Background subtraction using Artificial Immune Recognition System and Single Gaussian (AIRS-SG)

Wafa Nebili[1] · Brahim Farou[1] · Hamid Seridi[1]

Abstract

Background subtraction is an essential step in the video monitoring process. Several models have been proposed to differentiate background pixels from foreground pixels. However, most of these methods fail to distinguish them in highly dynamic environments. In this paper, we propose a new method robust and more efficient for distinguishing moving objects from static objects in dynamic scenes. For this purpose, we propose to use a bio-inspired approach based on the Artificial Immune Recognition System (AIRS) as a classification tool. AIRS separates antibodies, represented by the pixels of the background model, from the antigens that model foreground pixels representing moving objects. Each pixel is modeled by a feature vector containing the attributes of a Gaussian. Only the pixels classified as background are taken into account by the system and updated in the model. This combination has allowed to benefit from two advantages: the power of AIRS to provide an online update of system parameters and the ability of Gaussians to adapt to scene variations at the pixel level. To test the proposed approach, six videos representing the dynamic background category of the CDnet 2014 dataset are selected. Obtained results proved the effectiveness of this new process in terms of quality and complexity compared to other state-of-the-art methods.

Keywords Video surveillance · SG · Background subtraction · Moving objects · Foreground pixel · AIRS

1 Introduction

Background subtraction (BS) also called motion detection or foreground detection, is a crucial step in many computer vision application like: video surveillance [7, 35], multimedia

✉ Wafa Nebili
  nebili.wafa@univ-guelma.dz

  Brahim Farou
  farou.brahim@univ-guelma.dz

  Hamid Seridi
  seridi.hamid@univ-guelma.dz

[1] LabSTIC, 8 mai 1945 Guelma University, POB 401, 24000 Guelma, Algeria

[11] and optical motion capture [5], etc. BS consists to model the background before detecting the moving objects (foreground). Generally, the moving objects are a humans, cars, texts, etc. The intuitive way to model, the background is to train the system using a set of frames devoid of moving objects. After that, we applied an on-line or off-line process to extract the foreground from frames. In the on-line process, the background model updated during the whole execution to pick up any new changes in the background within the video sequence, but in the off-line process, the background model is unchanged. An efficient method for detection moving objects must ensure a good separation between the background and the foreground, with gain in execution time and memory space. Single Gaussian (SG) is among the most popular methods that have achieved great success in the detection of moving objects, since it is simple, very fast and inexpensive in the calculation. However, this method is sensitive to the fast variations of pixel. For example, when the background is dynamic, SG cannot memorize all states of pixel. Several research works are proposed to improve SG results quality, among these methods: [10, 17, 39], etc.

Recently, with the development of the technology, there exist many bio-inspired techniques and intelligent algorithms for scientific and engineering computing. One of these techniques, Artificial Neural Network (ANN), which is used to mathematically model the intelligence of the human brain [1]. AIRS also is a bio-inspired technique proposed by [56] and it describes the recognition of self cells from strange cells in the human body. Another algorithm proposed by Yang and Xin-She [59] consists to model the flight of bats during their search on the food source. A novel extension of this algorithm has been proposed in the work of Yong et al. [61]. In the context of the optimization, Ishibuchi et al. [19] proposed an optimization algorithm with an objective function that simultaneously solves four or more contradictory objectives, this algorithm named Many-objective optimization problems (MaOPs). An enhancing of MaOPs is proposed by Li et al. to improve the selected features and the accuracy [28]. Moreover, there are many other bio-inspired algorithms like: IBEA-SVM [29], PSO [8, 18, 23, 60], etc.

In this paper, we propose a new approach for background subtraction in dynamic scenes. The proposed system performs a combination of AIRS and SG for background modeling. The choice of the AIRS classifier among the other classifiers is determined by three main factors. The first one is the concordance of the model with the treated problem, knowing that the latter allows to separate all that is body (in our case the background) with all that is foreign (in our case moving objects). The second factor is related to the dynamic nature of the videos processed. Indeed, most of the methods cited in the state of the art do not make it possible to adapt easily with frequent changes of the backgrounds and which in most cases require several frames. The AIRS classifier can easily cope with this type of problem, thanks to the cloning operations, which allows to create several valid backgrounds (background model) with dynamic management of the number of models using only the current frame which considerably increases the success rate. The ability to update the system when it is operational is another factor that gives the AIRS classifier an advantage over other classifiers. Indeed, the online update allows the systems to familiarize with the possible changes in the background without going through a re-initialization step which requires a system shutdown during a long learning period (for example: growing tree, changing the color of a wall, etc.).

We have made some improvements in the affinity measurement, the competition for resources and development of a candidate memory cells process and in the principle of memory cell introduction so that it can meet the requirements of the background subtraction process.

The reminder of the paper is organized as follows: Section 2 presents a state of the art on changes detection in dynamic scenes. Sections 3 and 4 explain respectively the basic SG and AIRS methods. Section 5 focuses on the description of the proposed approach. Experiments are discussed in Section 6, and we end with a conclusion and some perspectives in Section 7.

2 Related work

Background subtraction in dynamic scenes is a difficult problem that requires an effective method to ensure a good separation between the background and the foreground. Several studies have been proposed to improve the background subtraction and to reduce noisy due to misidentified pixels or the presence of haze in the scene [62, 64]. These studies can be divided into five groups (Fig. 1). One of them focuses on the selection and the combination of good feature (color, edge, texture, etc.), while the others try to develop methods that can deal with all the possible scenarios in order to separate the moving objects from the static ones.

2.1 Features selection based approaches

The selection of the best characteristics is an essential way that allows the least efficient classifier to well separate between classes. It is for this reason that several works have focused on the selection and combination of several characteristics in order to increase the discriminating power of classifiers. In color and texture context, St-Charles et al. [47] used a binary spatio-temporal features and color information to detect local variations at the pixel level. Authors in [53] proposed a new method (M^4CD) that exploits color, texture, and other heterogeneous features to separate foreground from background pixels. However, the extraction of color and texture features requires a lot of time which influences on the execution time. Edges based features are also used in background subtraction, Allebosch et al. [2] propose a local ternary pattern descriptor with RGB color information to identify foreground pixels. The problem with this proposal types lies in the fact
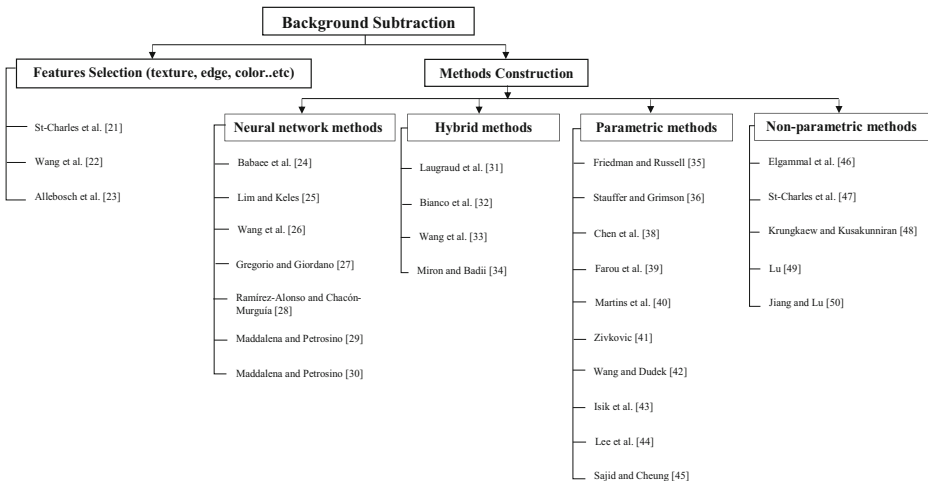

Fig. 1 Taxonomy of background subtraction methods

that all approaches require a robust edge detection method to achieve a good background subtraction.

