Combining Keypoint Clustering and Neural Background Subtraction for Real-time Moving Object Detection by PTZ Cameras

Danilo Avola[1], Marco Bernardi[2], Luigi Cinque[2], Gian Luca Foresti[1] and Cristiano Massaroni[2]
[1]*Department of Mathematics, Computer Science, and Physics, Udine University, Via delle Scienze 208, 33100 Udine, Italy*
[2]*Department of Computer Science, Sapienza University, Via Salaria 113, 00198 Rome, Italy*

Keywords: Foreground Detection, Keypoint Clustering, Neural Background Subtraction, Moving Objects, PTZ Cameras.

Abstract: Detection of moving objects is a topic of great interest in computer vision. This task represents a prerequisite for more complex duties, such as classification and re-identification. One of the main challenges regards the management of dynamic factors, with particular reference to bootstrapping and illumination change issues. The recent widespread of PTZ cameras has made these issues even more complex in terms of performance due to their composite movements (i.e., pan, tilt, and zoom). This paper proposes a combined keypoint clustering and neural background subtraction method for real-time moving object detection in video sequences acquired by PTZ cameras. Initially, the method performs a spatio-temporal tracking of the sets of moving keypoints to recognize the foreground areas and to establish the background. Subsequently, it adopts a neural background subtraction to accomplish a foreground detection, in these areas, able to manage bootstrapping and gradual illumination changes. Experimental results on two well-known public datasets and comparisons with different key works of the current state-of-the-art demonstrate the remarkable results of the proposed method.

1 INTRODUCTION

Smart systems to automatically perform monitoring tasks (e.g., person re-identification) is playing a more and more important role. A main duty of these tasks is the moving object detection, since it allows the segmentation of the foreground moving elements, thus facilitating the reconstruction of the background of a video sequence. Anyway, the moving object detection presents a wide range of challenges, which are well reported in (Shaikh et al., 2014). These challenges mainly regard the management of the following dynamic factors of the scene:

- bootstrapping: construction of the background model is a complex task, especially when the initial frames of a video contain moving objects;
- illumination changes: gradual illumination changes can affect the moving object detection, especially in outdoor environments where the natural light changes over time;
- camouflage: foreground moving elements must be segmented from the scene, even if they have chromatic features similar to those of the background;
- shadows: shadows of the moving objects must be considered in the construction of the background.

In the last decades, several solutions have been proposed to face these issues (Bouwmans, 2014; Sobral and Vacavant, 2014) and, recently, great attention has been given to moving object detection algorithms based on artificial neural networks (ANNs) (Maddalena and Petrosino, 2008; Maddalena and Petrosino, 2014). ANNs present different advantages. In particular, their ability in adapting and learning new situations has played a key role in using these algorithms instead of the traditional approaches in video surveillance. In addition, these algorithms are proving to be very suitable for the management of those dynamic aspects that are the focus of the present paper, i.e., bootstrapping and illumination changes.

1.1 Foreground Detection by using PTZ Cameras

Foreground detection by using PTZ cameras has a rich literature. Initially, the frame-to-frame methods were the first presented approaches. They consist in identifying the overlapping regions between two consecutive frames and in analysing the pixel information inside them. In (Kang et al., 2003), an adaptive background model is generated by consecutive frames, and subsequently aligned by using of a geo-

Avola, D., Bernardi, M., Cinque, L., Foresti, G. and Massaroni, C.
Combining Keypoint Clustering and Neural Background Subtraction for Real-time Moving Object Detection by PTZ Cameras.
DOI: 10.5220/0006722506380645
In *Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2018)*, pages 638-645
ISBN: 978-989-758-276-9

metric transform. The work proposed in (Zhou et al., 2013), instead, utilized a motion segmentation algorithm, based on the methods reported in (Brox and Malik, 2010; Ochs and T.Brox, 2011), to analyse the point trajectories, to segment them into clusters, and to turn these clusters into dense regions. Another solution is shown in (Avola et al., 2017b), where a spatio-temporal tracking of sets of keypoints is used to distinguish the background from the foreground.

Subsequently, different attempts in improving the existing methods led the developers in exploring the frame-to-global approaches. An interesting work is presented in (Xue et al., 2011), where the authors proposed a panoramic Gaussian mixture model to cover the camera field of view and to register each new frame using a multi-layered correspondence ensemble. Probabilistic methods were also used, as reported by (Kwak et al., 2011; Elqursh and Elgammal, 2012), which proposed solutions based on Bayesian filters.

Different approaches are those based on machine learning techniques. In the work presented by (Ferone and Maddalena, 2014), moving object detection is performed by an original extension of a neural-based background subtraction approach. Lastly, in (Rafique et al., 2014), an algorithm based on Restricted Boltzman Machine (RBM) to learn and to generate the background model is reported.

