segmentation*, 1–16. CRC Press, Taylor and Francis Group. (2014).

22 M. A. Hearst, S. T. Dumais, E. Osman, *et al.*, "Support vector machines," *Intelligent Systems and their Applications, IEEE* **13**(4), 18–28 (1998).

23 P. A. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision* **57**(2), 137–154 (2004).

24 A. Sanin, C. Sanderson, and B. C. Lovell, "Shadow detection: A survey and comparative evaluation of recent methods," *Pattern Recognition* **45**(4), 1684 – 1695 (2012).

25 R. Cucchiara, C. Grana, M. Piccardi, *et al.*, "Detecting moving objects, ghosts and shadows in video streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**, 1337–1342 (2003).

26 J. Huang and C. Chen, "Moving cast shadow detection using physics-based features," in *Computer Vision and Pattern Recognition*, 2310–2317 (2009).

27 J. Hsieh, W. Hu, C. Chang, *et al.*, "Shadow elimination for effective moving object detection by gaussian shadow modeling.," *Image and Vision Computing* **21**(6), 505–516 (2003).

28 A. Leone and C. Distante, "Shadow detection for moving objects based on texture analysis," *Pattern Recognition* **40**(4), 1222–1233 (2007).

29 R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Ed)*, Wiley (2001).

30 A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys* **38**(4), 13 (2006).

31 J. Shi and C. Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition*, 593–600 (1994).

32 S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *AeroSense'97*, 182–193, International Society for Optics and Photonics (1997).

33 Y. Boykov and M. Jolly, "Interactive graph cuts for optimal boundary region segmentation of objects in n-d images," in *International Conference on Computer Vision*, **1**, 105–112 (2001).

34 O. Lézoray and L. Grady, Eds., *Image Processing and Analysis with Graphs: Theory and Practice*, Digital Imaging and Computer Vision, CRC Press / Taylor and Francis (2012).

35 Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(9), 1124–1137 (2004).

36 "Boss european project (on board wireless secured video surveillance)." https://www.multitel.be/projets/boss/ or http://celtic-boss.mik.bme.hu or https://www.celticplus.eu/project-boss/.
37 M. D. Rodriguez and M. Shah, "Detecting and segmenting humans in crowded scenes," in *15th international conference on Multimedia*, 353–356, ACM (2007).
38 A. Milan, L. Leal-Taixé, K. Schindler, *et al.*, "Joint tracking and segmentation of multiple targets," in *IEEE Conference on Computer Vision and Pattern Recognition*, 5397–5406 (2015).
39 H. Zhang and D. Xu, "Fusing color and texture features for background model," in *Fuzzy Systems and Knowledge Discovery*, *LNCS* **4223**, 887–893, Springer Berlin Heidelberg (2006).
40 C. Migniot, P. Bertolino, and J.-M. Chassery, "Automatic people segmentation with a template-driven graph cut," in *ICIP*, 31493152 (2011).
41 C. Yang, L. Zhang, H. Lu, *et al.*, "Saliency detection via graph-based manifold ranking," in *CVPR*, 31663173 (2013).

8.1 Biographies

Cyril Meurie received a Ph.D in Computer Science from the University of Caen in France in 2005. Since 2012, he is a permanent researcher at the French institute of science and technology for transport, development and network (IFSTTAR) in France. His research is focused on graph-based signal processing, colour/texure segmentation and computer vision for ITS applications : people detection and re-identification; objects detection and classification; sensor data fusion for mobile localization.

Olivier Lézoray received a M.Sc., a Ph.D, and an habilitation thesis in Computer Science (CS) from the University of Caen (UNICAEN), France, in 1995, 2000, and 2007. He is a full professor in the Multimedia and Internet of the Cherbourg Institute of Technology. His research interest are in graph-based signal processing, multidimensional mathematical morphology, and machine learning. He is a Senior member of the IEEE, and member of the IAPR and EURASIP.

Christophe Coniglio received M.Sc. degree in computer science in 2013 from the University of Technology of Belfort-Montbéliard. From 2013 to 2016, he was a Ph.D. student on Image Processing at the French institute of science and technology for transport development and networks (IFSTTAR). He worked on detection and re-identification of peoples in camera networks. In 2016, he created his company to set up a new value proposal on vision and especially on object recognition.

Marion Berbineau received the Engineer degree in electronics, automatic and metrology from Polytech'Lille and the PhD in electronics from the University of Lille respectively in 1986 and 1989. She is currently Research Director at IFSTTAR. His research interest are EM propagation and modeling, radio channel characterization and modeling for transport and complex environments; signal processing for wireless communication and localization systems in multipath environments, MIMO systems and Cognitive Radio for ITS and railway applications.

List of Figures