2.2 Neural network based approaches

Neural network methods have been widly used for enhancing the performance of the background subtraction systems in dynamique scenes. Babaee et al. [3] proposed a background subtraction system with deep Convolutional Neural Network (CNN). Following the same perspectives, authors in [31] used a triplet convolutional neural networks to build a robust background model. Wang et al. [55] presented a semi-automatic CNN for motion detection which uses a few examples that manually describe the moving object for the training. When the training step is finished, a segmentation maps (foreground and background models) are produced to be used later in order to label the remain frames in the video sequence. Gregorio and Giordano [9] proposed a weightless neural network called WiSARDrp to model the background. Auto-Adaptive Parallel SOM Architecture (AAPSA) is a self-organized maps proposed by [42] for background subtraction. This method allows a parallel adaptation of rates to detect changes in video sequences. Authors in [36] presented a new method that creates a neural background model without any prior patterns. This model learns and adapts automatically with the scene variations. Furthermore, authors introduced the notion of spatial coherence in the background updating process (SC-SOBS) to provide a robust method to dealing with false detections. An extension of this work is proposed by Maddalena and Petrosino [37] in which they introduce the fuzzy concept in the update process of the background model.

Despite the high quality results achieved by the deep Neuronal Networks based methods, these latter needs an offline training mechanism which always requires a large number of ground truth examples to create a robust model which are, in the most cases, not available in real applications. In addition, this learning needs very powerful computers and takes a lot of time for training the system. However, the major problem remains the inability of this latter to adapt to the permanent changes of the background.

2.3 Hybrid methods

Recently, many approaches have focused on the combination of several algorithms to subtract the background. Laugraud et al. [25] combined information extracted from a semantic segmentation algorithm with information extracted from any background subtraction algorithm to identify moving objects. However, semantic segmentation is difficult and requires a very robust algorithm for the detection, localization and simultaneous segmentation of objects and semantic regions. Bianco et al. [4] also combined a set of algorithms for video change detection. They used the principle of genetic programming in: automatic selection of the best algorithms, the combination of their results and the application of the most appropriate post-processing operations on algorithm outputs. Nevertheless, the choice of the algorithms that constitute the initial population influences on the results quality. Furthermore, the evaluation of the fitness function is related to the used dataset and cannot be generalized under any case. In the same context Wang et al. [51] performed a hybridization between motion information, environmental change and object appearance to detect moving objects. In [40] authors proposed background subtraction system based on graph cut method that uses two algorithms: optical flow algorithm and GMM to identify moving objects. However, this system is not efficient to detect moving objects that park for a period of time.

2.4 Parametric based approaches

Gaussian Mixture model (GMM) is among the most commonly used algorithms to deal with motion detection in dynamic scenes. This method represents the history of each pixel separately with a mixture of Gaussian probability distribution. The first proposal of this model was made by Friedman and Russell [14]; however, the standard model with a well-defined equations was suggested by Stauffer and Grimson [48]. Since that, many works have tried to use GMM in solving the problems of background disturbance [22]. Chen et al. [6] provided a sharable GMM model in a well-defined region. The idea is that each pixel search for the most matched and optimal model from the foreground and background models of their neighborhood that can represent their state. This proposition reduced the noises caused by the local small movements. The major drawback of this technique lies in the size of the search region on shareable models. Indeed an efficient method for creating homogeneous regions is required. In the same context, Farou et al. [13] proposed a Gaussian mixture with background spotter, which assigns a spotter (agent) to each frame block to detect changes in the background. Many extensions have been proposed to improve the model adaptation speed and the dynamic number of Gaussians in GMM, among them: Martins et al. [38] presented a MOG with dynamic learning rate (Boosted MOG (BMOG)) and Zivkovic [66] added a recursive equations that automatically select the number of Gaussians needed for each pixel at each observed scene. However, all these improvements remain valid under very specific constraints and environments.

Wang and Dudek [52] proposed a new adaptive algorithm for modeling the background in video scenes. This method represents the history of each pixel with a only few templates (background value). The update process is performed at each period by removing the least useful background values of the background model. This method provides very good execution speed due to the low algorithmic complexity; however, the number of samples used to model the background does not represent all the pixel states, especially when the pixel variations are very fast. Authors in [20] proposed a new change detection method that used the sliding window technique with a feedback mechanism to control and update the parameters of the background model. This method allows to build a robust background model that can be adjusted dynamically with the scenes variations. However, the result quality depends on the size of the sliding window. In order to remove the false positives caused by the misclassified pixels, authors in [26] proposed to perform a matching between the foreground and the false positive pixels for background subtraction. The development of this method is very difficult since it requires to find the unique trend of the dynamic background and to model it mathematically in order to supervise the background regions. Authors in [43] proposed a universal background subtraction system based on the notion of a megapixel (MP). It consists to merge multiple background models in megapixels following a clustering algorithm to detect moving objects. The use of megapixels allows to reduce the number of isolated pixels. However, the results quality depends on the robustness of the clustering algorithm to create a homogeneous megapixels.

2.5 Non-parametric based approaches

In the same context, several studies have focused on the development of non-parametric background subtraction models. KDE (kernel density estimator) is one of these models that estimated the probability density function of the recent N values of each pixel with a kernel estimator. The model is adapted quickly to the scene variations which allows a very sensitive detection of moving objects without any parameters [12]. Authors

in [46] used an approach based on self-adjusted dictionary words to model the background. In this approach, the current state of each pixel is described with three values (visual word), which are estimated during the learning stage. After each pixel classification, internal parameters adjusted with a feedback mechanism to update regularly the background model. Krungkaew and Kusakunniran [24] addressed a new visual word dictionary method that uses the lab color space (light, color channels) for detecting moving objects in dynamic background. In [34] the author presented a multi-scale spatio-temporal background subtraction method that collects the history of the pixel samples and its neighbors at different scales to detect moving objects. This method requires a lot of time and a large memory space to transform pixel samples at different scales and to save it with its neighbors. The method proposed by [21] used the notion of weighted samples to build an effective background model. The model is initialized with a small number of samples that have a variable weight. During the execution, the system replaces the samples having low weight with other samples. However, an efficient method is required for initializing the weights. Furthermore, the sample size differs from scene to another. Indeed, when the variation of pixels is very fast a small sample increases the false detection.

3 Single Gaussian model method (SG)

Background modelling with a Single Gaussian (SG) was proposed by Wren et al. [58]. The model assumes that there is an independence between pixels and it represents the history of the last n pixel values with a probability density function (1).

$$P(P_t) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(P_t-u)^2}{2\sigma^2}} \tag{1}$$

This method creates a model for each pixel. The model is composed of the mean, variance and current pixel value. The mean describes the dominant color of the current pixel and the variance represents the viability of that pixel around the mean.

For each pixel of the current frame P_t, it is possible to determine whether it is a background or foreground pixel using (2). If (2) is verified the pixel belongs to the background pixel set, otherwise the pixel represents the foreground.

$$\frac{|P_t - u_t|}{\sigma_t} < 2.5 \tag{2}$$

To take into account the illumination changes of the scene in the model, the Gaussian parameters are updated according to (3) and (4).

$$u_t = (1 - \alpha)u_{t-1} + \alpha P_t \tag{3}$$

$$\sigma_t^2 = (1 - \alpha)\sigma_{t-1}^2 + \alpha(P_t - u_t)(P_t - u_t)^T \tag{4}$$

Where, the parameters u_t, σ_t denote respectively the mean and the variance of pixel P at time t.

The learning rate α determines the speed of adaptation.

4 Artificial Immune Recognition System (AIRS)

The natural immune system can determine which cells belong to it and which cells do not, and it can protect the human body from external objects. In 2001, the first supervised artificial immune system was proposed by [56]. The model is based on antigen-antibody representation, which measures a degree of "closeness" or similarity between training data (antigens) and cell B, also called affinity or stimulation degree. The algorithm takes an antigens as an input and produces a set of memory cells. Memory cells represent the model that can be used in the classification phase.