1.2 Main Contribution

Unlike the existing works, this paper presents an algorithm for real-time moving object detection (in video sequences acquired by PTZ cameras) based on the combination of keypoint clustering and neural background subtraction. The spatio-temporal tracking of the keypoints is used to estimate the camera movements and the scale variations (Avola et al., 2017b). In particular, this step is utilized to identify the candidate areas of foreground and to manage the bootstrapping problem. Subsequently, on these areas, an ANN implemented by using self-organizing feature maps (SOFMs) (Kohonen, 1982) is used to perform a neural background subtraction and to handle the gradual illumination changes. The main contribution of the paper can be summarized as follows:

1. a robust bootstrapping management, in real-time, by using an original keypoint clustering strategy;
2. a variation of the neural background subtraction method proposed in (Maddalena and Petrosino, 2014) through which also the video sequences acquired by PTZ cameras (and not only by static cameras) can be managed;
3. an adaptive use of the neural background subtraction method proposed in (Maddalena and Petrosino, 2014; Ferone and Maddalena, 2014) through which only candidate areas of the image (and not the whole image) can be analysed, thus reducing both computational time and noise.

The rest of the paper is structured as follows. Section 2 describes the proposed method. Section 3 presents two experimental sessions. In the first, the Hopkins 155 dataset (Tron and Vidal., 2007) is used to compare the accuracy of the proposed method with key works of the current state-of-the-art. In the second, the Airport MotionSeg dataset (Dragon et al., 2013) is used to evaluate the performance of the proposed method during zoom operations. Finally, Section 4 concludes the paper.

2 LOGICAL ARCHITECTURE

As shown in Figure 1, the system architecture is divided in different modules. The first module is the *Background Module Initialization*, where a model of the background composed by both a neural map and a set of keypoints, and linked descriptors, is created. The keypoints and descriptors are contained in two sets called K_{b_t} and D_{b_t}, respectively, and where t is a time instant. The self-organizing neural network created in this step is organized as a 3D matrix, denoted with β_t. In the first iteration, the sets K_{b_0} and D_{b_0} are extracted from f_0, i.e., the first frame acquired by the PTZ camera. The frames after f_0 are provided as input to the *Feature Matching* module, which finds the correspondences between $K_{b_{t-1}}$ and K_t, where K_t is the set of keypoints extracted from the frame f_t. The correspondences are stored in a collection containing all the matches, called Φ_t. By using Φ_t, the changes in the scene, due to the PTZ camera movements, are estimated in the *Camera Movements & Scale Changes* module. This module also analyses the displacement of the keypoints inside the scene and identifies the set of candidate keypoints that belongs to the foreground, called K_{F_t}. In *Keypoint Clustering & Foreground Area Detection* module, the keypoints in K_{F_t} are grouped in clusters that indicate the areas of the foreground elements. These areas of pixel, called A_t, are used to perform the foreground segmentation in the *Neural Background Subtraction* module. Moving objects are represented by a set of blobs, called $Mask_{F_t}$. At each iteration, all components of the background model are updated by the *Background Model Updating* module. The updating of the weight vectors in β_t is performed according to the position of the pixels. If a pixel belongs to the foreground, its weight vector is not updated. This last phase allows to obtain a robust model.

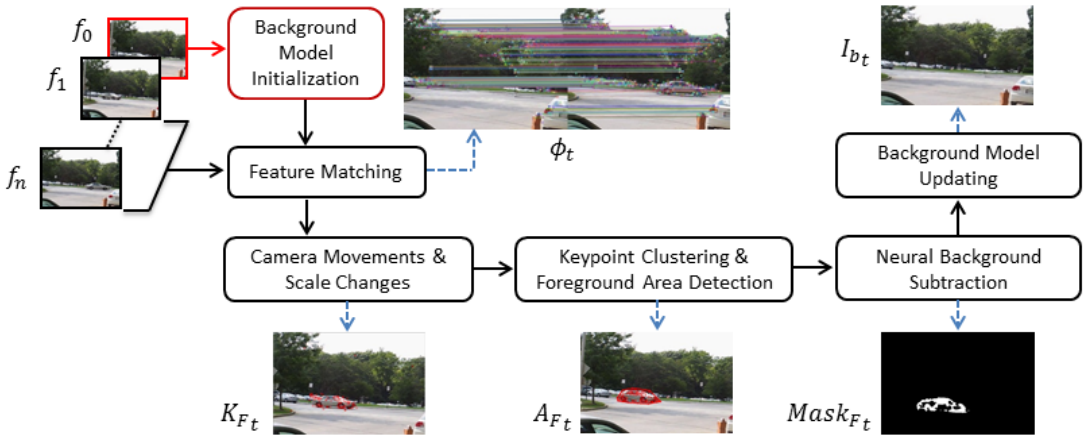


Figure 1: Logical architecture of the proposed system.