The AIRS process is described in four stages:

–   **Initialization** This phase can be considered as a data pre-processing stage, because all data set items are normalized to ensure that the affinity between any two-feature vectors of the dataset is always in the range of [0,1]. After normalization, the Affinity Threshold (AT) is calculated. AT represents the average affinity value of all training data items (see (5)).

$$AT = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} Affinity(ag_i, ag_j)}{\frac{n(n-1)}{2}} \tag{5}$$

   Where, n represents the number of antigens in training data and Affinity (ag_i, ag_j) is the euclidean distance between two pairs of antibodies or antigens. Affinity threshold will be used later in the memory cell replacement process. The final step in this stage consists to initialize the memory cells set MC and ARB populations with a set of antigens ag randomly selected from the training data AG. This step is optional, ARBs and MC set can be initialized with ∅. All the following stages are applied only on one antigen ag.

–   **Memory cell identification and ARB generation** This stage is composed of two mechanisms; the first mechanism consists of identifying the memory cell mc_{match} the most similar to the antigen ag and which has the same class of this latter. mc_{match} is calculated using (6).

$$mc_{match} = argmax_{mc \in MC_{ag.c}} Stimulation(ag, mc) \tag{6}$$

$$Stimulation(ag, mc) = 1 - Affinity(ag, mc) \tag{7}$$

   If $MC_{ag.c} = ∅$, then $mc_{match} \leftarrow ag$ and $MC_{ag.c} \leftarrow MC_{ag.c} \cup ag$. After identifying the memory cell mc_{match}, AIRS generates new ARBs clones using mc_{match}. It creates $NumClones$ new clones. Each clone represent a feature vector mc_{match} that is muted with an empirically fixed rate between 0 and 1. The number of clones is calculated with (8):

$$NumClones = Hmr \times Clonal\_rate \times Stimulation(ag, mc_{match}) \tag{8}$$

   Where, $Hyper\_mutation\_rate$ (Hmr) and $Clonal\_rate$ are two integer values selected empirically. The mutation function is defined in [57].

–   **Competition for resources and development of a candidate memory cell** All the new clones generated in the previous stage, mc_{match} and all ARBs remained from previous antigen reactions are added to the ARB set (AB). To generate better representatives, the method has a mechanism to organize the survival of individuals within the ARB populations. This mechanism involves a process of allocated resource based on the sharing of cumulative resources according to the class of the antigen.

   ab, which have the same class of the antigen, get the half of the cumulative resources. The remaining half of resources will be split on the ab that belong to other classes.

ARBs of each class have a maximum allocation, if the sum of the resources allocated by that class exceeds the maximum allocation, in this case, resources are removed from the least stimulated ab. If the resources of any ab are minimized to zero with this process, this ab will be removed from ARB populations.

Number of resources for each $ab \in AB$ is calculated with the (9).

$$ab.resources = ab.stimulation \times Clonal\_rate \qquad (9)$$

Competition for resources process is repeated until the average stimulation of each class is greater than stimulation threshold. The stopping criteria allows ARBs to profits the opportunity to produce mutated offspring. If the stopping criterion is met, the remaining ab are used to select a candidate memory cell ($mc_{candidate}$). $mc_{candidate}$ represents the ab that most stimulated with ag and which have the same class of the latter.

– **Memory cell introduction** This stage is the last step in the training process of one antigen. Candidate memory cell $mc_{candidate}$ which has a stimulation greater than that of mc_{match} is introduced into MC as a new memory cell. If the affinity between $mc_{candidate}$ and mc_{match} is less than $AT \times ATS$, $mc_{candidate}$ replaces mc_{match} in MC set. Affinity Threshold Scalar (ATS) is a value between 0 and 1 chosen by the user.

5 The proposed approach

This paper proposes a new method for subtracting the background through a fixed camera in a video surveillance system by performing a combination between AIRS and SG. AIRS algorithm is used as a classification tool that separates antibodies represented by pixels belonging to the background model from antigens that model pixels belonging to the foreground representing moving objects.

AIRS allows to generate a set of background representatives with an iterative process that searching among the existing models the Gaussians which have a degree of similarity closest to the current value of the pixel. In the mutation step, the best representative among all Gaussians previously generated is muted according to an empirically defined rate to predict the values that a pixel can take. After creating new clones, a filtering process is applied on this clones to increase precision. The current pixel is classified as background pixel if it has at a minimum one similar element in all representatives. This combination allowed us to benefit from the advantages of AIRS which allows an online update of the system parameters and Gaussians power to adapt with scene variations at the pixel level. Figure 2 presents the global architecture of our system and details the different modules that composes it.

5.1 Pre-processing

This is a preliminary step in any video processing. During this step, the video is split into sequences of 2D images called frames, which represents the input of our system. These frames are encoded in RGB mode, unfortunately this mode is sensitive to the light effects [15], and the use of this latter does not ensure the creation of an efficient model. HSV is a more robust mode for changing lighting, since it separates the luminosity represented with V component and the chromatic properties represented by H and S components.

In most cases the component V is ignored to reduce the light effects [50, 65]. After many experiments, we noticed that the use of S component did not add a significant quality to the system compared with the gain obtained in processing time by removing the latter. Indeed,
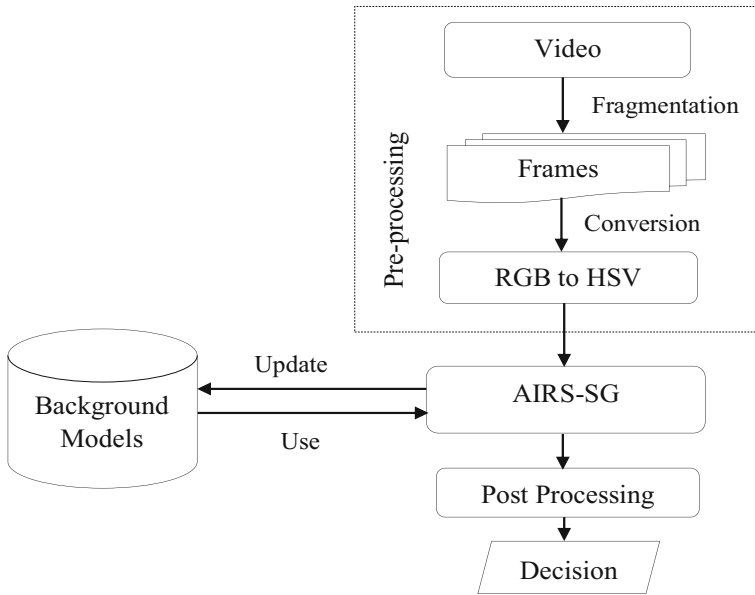
Fig. 2 Proposed architecture

S component express the shadow effect on color. For this purpose, only the H component is taken into consideration.

5.2 AIRS-SG

In dynamic scenes, single Gaussian is sensitive to fast pixel variations. Indeed, a single Gaussian cannot memorize the old states of the pixel. To overcome this problem, we have proposed a new background subtraction method called AIRS-SG. This method uses the principle of SG approach and the robustness of AIRS algorithm to create an efficient model to fast pixel variation. The proposed model considers the pixel value H as antigens (ag) and the corresponding Gaussians as memory cells (mc) that have recognized this antigen. In the following, we will detail the principle of the proposed model AIRS-SG.

5.2.1 Initialization and the creation of first AIRS-SG model

This is a standard step in any background subtraction algorithm. It plays a very important role in the measurement, definition and identification of parameters. In our case, it consists to use SG approach on the first images of the video to initialize the MC set of the each pixel. This step is essential for a faster convergence of the recognition system. ARB populations remains empty during the initialization stage and its construction is done gradually as the system operates.

The affinity measure defined in the AIRS algorithm is not adequate in our context. For this purpose, we proposed a new affinity measurement, that represents the distance between each pixel and their models according to (10).

$$\frac{|P_t - u_t|}{\sigma_t} \tag{10}$$

The novel affinity threshold is calculated with (11).

$$AT = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} \frac{P_{i,j} - u_{i,j}}{\sigma_{i,j}}}{n \times m} \tag{11}$$

Where, n, m is the number of rows and columns in an image; $P_{i,j}$, $u_{i,j}$, $\sigma_{i,j}$ are respectively the pixel value, the mean and the variance of the model provided by SG.

5.2.2 Memory cell identification and ARB generation

For each pixel P_t (ag_t), we generate a subset named $MC_{background}$ from the MC set that verifies (12). Each $mc \in MC_{background}$ is represented by a Gaussian (pixel value, mean u, variance σ). The pixel with an empty $MC_{background}$, will be defined as a foreground pixel. In the case where $MC_{background}$ is not empty, the pixel belongs to the background pixels. After this step, AIRS-SG performs a research process on the closest mc called mc_{match} from $MC_{background}$ according to (13). mc_{match} is mutated with a rate between 0 and 1 to generate new clones. These latter are grouped together and placed with mc_{match} in AB set.

$$MC_{background} = \left\{ \frac{ag_t - u_{mc}}{\sigma_{mc}}, mc \in MC \right\} \tag{12}$$

$$mc_{match} = argmin_{mc \in MC} \left(MC_{background} \right) \tag{13}$$

The used mutation function has also been adapted to take into consideration the proposed feature vector. The novel mutation function is defined as follows:

Algorithm 1 Mutation function proposed.

Require: $g_t \in G, b \in \{0, 1\}$
Ensure: g_t, b
  for each $g_{t_i} \in g_t, i \in [2, 3]$ **do**
    $change \leftarrow Rand()$
    $changeto \leftarrow Prand()$
    if $(change < mutation\_rate)$ **then**
      $g_{t_i} \leftarrow changeto \times random\_value + 1$
      $b \leftarrow true$
    end if
  end for
  if $(b \equiv true)$ **then**
    $change \leftarrow Rand()$
    $changeto \leftarrow Prand()$
    if $(change < mutation\_rate)$ **then**
      $g_{t_1} \leftarrow changeto$
    end if
  end if

$Rand()$: a function that return a random value between 0 and 1.
   $Prand()$: a function that return a random value between 0 and 255.

5.2.3 Competition for resources and development of a candidate memory cells

During this stage, the *ab* generated in the previous step are filtred, to keep only which represents the background according to (14).

$$\frac{P_{ab} - u_{ab}}{\sigma_{ab}} < 2.5 \tag{14}$$

P_{ab}, u_{ab}, σ_{ab}, are respectively the pixel value, the mean and the variance of the Gaussian *ab*.

ab selected are grouped in $MC_{candidate}$ set. We proposed to select all *ab* that satisfies (14) instead of keeping only the best *ab*, such it has been proposed in basic AIRS. This proposition has allowed us to manage the multi-modality of the background.

5.2.4 Memory cells introduction

Each element of $MC_{candidate}$ that has a lower affinity to mc_{match} affinity is introduced as a new memory cell into MC set. If the affinity between $mc_{candidate}$ and mc_{match} is less than $AT \times ATS$, mc_{match} will be removed from the MC set. This step has also benefited from a small contribution in the updating process of memory cell number introduced in the global MC model. Indeed, AIRS as it has been proposed allows to introduce only one memory cell for each iteration. This new mechanism gives the system a high opportunity to produce more representative model. Figure 3 illustrates AIRS-SG flowchart.

5.3 Post processing

The proposed approach produces binary images containing only black and white pixels, black pixels represent the background and white pixels define the foreground (the moving object). However, all background subtraction methods create cuts in the detected objects, isolated pixels and some gaps generated by false detections. In order to overcome these problems we have applied morphological operations on these images.

Dilation operation allows to cover the gaps inside foreground objects, in contrast to erosion operation which eliminates isolated pixels mainly generated by the presence of dust and illumination changes, etc. After the morphological operations, we applied a median filter to correct edges of moving objects.

6 Experiments and results

6.1 Settings and datasets

The proposed method is implemented in Java on a computer equipped with a fourth-generation Intel Core i3 processor and 4 GB memory capacity. In this section we present and discuss obtained results on the CDnet 2014 dataset [54]. CDnet 2014 is a dataset with different types of scenes captured in real conditions. It contains 53 videos divided into 11 categories: Baseline (BL), Dynamic Background (DB), Camera Jitter (CJ), Intermittent Object Motion (IOM), Shadow (Shd), Thermal (Th), Bad Weather (BW), Low Framerate (LFR), Night Videos (NV), Pan Tilt Zoom (PTZ) and Turbulence (Tb). Each category has from 4 to 6 videos. Spatial resolutions of videos varied from 320×240 to 720×576. Authors of this dataset identified training and test data for each video to facilitate comparative
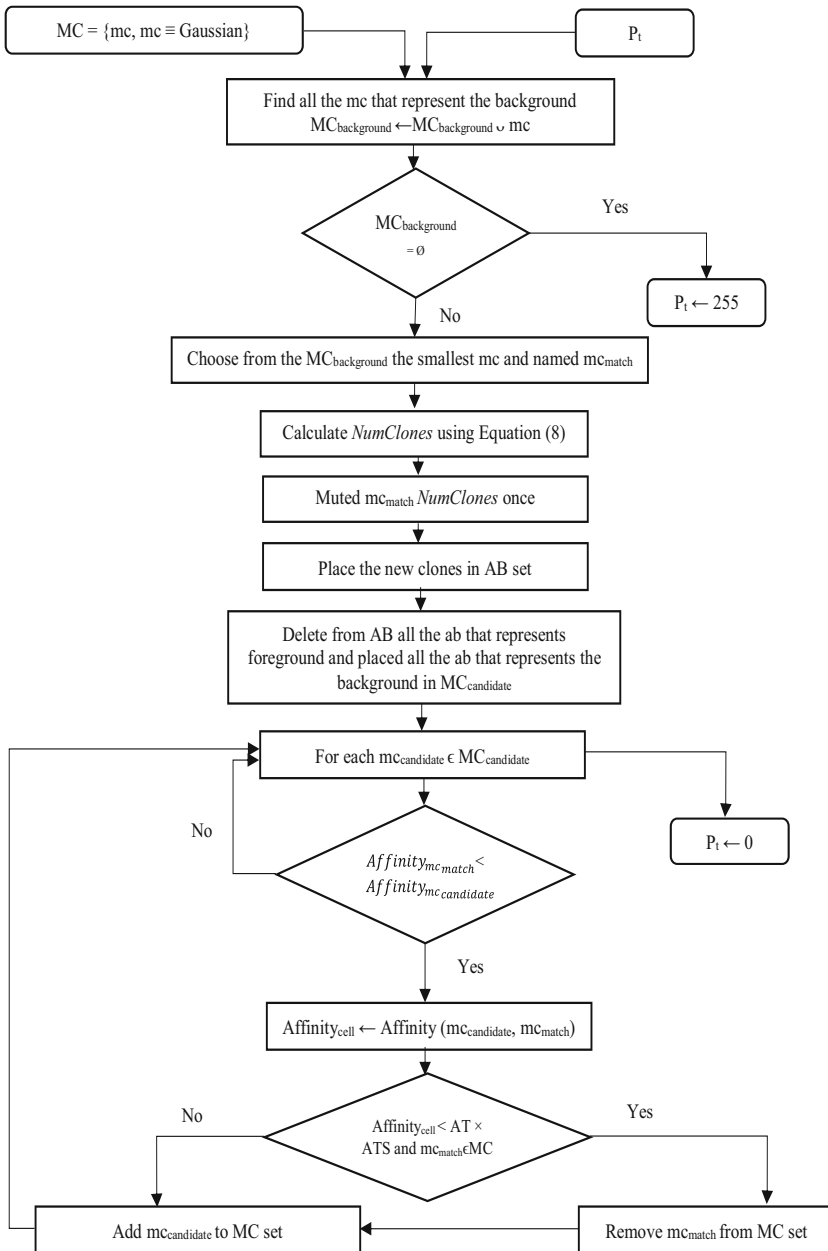
Fig. 3 AIRS-SG flowchart

operation. Noted that in this paper we have interested to problems related to moving objects detection in dynamic environments. In fact, this latter has posed problems for the majority of background subtraction systems. For this purpose, we have selected from CDnet 2014 dataset only videos of dynamic background category (DB). Table 1 shows some details about sequences used in the experiments.