2.1 Background Model Initialization

Given a time instant t, the background model at the previous iteration is composed by an RGB image, called $I_{b_{t-1}}$, the set of keypoints $K_{b_{t-1}}$, the set of descriptors $D_{b_{t-1}}$, and a self-organising neural network, called β_{t-1}. The $I_{b_{t-1}}$ image represents an approximation, at time instant $t-1$, of the scene containing only the background elements (when $t=0$, I_{b_0} is equal to f_0). Notice that, f_0 can also contain foreground elements, in fact the use of the keypoint clustering (as shown in Section 2.4) can manage the bootstrapping problem even if the initial scene is populated by moving elements. In the proposed solution, β_t is represented by a 3D matrix with N rows, P columns, and n layers, whose number, i.e., $n=6$, was chosen according to empiric tests. Each layer L_i, with $1 \leqslant i \leqslant n$, contains, for each pixel $x \in I_{b_t}$, $L \times N$ weight vectors, called $m_{L_i}(x)$. The whole set of layers, L_i, composes the map β_t. At time instant $t=0$, the weights of the vectors $m_i(x)$ in β_0 are initialized by using the pixel brightness values of f_0. Also the sets K_b and D_b are extracted from f_0 by using the feature extractor A-KAZE (Alcantarilla et al., 2013), which is able to compute and to describe the visual features with faster performance with respect to other popular feature extractors, such as: SURF (Bay et al., 2008) or ORB (Rublee et al., 2011).

2.2 Feature Matching

In this phase, inspired by the work proposed in (Avola et al., 2017a), the features of the background model are compared with those extracted from the current frame. A set of keypoints, K_t, with their descriptors, D_t, are extracted from f_t. In this step, the features of the background model are separated to those that do not belong to it. The K-Nearest Neighbours (KNN) approach (Bishop, 2011) was chosen to perform the match between the descriptors in $K_{b_{t-1}}$ and those in K_t. Considering that A-KAZE extracts descriptors composed of binary values, the Hamming distance was used. Like the work proposed in (Avola et al., 2017b), for each $k \in K_{b_{t-1}}$, the two best matches between k and K_t were found by using a KNN with $K=2$. Subsequently, the ratio between these two matches is computed as follows:

$$ratio = \frac{hDist(k,k_1')}{hDist(k,k_2')} \quad (1)$$

where, $k', k'' \in K_t$ are the keypoints that have a match with k. The Hamming distances from k are expressed by $hDist(k,k_1')$ and $hDist(k,k_2')$, respectively. The ratio in $[0,1]$ expresses the proximity of Hamming distances between two different matches. If the ratio value is high, the two distances are close. When the ratio is over a threshold r (where r defines the maximum closeness between the two distances), all matches of k are discarded. A low value of r implies a low presence of undesired matches. Based of several empirical tests, we fixed the value of r to 0.60. All valid best matches are inserted into the Φ_t set, while all the keypoints, in K_t, without a valid match with $K_{b_{t-1}}$ are inserted in a set called K_{diff}.

Unlike the work presented in (Avola et al., 2017b), the proposed method performs a comparison between the sets K_{diff} and $K_{F_{t-1}}$, in addition to the only comparison between $K_{b_{t-1}}$ and K_t. Notice that, this task can be performed only when a foreground element is identified in the frame f_{t-1}. The set of these matches are called Φ_d. All keypoints in K_{diff} with a match in Φ_{d_t} are inserted in the K_{F_t} set, the latter represents the collection of the candidate foreground keypoints inside the frame f_t. This last step has been added to

also identify the foreground elements that do not perform movements in a current frame.

2.3 Camera Movement and Scale Change Estimation

The foreground keypoints, the camera movements, and the scale changes are estimated by using the matches inside Φ_t. The first step is to distinguish the background keypoints from the moving object keypoints, by using a 3x3 homography matrix, called H. This matrix describes the relation between two consecutive frames and maps the coordinates of a point x_1 in f_{t-1} into the coordinates of a point x_2 in f_t:

$$x_2 = Hx_1 \tag{2}$$

In our case, H is used to map the coordinates of the keypoints inside $I_{b_{t-1}}$ in the coordinates of the keypoints inside f_t. This task is performed by the RANdom SAmple Consensus (RANSAC) algorithm (Fischler and Bolles, 1981) by using the matches contained in Φ_t. The initial estimated homography, H, is refined by using the Levenberg-Marquardt optimization (Marquardt and Donald, 1963) that minimizes the re-projection error. For each match $(k,k^{'}) \in \Phi_t$, the following condition must be verified:

$$z = \begin{cases} 1 & \text{if } \sqrt{(k-k^{'})^2} - \sqrt{(k-k_h)^2} \geq \tau_1 \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