Algorithm 2 AIRS-SG.

Require: $AG = \{ag_1, ag_2, .., ag_n, ag_1, ag_2, ..ag_m, \}, ag_t \equiv P_t$
Ensure: ag_t
  Initialization

  Begin
  MC: A set of memory cells mc, $mc \equiv g_t$.
  AB $\leftarrow \emptyset$
  $AT \leftarrow \dfrac{\sum_{i=1}^{n} \sum_{j=1}^{m} \frac{P_{i,j} - u_{i,j}}{\sigma_{i,j}}}{n \times m}$

  End
  for each $ag_t \in AG$ **do**
      $MC_{background} = \left\{ \frac{ag_t - u_{mc}}{\sigma_{mc}} < 2.5, mc \in MC \right\}$
      if $\left(MC_{background} \equiv \emptyset \right)$ **then**
         $ag_t \leftarrow 255$
      else
         $mc_{match} \leftarrow argmin_{mc \in MC} \left(MC_{background} \right)$
         $NumClones \leftarrow Clonal\_rate \times Hyper\_mutation\_rate \times Affinity \left(ag_t, mc_{match} \right)$
         $MU \leftarrow \emptyset$
         $MU \leftarrow MU \cup mc_{match}$
         while $|MU| < NumClones$ **do**
            $mut \leftarrow false$
            $mc_{clone} \leftarrow mc_{match}$
            $mc_{clone}, mut \leftarrow mutation \left(mc_{clone}, mutation\_rate, mut \right)$
            if $(mut \equiv true)$ **then**
               $MU \leftarrow MU \cup mc_{clone}$
            end if
         end while
         $AB \leftarrow AB \cup MU$
         $MC_{candidate} \leftarrow \emptyset$
         for each $ab \in AB$ **do**
            if $\left(\frac{P_{ab} - u_{ab}}{\sigma_{ab}} > 2.5 \right)$ **then**
               $AB \leftarrow AB - ab$
            else
               $MC_{candidate} \leftarrow MC_{candidate} \cup ab$
            end if
         end for
         $Find \leftarrow false$
         $Affinity_{mc_{match}} \leftarrow \frac{ag_t - u_{mc_{match}}}{\sigma_{mc_{match}}}$
         for each $mc_{candidate} \in MC_{candidate}$ **do**
            $Affinity_{mc_{candidate}} \leftarrow \frac{ag_t - u_{mc_{candidate}}}{\sigma_{mc_{candidate}}}$
            if $\left(Affinity_{mc_{candidate}} < Affinity_{mc_{match}} \right)$ **then**
               $Affinity_{cell} \leftarrow (Affinity_{mc_{match}} + Affinity_{mc_{candidate}})/2$
               if $((Affinity_{cell} < AT \times ATS) \& (Find \equiv false))$ **then**
                  $Find \leftarrow true$
                  $MC \leftarrow MC - mc_{match}$
               end if
               $MC \leftarrow MC \cup mc_{candidate}$
            end if
         end for
         $ag_t \leftarrow 0$
      end if
  end for

Table 1 Description of Dynamic Background category of CDnet 2014 dataset

Category	Number of videos	Video name	Total number of frames	Resolution
Dynamic Background (DB)	6	Boats	7999	320×240
		Canoe	1189	320×240
		Fall	4000	720×480
		Fountain01	1184	432×288
		Fountain02	1499	432×288
		Overpass	3000	320×240

In addition to CDnet 2014 dataset, the proposed approach is also tested on Wallflower (WavingTrees) [49], WaterSurface [27], UCSD [45] and Fountain [27] datasets. The set of parameters used for all the datasets: learning rate (α), $mutation\_rate$, $Clonal\_rate$, $Hyper\_mutation\_rate$ and Affinity Threshold Scalar (ATS) have been set respectively to 0.01, 0.1, 10, 4, 0.01. This choice was fixed empirically after several tests with respect to the influence of parameters on results quality.

AIRS algorithm as many other classification tools need to set the parameters empirically and we did not find any work that can give us a method to get correctly these values. We noticed that the $mutation\_rate$ should be kept very low otherwise convergence may be delayed unnecessarily. The $Clonal\_rate$ controls the number of generated clones in each iteration. In our case we used 100 times the mutation rate due to the small population used in the system. The $Hyper\_mutation\_rate$ controls the number of mutated clones. This latter is set to 4 due to the number of permutations between features (3 possibilities when the mutation is done 2 by 2 and 1 if all features muted at the same time). The Affinity Threshold Scalar is set very low to ensure that the dissimilarity between foreground and background is kept since our feature vector contains only 3 characteristics and the mutation can easily bring the background closer to the foreground.

6.2 Performance evaluation

6.2.1 Qualitative results

This sub-session represents some images of our obtained segmentation on datasets (CDnet 2014 (Dynamic Background), Wallflower (WavingTrees), WaterSurface, UCSD and Fountain) based on ground truth (Tables 2 and 3).

However, this comparison is not sufficient, since the observable result does not ensure a delicate representation of system performance. Therefore, quantitative tests are required in this type of system to describe objectively the robustness of the entire system based on a set of criterions.

6.2.2 Quantitative results

The aim of the experimentation is to prove the efficacy of our system in background subtraction. To evaluate the performance of the proposed system, authors of CDnet 2014 proposed to use seven measures:

1. Recall (Re) : $\frac{TP}{TP+FN}$

Table 2 Some images of obtained results for the Dynamic Background category of the dataset (CDnet 2014)
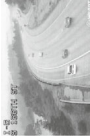
Video name	Fountain01	Fountain02	Canoe	Boats	Overpass	Fall
Test images						
Ground truth						
Results						

Table 3 Some results on some videos of the datasets Wallflower (WavingTrees), WaterSurface, Fountain, UCSD

Video name	WavingTrees	WaterSurface	Fountain	Bottle	Chopper
Test images					
Ground truth					
Results					
Video name	Boats	Birds	Freeway	Peds	Rain
Test images					
Ground truth					
Results					

2. Specificity (Sp) : $\frac{TN}{TN+FP}$
3. False Positive Rate (FPR) : $\frac{FP}{FP+TN}$
4. False Negative Rate (FNR) : $\frac{FN}{TP+FN}$
5. Percentage of Wrong Classifications (PWC) : $100 \times \frac{(FN+FP)}{(TP+FN+FP+TN)}$
6. F_Measure : $\frac{2 \times Precision \times Recall}{Precision+Recall}$
7. Precision (Pre) : $\frac{TP}{TP+FP}$

With:

– True positive (TP): the result is positive(255), while the ground truth is also positive(255)
– False positive (FP): the result is positive(255), but the ground truth is negative(0)
– True negative (TN): the result is negative(0), while the ground truth is also negative(0)
– False negative (FN): the result is negative(0), but the ground truth is positive(255)

Furthermore these parameters, we have calculated the complexity (Table 4) and the accuracy (Table 5) to well evaluate our proposition.

– Accuracy = Number of pixels well detected /Total number of pixels

The complexity of AIRS-SG is: $O(n, m)$

– $T(n \times m) = |AG| \times (5 \times |MC_{candidate}| + |AB| + 14 \times Numclones + 8) + n \times m + 1$

Where: $|AG| = n \times m$

To properly evaluate the proposed system, the results were compared with hand-segmented images called ground truth. These are provided with videos of CDnet 2014 authors (Tables 6, 7, 8, 9, 10 and 11).

6.2.3 Discussions

In this section we will discuss the obtained results on the dynamic background category of the CDnet 2014 dataset.

Our system has achieved good results in Canoe and Boats videos, and acceptable results in Fountain01, Fountain02 and Fall videos. However, our system has failed in Overpass video due to the similarity in colors between the foreground and background which lead the system to make errors and getting inferior rates compared with other state of the art methods. This failing in our system is related to the use of a single component H of HSV color space. Noted that our objective is to detect moving objects. Despite the low rate of some evaluation criteria, our system has succeeded in its task (Detection even partial with marginal false detection) and the qualitative results show the effectiveness of the proposed system to detect moving objects in dynamic scene. The partial detection of moving objects influences on performance measurements based on total quantity of pixels detected in relation to the ground truth.

Deep learning methods exceeded our system due to the use of in-depth learning without features. Even with the success of these latter, they require a lot of time, several learning data and a super computer to achieve such results. In addition, the application of deep learning methods in a real environment present several problems. Indeed, the nature of the environment can include changes in the structure itself (growing tree, changing the color of a wall, new building, etc.) which forces the system to go through a re-initialization step which requires a system shutdown during a long learning period (several days/months). This type

Table 4 Algorithms complexity

Algorithm	Complexity
FgSegNet_v2 [32]	$O(l \times h^N)$ with h hidden neurons, N inputs, and l hidden layers
Cascade CNN [55]	$O(l \times h^N)$ with h hidden neurons, N inputs, and l hidden layers
IUTIS-5 [4]	$O(n^2)$
DeepBS [3]	$O(l \times h^N)$ with h hidden neurons, N inputs, and l hidden layers
WisenetMD [26]	$log_2(R_{Regions})$
SuBSENSE [47]	$O(n \times m)$
CwisarDRP [9]	$O(l \times h^N)$ with h hidden neurons, N inputs, and l hidden layers
M^4CD V2.0 [53]	$O(n \times m)$
SWCD [20]	$O(n \times m)$
MBS [43]	$O(n \times m)$
GMM [48]	$O(n \times m)$
BMN-BSN [41]	$O(l \times h^N)$ with h hidden neurons, N inputs, and l hidden layers
KDE [12]	$O(n \times m)$
AMBER [52]	$O(n \times m)$
FCFNE [63]	$log_N(n \times m)$
LCBC [16]	$O(n \times m)$
AAPSA [42]	$O(r_{neurals} \times n)$
CL-VID [33]	$O(M \times N \times D)$ with M is the dataset size, N is the number of neurons and D is the input dimension
CP3-Online [30]	$O(n \times m \times k_{clusters})$
SSOBS [44]	$O(r_{neurons} \times n)$
AIRS-SG	$O(n \times m)$ with n is the width of the video frame and m is the height of the video frame

of problem is shared with all methods that use offline learning. In the same context, most of the methods that have published their results in CDnet 2014 dataset use supervised learning and their systems have learned videos very well, which justifies the high recognition rate.

AIRS-SG has a linear complexity depending on the number of pixels in a video frame. Our system has a better complexity than deep learning methods and we are at the same level of complexity with the majority of methods cited in the state of the art except WisenetMD

Table 5 Accuracy, Re, Sp, FPR, FNR, PWC, precision, F-measure of CDnet 2014 dynamic background category

Video name	Accuracy	Re	Sp	FPR	FNR	PWC	Pre	F-Measure
Fountain01	95.64%	0.5476	0.9996	0.0003	0.4523	0.0764	0.5390	0.5433
Fountain02	99.56%	0.4945	0.9992	0.0007	0.5054	0.1841	0.5850	0.5359
Canoe	99.17%	0.9773	0.9929	0.0070	0.0226	0.7462	0.7994	0.8794
Boats	99.23%	0.8460	0.9933	0.0066	0.1539	0.7557	0.4459	0.5840
Overpass	96.67%	0.3917	0.9949	0.0050	0.6082	1.3122	0.5141	0.4446
Fall	98.21%	0.4251	0.9930	0.0069	0.5748	1.6119	0.4989	0.4590
The average	98.08%	0.6137	0.9955	0.0044	0.3862	0.7811	0.5637	0.5744

Table 6 Comparison of quantitative results with well-known background subtraction methods on Fontain01 video

Method Name	Video Name	Type of learning	Accuracy	Re	Sp	FPR	FNR	PWC	Pre	F-Measure
FgSegNet_v2 [60]		Offline/Supervised	95.59%	0.9974	1.0000	0.0000	0.0026	0.0012	0.9887	0.9930
Cascade CNN [26]		Offline/Supervised	95.58%	0.9548	0.9999	0.0001	0.0452	0.0163	0.8637	0.9070
IUTIS-5 [32]		Offline/Supervised	95.56%	0.8663	0.9998	0.0002	0.1337	0.0310	0.7836	0.8229
DeepBS [24]		Offline/Supervised	95.56%	0.7241	0.9999	0.0001	0.2759	0.0362	0.8186	0.7685
WisenetMD [44]		Online/Not supervised	95.55%	0.8774	0.9996	0.0004	0.1226	0.0473	0.6622	0.7547
SuBSENSE [21]		Online/Not supervised	95.55%	0.8771	0.9996	0.0004	0.1229	0.0477	0.6599	0.7531
CwisarDRP [27]		Online/Not supervised	95.54%	0.7181	0.9997	0.0003	0.2819	0.0528	0.6694	0.6929
M^4CD V2.0 [22]		Online/Not supervised	94.96%	0.8425	0.9935	0.0065	0.1575	0.6621	0.0973	0.1744
SWCD [43]	Fontain01	Online/Not supervised	95.56%	0.7122	0.9999	0.0001	0.2878	0.0375	0.8127	0.7591
MBS [45]		Online/Not supervised	95.54%	0.3771	0.9999	0.0001	0.6229	0.0570	0.8557	0.5235
GMM [36]		Online/Not supervised	94.06%	0.7973	0.9841	0.0159	0.2027	1.6017	0.0401	0.0763
BMN-BSN [61]		Offline/Supervised	91.35%	0.5569	0.9560	0.0440	0.4431	4.4336	0.0104	0.0204
KDE [46]		Online/Not supervised	94.52%	0.7930	0.9890	0.0110	0.2070	1.1162	0.0565	0.1055
AMBER [42]		Online/Not supervised	95.56%	0.7582	0.9998	0.0002	0.2418	0.0378	0.7807	0.7693
FCFNE [62]		Online/Not supervised	N/A	0.8817	N/A	0.0004	0.1183	0.0528	0.6963	0.7792
LCBC [63]		Online/Not supervised	N/A	0.5402	0.9963	0.0037	0.4598	0.4076	0.1082	0.1802
AAPSA [28]		Online/Not supervised	95.49%	0.5128	0.9993	0.0007	0.4872	0.1076	0.3880	0.4417
SOBS_CF [30]		Online/Not supervised	94.63%	0.7378	0.9901	0.0099	0.2622	1.0065	0.0586	0.1085
CL-VID [64]		Online/Not supervised	92.77%	0.9327	0.9705	0.0295	0.0673	2.9518	0.0256	0.0498
CP3-Online [65]		Online/Not supervised	95.00%	0.7351	0.9941	0.0059	0.2649	0.6142	0.0934	0.1658
SSOBS [66]		Online/Not supervised	91.86%	0.6660	0.9612	0.0388	0.3340	3.9081	0.0140	0.0275
AIRS-SG		Online/Not supervised	95.64%	0.5476	0.9996	0.0004	0.4524	0.0764	0.5391	0.5433