where, $k_h = Hk$ is the estimated position of k in f_t, and τ_1 is a tolerance on the difference between the estimated distance by homography and the estimated distance by Φ_t. If $z = 0$, the keypoint $k^{'}$ is a background keypoint and it is inserted in K_{b_t}. When τ_1 has a low value, a large number of keypoints results static and fitted in the background, on the other hand, in this case a large number of false positives inside K_{F_t} can occur. Instead, if τ_1 has a high value, the estimation of the keypoint movements is less restrictive, but a large number of false negatives can occur. Based on empirical tests, the value of τ_1 has been fixed to 2.0. On the contrary, when $k^{'}$ is a foreground keypoint, it is inserted in K_{F_t}. When all the candidate keypoints in K_{b_t} are found, their matches in Φ_t are used to compute the affine transformation matrix between $I_{b_{t-1}}$ and f_t, thus computing the information required to estimate the movements performed by the PTZ camera. Given three pairs of matches (k_a,k_b), (k_c,k_d), and (k_e,k_f) $\in \Phi_t$ with k_b, k_d, and $k_f \in K_t$, the affine transformation matrix, A, can be calculated as follows:

$$A = \begin{bmatrix} \lambda_x & 0 & t_x \\ 0 & \lambda_y & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x_{k_b} & x_{k_d} & x_{k_f} \\ y_{k_b} & y_{k_d} & y_{k_f} \end{bmatrix} \begin{bmatrix} x_{k_a} & x_{k_c} & x_{k_e} \\ y_{k_a} & y_{k_c} & y_{k_e} \\ 1 & 1 & 1 \end{bmatrix}^{-1} \tag{4}$$

where, t_x and t_y are the translations on the x and y axes, respectively. While, λ_x and λ_y are the variation scale on the same axes. Notice that, when a zoom is performed, we obtain that $\lambda_x = \lambda_y$.

Subsequently, the weights in β_{t-1} must be aligned according to the new spatial position of its correlated pixel. By using t_x, t_y and λ, the common portion of the scene between f_{t-1} and f_t is estimated. This area is expressed by a bounding box R_{t-1} in f_{t-1} and a bounding box R_t in f_t. If $t_x \neq 0$ or $t_y \neq 0$, the weights of the pixels in R_{t-1} must be moved to the new region R_t, for each layer $L_i \in \beta_{t-1}$. This alignment generates a new self-organising map, called $\beta_t^{'}$. If $\lambda \neq 1$, a zoom operation occurs and an interpolation to update the layers of β_{t-1} is applied. For each layer $L_i \in \beta_{t-1}$, the weights of the pixels inside R_{t-1} are interpolated in R_t of $L_i^{'} \in \beta'_t$. The pan, tilt, and zoom-out operations generate new pixels in f_t, which require an initialization of their weight vectors in $\beta_t^{'}$. For each $p \notin R_t$ is required an initialization, then $\forall L_i^{'} \in \beta_t^{'}$, $L_i^{'}(p) = f_t(p)$ is obtained. When $t_x = t_y = 0$ and $\lambda = 1$, the alignment of the weights in β_{t-1} is not necessary.

2.4 Keypoint Clustering and Foreground Detection

The areas that include the foreground elements in f_t are obtained by using a clustering algorithm applied on the keypoints contained in K_{F_t}. The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm (Ester et al., 1996) is chosen for the reason that it does not require to specify a-priory the number of clusters, moreover it is suitable to manage the presence of noise. The DBSCAN algorithm requires two parameters. The first, called τ_2, is the radius used to search the neighbouring keypoints. The second, called $MinPts$, is the minimum number of neighbouring keypoints that are required to form a single cluster. With a low value of τ_2, small clusters are obtained, but a certain amount of information can be lost. On the contrary, with a high value of τ_2, large clusters are created, but a high level of noise can be included. The result of this step consists of a set of clusters, $C_t = (c_1, c_2, ..., c_m)$. Each $c_i \in C_t$ is a portion of f_t and is associated to a set of pixels, called α_c. The entire collection of these areas α_c, called A_t, indicates all the regions that contain foreground elements in f_t.