Table 7 Comparison of quantitative results with well-known background subtraction methods on Fontain02 video

Method Name	Video Name	Type of learning	Accuracy	Re	Sp	FPR	FNR	PWC	Pre	F-Measure
FgSegNet_v2 [60]		Offline/Supervised	99.85%	0.9978	1.0000	0.0000	0.0022	0.0024	0.9910	0.9944
Cascade CNN [26]		Offline/Supervised	99.84%	0.9822	0.9999	0.0001	0.0178	0.0131	0.9580	0.9699
IUTIS-5 [32]		Offline/Supervised	98.83%	0.9530	0.9999	0.0001	0.0470	0.0192	0.9576	0.9553
DeepBS [24]		Offline/Supervised	99.82%	0.8784	1.0000	0.0000	0.1216	0.0305	0.9778	0.9254
WisenetMD [44]		Online/Not supervised	99.83%	0.9247	0.9999	0.0001	0.0753	0.0230	0.9668	0.9453
SuBSENSE [21]		Online/Not supervised	99.83%	0.9232	0.9999	0.0001	0.0768	0.0235	0.9658	0.9441
CwisarDRP [27]		Online/Not supervised	99.81%	0.9372	0.9998	0.0002	0.0628	0.0358	0.9004	0.9184
M^4CD V2.0 [22]		Online/Not supervised	99.82%	0.9682	0.9998	0.0002	0.0318	0.0291	0.9035	0.9348
SWCD [43]	Fontain02	Online/Not supervised	99.82%	0.9327	0.9998	0.0002	0.0673	0.0299	0.9285	0.9306
MBS [45]		Online/Not supervised	99.82%	0.8818	0.9999	0.0001	0.1182	0.0324	0.9647	0.9214
GMM [36]		Online/Not supervised	99.76%	0.8717	0.9994	0.0006	0.1283	0.0917	0.7451	0.8035
BMN-BSN [61]		Offline/Supervised	99.76%	0.8837	0.9994	0.0006	0.1163	0.0857	0.7579	0.8160
KDE [46]		Online/Not supervised	99.77%	0.8528	0.9995	0.0005	0.1472	0.0788	0.7955	0.8232
AMBER [42]		Online/Not supervised	99.82%	0.9644	0.9998	0.0002	0.0356	0.0318	0.8959	0.9289
FCFNE [62]		Online/Not supervised	N/A	0.9370	N/A	0.0002	0.0630	0.0345	0.9463	0.9416
LCBC [63]		Online/Not supervised	N/A	0.8939	0.9999	0.0001	0.1061	0.0349	0.9408	0.9168
AAPSA [28]		Online/Not supervised	99.14%	0.9222	0.9930	0.0070	0.0778	0.7118	0.2220	0.3579
SOBS_CF [30]		Online/Not supervised	99.77%	0.9257	0.9993	0.0007	0.0743	0.0833	0.7475	0.8271
CL-VID [64]		Online/Not supervised	99.34%	0.9671	0.9949	0.0051	0.0329	0.5117	0.2919	0.4484
CP3-Online [65]		Online/Not supervised	99.69%	0.6391	0.9992	0.0008	0.3609	0.1576	0.6321	0.6356
SSOBS [66]		Online/Not supervised	99.25%	0.5993	0.9949	0.0051	0.4007	0.5989	0.2009	0.3009
AIRS-SG		Online/Not supervised	99.56%	0.4945	0.9992	0.0008	0.5055	0.1842	0.5850	0.5360

Table 8 Comparison of quantitative results with well-known background subtraction methods on Canoe video

Method Name	Video Name	Type of learning	Accuracy	Re	Sp	FPR	FNR	PWC	Pre	F-Measure
FgSegNet_v2 [60]		Offline/Supervised	98.36%	0.9993	1.0000	0.0000	0.0007	0.0037	0.9996	0.9995
Cascade CNN [26]		Offline/Supervised	98.28%	0.9935	0.9993	0.0007	0.0065	0.0290	0.9804	0.9869
IUTIS-5 [32]		Offline/Supervised	98.01%	0.9052	0.9997	0.0003	0.0948	0.3642	0.9912	0.9462
DeepBS [24]		Offline/Supervised	98.22%	0.9793	0.9992	0.0008	0.0207	0.1460	0.9795	0.9794
WisenetMD [44]		Online/Not supervised	97.54%	0.7677	0.9998	0.0002	0.2323	0.8430	0.9925	0.8658
SuBSENSE [21]		Online/Not supervised	97.16%	0.6590	0.9998	0.0002	0.3410	1.2236	0.9933	0.7923
CwisarDRP [27]		Online/Not supervised	97.76%	0.8296	0.9999	0.0001	0.1704	0.6164	0.9956	0.9051
M^4CD V2.0 [22]		Online/Not supervised	98.00%	0.4653	0.9996	0.0004	0.5347	0.3702	0.8934	0.6119
SWCD [43]	Canoe	Online/Not supervised	97.78%	0.9402	0.9960	0.0040	0.0598	0.5971	0.8963	0.9177
MBS [45]		Online/Not supervised	97.93%	0.8879	0.9995	0.0005	0.1121	0.4407	0.9863	0.9345
GMM [36]		Online/Not supervised	97.56%	0.8659	0.9964	0.0036	0.1341	0.8225	0.8982	0.8817
BMN-BSN [61]		Offline/Supervised	98.32%	0.9862	0.9847	0.0153	0.0138	1.5268	0.7027	0.8206
KDE [46]		Online/Not supervised	97.59%	0.8315	0.9980	0.0020	0.1685	0.7860	0.9396	0.8822
AMBER [42]		Online/Not supervised	97.85%	0.9590	0.9960	0.0040	0.0410	0.5286	0.8986	0.9278
FCFNE [62]		Online/Not supervised	N/A	0.9114	N/A	0.0017	0.0886	0.5464	0.9589	0.9346
LCBC [63]		Online/Not supervised	N/A	0.9394	0.9978	0.0022	0.0606	0.4277	0.9398	0.9396
AAPSA [28]		Online/Not supervised	97.66%	0.7977	0.9999	0.0001	0.2023	0.7240	0.9973	0.8864
SOBS_CF [30]		Online/Not supervised	98.01%	0.9633	0.9976	0.0024	0.0367	0.3615	0.9364	0.9497
CL-VID [64]		Online/Not supervised	97.86%	0.9607	0.9961	0.0039	0.0393	0.5114	0.9014	0.9301
CP3-Online [65]		Online/Not supervised	97.76%	0.8969	0.9974	0.0026	0.1031	0.6130	0.9276	0.9120
SSOBS [66]		Online/Not supervised	81.29%	0.2240	0.8605	0.1395	0.7760	14.3509	0.0100	0.0192
AIRS-SG		Online/Not supervised	99.17%	0.9773	0.9930	0.0070	0.0227	0.7463	0.7994	0.8795

Table 9 Comparison of quantitative results with well-known background subtraction methods on Boats video