2.5 Neural Background Subtraction

In this stage, the pixels inside the areas in A_t are analyses to find foreground elements. The background subtraction process works as follows:

- For each pixel $g \notin \alpha$, and $\forall \alpha \in A_t$, the value of $Mask_{F_t}(g)$ (i.e., the value of the pixel g inside the mask) is set to 0.
- For each pixel $p \in \alpha$, and $\forall \alpha \in A_t$:

$$Mask_{F_t}(p) = \begin{cases} 1 & \text{if } \frac{|\Omega_p|}{|H_p|} \leq 0.5; \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

where, $H_p = p^{'} : |p - p^{'}| \leq h$, is a 2D spatial neighbourhood of p (including p) with a radius of length h (in the proposed method $h = 2$). Instead, Ω_p is the set of pixels in H_p that satisfies the following condition:

$$\Omega_p = \{p^{'} \in H_p : d(\beta_t^{'}(p^{'}), f_t(p^{'})) \leq \varepsilon\} \quad (6)$$

where, $d(\beta_t^{'}(p^{'}), f_t(p^{'}))$ is the best match distance between the value of pixel $p^{'}$ in f_t and the weights of pixel $p^{'}$ inside $\beta_t^{'}$. More details on this distance are reported in Section 2.5.1. The threshold ε indicates the maximum distance that a pixel $p^{'}$ can have with its best match. If the best match distance is greater than ε, then $p^{'}$ is defined as a foreground pixel. Notice that, high values of the threshold allow to successfully process background pixels with significant changes (in terms of speed). Anyway, these values can made the method short-sighted in capturing slow changes of the foreground scene. The Eq. 5 indicates a measure of how many pixels in the neighbourhood of p have correspondence with the model. The pixel p can be considered a background pixel only if its value is less than 0.5 (i.e, less than half of the pixels in its neighbourhood does not belong to the foreground). In general, the value of 0.5 can be considered a balanced estimation for background and foreground elements.

2.5.1 Find Best Match

The best match of a pixel p with its model in $\beta_t^{'}$ is computed as follows:

$$d(\beta_t^{'}(p), f_t(p)) = \min_{i=1,\ldots,n} d(m_{L_i^{'}}(p), f_t(p)) \quad (7)$$

where, $d(\bullet, \bullet)$ is a metric based on the used color space to construct the self-organising map. In the proposed method, the Euclidean distance between the value of pixel and its weight vector of the layer L_i using the HSV colour model, is used. The metric can be expressed as follows:

$$d(L_i^{'}(p), f_t(p)) = ||(v_1 s_1 \cos(h_1), v_1 s_1 \sin(h_1), v_1) - (v_2 s_2 \cos(h_2), v_2 s_2 \sin(h_2), v_2)||_2^2 \quad (8)$$

where, v_1, s_1 and h_1 are the value, the saturation, and the hue of the pixel p, respectively. Instead, v_2, s_2 and h_2 are the same values of the weights in $m_{L_i^{'}}(p)$.

2.5.2 Cast Shadow

With the aim to manage complex scenarios, the proposed method inherits, from the current literature, a remarkable approach to make better the background subtraction stage, thus improving the whole algorithm. Moving objects can generate shadows, which require to be managed and included in the background model. Based on the work reported in (Cucchiara et al., 2003), the proposed method performs a detection of the cast shadow pixels, which are considered as background and updated by the Eq. 9.

2.6 Background Model Updating

The reinforcement of the self-organising map is a necessary step to recognize moving objects and to distinguish them from the changes of the background. During this task, the updating of a pixel weight influences all the weights belonging to the same layer. So a new updated self-organising map, called β_t, for the frame f_t is obtained. Moreover, a new set of internal layers of β_t, called L_i^*, for $1 \leq i \leq n$, is computed. The update relation is defined as follows:

$$m_{L_i^*}(p) = (1 - \varphi(p, p^{'}))m_{L_i^{'}}(p) + \varphi(p, p^{'}) f_t(p) \;\; \forall p^{'} \in N_p \quad (9)$$

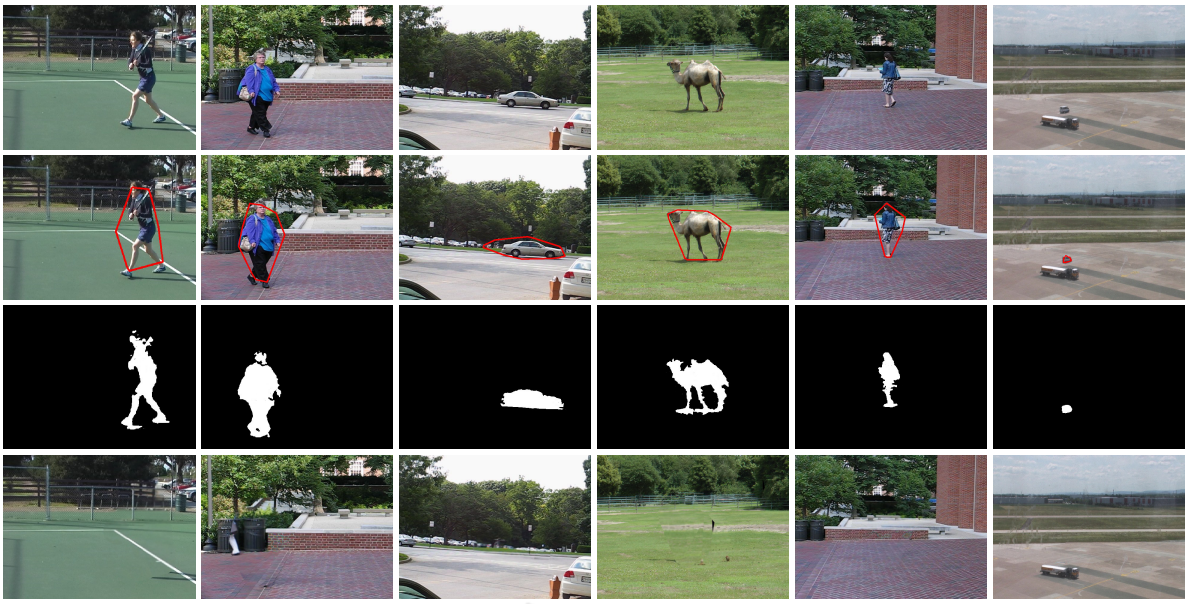
where, $p \in \alpha$ ($\alpha \in A_t$) and $N_p = \{p^{'} \in f_t : |p - p^{'}| < w_{2D}\}$ is a 2D squared spatial neighbourhood of p that includes p. The radius of N_p is expressed by w_{2D} (here fixed as: $w_{2D} = 1$). Actually, $\varphi(p, p^{'})$ is defined as follows:

$$\varphi(p, p^{'}) = \gamma g(p - p^{'})(1 - Mask_{F_t}) \quad (10)$$

where, $g(p - p^{'})$ is a 2D Gaussian low-pass filter (Burt, 1981) and γ is the learning rate. A large value of γ produces a fast learning step of the self-organising map with respect to the changes of the scene. In this work, after different empirical tests, γ was set to 0.05. On the contrary, a small value of the parameter reduces the false negatives since the map learns less rapidly. For each pixel outside the areas in the A_t set, only the weight of every layer L_i^* in β_t is processed as follows:

$$m_{L_i^*}(p) = \begin{cases} f_t(p) & \text{if } i = n; \\ m_{L_{i+1}^{'}}(p) & \text{otherwise.} \end{cases} \quad (11)$$

In this way, the most recent frame information, in the n^{th} layer, is stored, while the older information contained in the 0^{th} layer is removed. The updating of the weights of these pixels is fundamental to manage the bootstrapping problem. Even if, in the background initialization phase, foreground elements are included, their location is identified by the clusters.



(a) (b) (c) (d) (e) (f)

Figure 2: Examples of moving object detection and background updating. The video sequences from the column (a) up to the column (e) belong to the Hopkins 155 dataset. The last sequence belongs to the Airport MotionSeg dataset. For each column (from the top to the bottom), the first picture is the generic frame, the second is the keypoint clustering stage, the third is the moving object mask, and the last is the background updating. From the first up to the last column the following video are shown: tennis, people2, cars6, camel01, people1, and bus.

This means that each pixel outside the clusters belongs to the background. The proposed method can estimate, since from the first frames, the moving objects without an initial estimation of the background. The latter can be considered a concrete overcoming of the current state-of-the-art.

3 EXPERIMENTAL RESULTS

This section reports the experimental results performed on the Hopkins 155 dataset (Tron and Vidal., 2007) and on the Airport MotionSeg dataset (Dragon et al., 2013). The first was used to perform a comparison, in terms of *Precision* $(Prec)$, *Recall* (Rec), and $F1-Measure$ $(F1)$, with selected key works of the current literature. The second was used to prove the effectiveness of the proposed method during zoom-in/zoom-out operations. Notice that, the latter task is rather unusual since the majority of works in this application field do not consider moving object detection along with zoom operations. Observe also that, in the proposed experiments, the values of ε and γ was set to 0.005 and 0.05, respectively, thanks to the preliminary empirical tests performed on both datasets. Conversely, the parameters τ_2 and *MinPts*, that depend of several factors, including the image resolution and the keypoint distribution, and whose values are reported in Table 1, were established on the basis of the observations derived by the OPTICS algorithm (Ankerst et al., 1999).

Table 1: Values of the τ_2 and *MinPts* parameters on the basis of the OPTICS algorithm. The first five videos (from the top) belong to the Hopkins 155 dataset, the last belongs to the Airport MotionSeg dataset.

Video	Resolution	τ_2	*MinPts*
Camel01	680x540	60	6
Cars6	640x480	70	5
People1	640x480	60	5
People2	640x480	60	5
Tennis	530x380	60	5
Bus	1440x1080	80	3

3.1 Experimental Evaluation

In Figure 2, visual representations of the results obtained by the proposed method during the different stages of the architecture are reported. The first row shows an example of source frame for each of the five challenging video sequences, the second row presents the related foreground keypoint clustering. The computed clusters allow the method to reduce the noise in the background subtraction stage, as depicted in the third row. The clusters also limit the computational

Table 2: Comparison with key works of the current literature on the basis of the average of the Precision, Recall, and F1-Measure metrics. The people1, people2, cars6, camel01, and tennis video sequences belong to the Hopkins 155 dataset. The last video sequence, bus, belongs to the Airport MotionSeg dataset.

	people1			people2			cars6			camel01			tennis			bus		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Proposed method	0.809	**0.954**	0.876	0.836	0.984	0.903	0.800	0.956	0.871	0.790	0.962	0.868	0.801	**0.991**	**0.895**	0.882	0.943	0.911
(Avola et al., 2017b)	0.765	0.917	0.840	N.A.	N.A.	N.A.	0.785	0.910	0.840	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A
(Kwak et al., 2011) - with NBP	0.950	0.930	**0.940**	0.850	0.760	0.828	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A
(Kwak et al., 2011) - without NBP	0.910	0.760	0.828	0.910	0.220	0.286	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A
(Elqursh and Elgammal, 2012) - 1	0.940	0.850	0.893	0.840	**0.990**	0.909	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	0.860	0.920	0.890	N.A.	N.A.	N.A
(Elqursh and Elgammal, 2012) - 2	**0.970**	0.880	0.923	0.850	0.970	0.906	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	**0.900**	0.810	0.850	N.A.	N.A.	N.A
(Ferone and Maddalena, 2014)	0.958	0.923	**0.940**	**0.931**	0.971	**0.950**	**0.866**	0.964	**0.913**	N.A.	N.A.	N.A.	N.A.	N.A.	N.A	N.A.	N.A.	N.A.
(Brox and Malik, 2010)	0.890	0.775	0.829	N.A.	N.A	N.A	0.824	**0.994**	0.901	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A
(Zhou et al., 2013)	0.936	0.933	0.934	0.925	0.965	0.945	0.837	0.984	0.905	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A
(Sheikh et al., 2009)	0.780	0.630	0.697	0.730	0.830	0.777	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	0.270	0.830	0.400	N.A.	N.A.	N.A

time required by the moving object detection stage considering that only some portions of the frame f_t are processed (and not the whole frame). The area analysed by the method is slightly greater than the area delimited by the clusters, this to be sure not to miss parts of the foreground elements. Typically, the method considers an area of the 20% bigger. We chosen this value by empirical evaluations, which shown, in the worst case, an amount of pixels outside the clusters of about the 18%. Finally, the last row reports the reconstruction of the background, I_{b_t}, for each video. The computational time required by the construction of the clusters does not influence the performance of the method since the number of keypoints that has to be analysed is significantly lower than the number of pixels that has to be processed without using the clustering based approach.

The correctness of the moving object detection algorithm was estimated by using three well-known metrics: *Precision* $(Prec)$, *Recall* (Rec), and $F1-$ *Measure* $(F1)$. The metrics were computed on the bases of the pixels correctly assigned to the background and foreground in relation to the given ground-truth, more specifically:

$$\text{Rec} = \frac{TP}{TP+FN} \tag{12}$$

$$\text{Prec} = \frac{TP}{TP+FP} \tag{13}$$

$$\text{F1} = \frac{2(Rec)(Prec)}{Rec+Prec} \tag{14}$$

where, TP, FP, FN, and TN are the number of true positive, false positive, false negative, and true negative, respectively, in terms of number of pixels inside and outside of the related portion of the image (i.e., background or foreground). In Table 2 the results of the proposed solution are shown. We have chosen to perform the experiments on those video sequences since they, almost all, are directly comparable with selected key works of the current state-of-the-art. The results show that, as regards the recall values, the proposed method achieves good performance, especially in the people1 and tennis video sequences, where it reaches the best results. In people2 and car6 video sequences, even if the method does not obtain the higher values, the recall measure is very close to the best works in the literature. Moreover, in the tennis video sequence the obtained results, as regards the recall and F1 metrics, exceed the exterminated key works. Notice that, high recall values obtained by the proposed method highlight that it is able to capture almost all the pixels that compose the foreground objects. This last factor is very important for the application of additional processing, such as object classification and people re-identification.

No results were found in other works for the camel01 video, anyway we have tested it because, in our opinion, it is a very interesting sequence. In addition, we have observed that several works do not treat zoom-in and zoom-out operations, make the comparison a very hard task. We have adopted a very challenging video sequence of the Airport MotionSeg dataset, i.e., bus, to test the proposed method during these operations. The obtained results have been extremely satisfying with all the computed metrics. The bus video sequence does not provide the ground-truth of the foreground. To solve this gap, the likely foreground pixels were computed by a semi-automatic segmentation process. Summarizing, the method has shown to work properly with different challenging video sequences. The obtained results have pointed out that the proposed strategy is highly practical and consistent. Finally, the use of the keypoints allows the method to be used in real-time application fields.

4 CONCLUSIONS

This paper presents a combined keypoint clustering and neural background subtraction method for real-time moving object detection in video sequences acquired by PTZ cameras. The experimental results performed on two well-known public datasets demonstrate the effectiveness of the proposed approach compared with selected key works of the current state-of-

the-art. The reported solution shows different contributions with respect to the current literature, including the management of the bootstrapping and illumination change issues, the real-time processing, an original keypoint clustering strategy, and a novel pipeline based on the neural background subtraction.

REFERENCES

Alcantarilla, P., Nuevo, J., and Bartoli, A. (2013). Fast explicit diffusion for accelerated features in nonlinear scale spaces. pages 1–12. British Machine Vision Conference (BMVC).

Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. (1999). Optics: Ordering points to identify the clustering structure. *SIGMOD Record*, 28(2):49–60.

Avola, D., Bernardi, M., Cinque, L., Foresti, G. L., and Massaroni, C. (2017a). Adaptive bootstrapping management by keypoint clustering for background initialization. *Pattern Recognition Letters*, 100:110–116.

Avola, D., Cinque, L., Foresti, G. L., Massaroni, C., and Pannone, D. (2017b). A keypoint-based method for background modeling and foreground detection using a ptz camera. *Pattern Recognition Letters*, 96:96–105.

Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. (2008). Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359.

Bishop, C. M. (2011). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer.

Bouwmans, T. (2014). Traditional and recent approaches in background modeling for foreground detection: An overview. *Computer Science Review*, 11-12:31–66.

Brox, T. and Malik, J. (2010). Object segmentation by long term analysis of point trajectories. pages 282–295. European Conference on Computer Vision (ECCV).

Burt, P. J. (1981). Fast filter transform for image processing. *Computer Graphics and Image Processing*, 16(1):20–51.

Cucchiara, R., Grana, C., Piccardi, M., and Prati, A. (2003). Detecting moving objects, ghosts and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1337–1342.

Dragon, R., Ostermann, J., and Gool, L. V. (2013). Robust realtime motion-split-and-merge for motion segmentation. pages 59–69. German Conference on Pattern Recognition (GCPR).

Elqursh, A. and Elgammal, A. (2012). Online moving camera background subtraction. pages 228–241. European Conference on Computer Vision (ECCV).

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., Simoudis, E., Han, J., and Fayyad, U. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD).

Ferone, A. and Maddalena, L. (2014). Neural background subtraction for pan-tilt-zoom cameras. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(5):571–579.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Kang, S., Paik, J.-K., Koschan, A., Abidi, B. R., and Abidi, M. A. (2003). Real-time video tracking using ptz cameras. volume 5132, pages 103–111. International Conference on Quality Control by Artificial Vision.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69.

Kwak, S., Lim, T., Nam, W., Han, B., and Han, J. H. (2011). Generalized background subtraction based on hybrid inference by belief propagation and bayesian filtering. pages 2174–2181. IEEE International Conference on Computer Vision (ICCV).

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

Maddalena, L. and Petrosino, A. (2014). The 3dsobs+ algorithm for moving object detection. *Computer Vision and Image Understanding*, 122:65–73.

Marquardt and Donald (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441.

Ochs, P. and T.Brox (2011). Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. pages 1583–1590. IEEE International Conference on Computer Vision (ICCV).

Rafique, A., Sheri, A. M., and Jeon, M. (2014). Background scene modeling for ptz cameras using rbm. pages 165–169. International Conference on Control, Automation and Information Sciences (ICCAIS).

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: An efficient alternative to sift or surf. pages 2564–2571. IEEE International Conference on Computer Vision (ICCV).

Shaikh, S. H., Saeed, K., and Chaki, N. (2014). *Moving Object Detection Using Background Subtraction*. SpringerBriefs in Computer Science. Springer International Publishing.

Sheikh, Y., Javed, O., and Kanade, T. (2009). Background subtraction for freely moving cameras. pages 1219–1225. IEEE International Conference on Computer Vision (ICCV).

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

Tron, R. and Vidal., R. (2007). A benchmark for the comparison of 3-d motion segmentation algorithms. pages 1–8. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR).

Xue, K., Ogunmakin, G., Liu, Y., Vela, P. A., and Wang, Y. (2011). Ptz camera-based adaptive panoramic and multi-layered background model. pages 2949–2952. IEEE International Conference on Image Processing (ICIP).

Zhou, X., Yang, C., and Yu, W. (2013). Moving object detection by detecting contiguous outliers in the low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):597–610.