Method Name	Video Name	Type of learning	Accuracy	Re	Sp	FPR	FNR	PWC	Pre	F-Measure
FgSegNet_v2 [60]		Offline/Supervised	99.56%	0.9989	0.9999	0.0001	0.0011	0.0128	0.9810	0.9899
Cascade CNN [26]		Offline/Supervised	99.55%	0.9977	0.9998	0.0002	0.0023	0.0224	0.9676	0.9824
IUTIS-5 [32]		Offline/Supervised	99.32%	0.6147	0.9999	0.0001	0.3853	0.2527	0.9723	0.7532
DeepBS [24]		Offline/Supervised	99.36%	0.7098	0.9998	0.0002	0.2902	0.2060	0.9490	0.8121
WisenetMD [44]		Online/Not supervised	99.28%	0.5830	0.9997	0.0003	0.4170	0.2943	0.9178	0.7130
SuBSENSE [21]		Online/Not supervised	99.26%	0.5596	0.9997	0.0003	0.4404	0.3107	0.9106	0.6932
CwisarDRP [27]		Online/Supervised	99.37%	0.8214	0.9992	0.0008	0.1786	0.1946	0.8618	0.8411
M^4CD V2.0 [22]		Online/Not supervised	99.20%	0.9542	0.9978	0.0022	0.0458	0.3769	0.9403	0.9472
SWCD [43]	Boats	Online/Not supervised	99.36%	0.9238	0.9984	0.0016	0.0762	0.2040	0.7877	0.8503
MBS [45]		Online/Not supervised	99.46%	0.8528	0.9998	0.0002	0.1472	0.1135	0.9620	0.9041
GMM [36]		Online/Not supervised	99.22%	0.7582	0.9980	0.0020	0.2418	0.3541	0.7014	0.7287
BMN-BSN [61]		Offline/Supervised	99.62%	0.9436	0.9997	3.0642	0.0564	0.0658	0.9511	0.9473
KDE [46]		Online/Not supervised	99.09%	0.6575	0.9973	0.0027	0.3425	0.4797	0.6089	0.6323
AMBER [42]		Online/Not supervised	99.38%	0.8758	0.9989	0.0011	0.1242	0.1889	0.8319	0.8533
FCFNE [62]		Online/Not supervised	N/A	0.8321	N/A	0.0018	0.1679	0.3197	0.7906	0.8108
LCBC [63]		Online/Not supervised	N/A	0.7806	0.9992	0.0008	0.2194	0.2174	0.8599	0.8183
AAPSA [28]		Online/Not supervised	99.33%	0.6284	0.9999	0.0001	0.3716	0.2429	0.9757	0.7644
SOBS_CF [30]		Online/Not supervised	99.46%	0.9122	0.9994	0.0006	0.0878	0.1098	0.9127	0.9124
CL-VID [64]		Online/Not supervised	99.35%	0.7236	0.9995	0.0005	0.2764	0.2182	0.9101	0.8062
CP3-Online [65]		Online/Not supervised	98.70%	0.8350	0.9922	0.0078	0.1650	0.8748	0.4044	0.5449
SSOBS [66]		Online/Not supervised	85.48%	0.6894	0.8445	0.1555	0.3106	16.0995	0.1400	0.2327
AIRS-SG		Online/Not supervised	99.23%	0.8461	0.9934	0.0066	0.1539	0.7557	0.4460	0.5841

Table 10 Comparison of quantitative results with well-known background subtraction methods on Overpass video

Method Name	Video Name	Type of learning	Accuracy	Re	Sp	FPR	FNR	PWC	Pre	F-Measure
FgSegNet_v2 [60]		Offline/Supervised	96.05%	0.9970	1.0000	0.0000	0.0030	0.0064	0.9982	0.9976
Cascade CNN [26]		Offline/Supervised	96.02%	0.9893	0.9997	0.0003	0.0107	0.0403	0.9808	0.9850
IUTIS-5 [32]		Offline/Supervised	95.88%	0.8837	0.9997	0.0003	0.1163	0.1858	0.9753	0.9272
DeepBS [24]		Offline/Supervised	95.91%	0.9158	0.9996	0.0004	0.0842	0.1523	0.9688	0.9416
WisenetMD [44]		Online/Not supervised	95.76%	0.8095	0.9994	0.0006	0.1905	0.3160	0.9470	0.8728
SuBSENSE [21]		Online/Not supervised	95.72%	0.7852	0.9994	0.0006	0.2148	0.3506	0.9437	0.8572
CwisarDRP [27]		Online/Supervised	95.86%	0.8862	0.9994	0.0006	0.1138	0.2095	0.9542	0.9189
M^4CD V2.0 [22]		Online/Not supervised	95.92%	0.9671	0.9990	0.0010	0.0329	0.1467	0.9266	0.9464
SWCD [43]		Online/Not supervised	95.67%	0.8550	0.9978	0.0022	0.1450	0.4078	0.8429	0.8489
MBS [45]	Overpass	Online/Not supervised	95.82%	0.8427	0.9996	0.0004	0.1573	0.2538	0.9632	0.8990
GMM [36]		Online/Not supervised	95.75%	0.8294	0.9990	0.0010	0.1706	0.3264	0.9191	0.8719
BMN-BSN [61]		Offline/Supervised	95.39%	0.8880	0.9944	0.0056	0.1120	0.7017	0.6832	0.7723
KDE [46]		Online/Not supervised	95.62%	0.8003	0.9981	0.0019	0.1997	0.4550	0.8512	0.8250
AMBER [42]		Online/Not supervised	95.92%	0.9849	0.9987	0.0013	0.0151	0.1439	0.9143	0.9483
FCFNE [62]		Online/Not supervised	N/A	0.9007	0.9994	0.0011	0.0993	0.2663	0.9326	0.9164
LCBC [63]		Online/Not supervised	N/A	0.9524	0.9994	0.0006	0.0476	0.1255	0.9538	0.9531
AAPSA [28]		Online/Not supervised	95.66%	0.7046	0.9998	0.0002	0.2954	0.4142	0.9808	0.8201
SOBS_CF [30]		Online/Not supervised	95.62%	0.9617	0.9959	0.0041	0.0383	0.4522	0.7627	0.8507
CL-VID [64]		Online/Not supervised	95.63%	0.9393	0.9963	0.0037	0.0607	0.4421	0.7772	0.8506
CP3-Online [65]		Online/Not supervised	95.56%	0.6724	0.9991	0.0009	0.3276	0.5236	0.9141	0.7749
SSOBS [66]		Online/Not supervised	90.33%	0.4333	0.9473	0.0527	0.5667	5.9630	0.1004	0.1630
AIRS-SG		Online/Not supervised	96.67%	0.3917	0.9950	0.0050	0.6083	1.3123	0.5142	0.4447

Table 11 Comparison of quantitative results with well-known background subtraction methods on Fall video

Method Name	Video Name	Type of learning	Accuracy	Re	Sp	FPR	FNR	PWC	Pre	F-Measure
FgSegNet_v2 [60]		Offline/Supervised	99.79%	0.9949	1.0000	0.0000	0.0051	0.0130	0.9977	0.9963
Cascade CNN [26]		Offline/Supervised	99.68%	0.9611	0.9994	0.0006	0.0389	0.1282	0.9664	0.9637
IUTIS-5 [32]		Offline/Supervised	99.57%	0.9586	0.9984	0.0016	0.0414	0.2318	0.9147	0.9361
DeepBS [24]		Offline/Supervised	99.14%	0.9182	0.9947	0.0053	0.0818	0.6691	0.7563	0.8294
WisenetMD [44]		Online/Not supervised	99.36%	0.8752	0.9977	0.0023	0.1248	0.4471	0.8728	0.8740
SuBSENSE [21]		Online/Not supervised	99.34%	0.8567	0.9978	0.0022	0.1433	0.4692	0.8758	0.8661
CwisarDRP [27]		Online/Supervised	99.18%	0.7820	0.9976	0.0024	0.2180	0.6260	0.8524	0.8157
M^4CD V2.0 [22]		Online/Not supervised	96.57%	0.9134	0.9686	0.0314	0.0866	3.2409	0.3439	0.4996
SWCD [43]		Online/Not supervised	99.40%	0.8515	0.9985	0.0015	0.1485	0.4091	0.9117	0.8806
MBS [45]	Fall	Online/Not supervised	97.80%	0.7421	0.9842	0.0158	0.2579	2.0099	0.4584	0.5668
GMM [36]		Online/Not supervised	95.75%	0.8838	0.9608	0.0392	0.1162	4.0532	0.2892	0.4358
BMN-BSN [61]		Offline/Supervised	95.65%	0.9473	0.9586	0.0414	0.0574	4.1647	0.2919	0.4463
KDE [46]		Online/Not supervised	92.90%	0.8721	0.9319	0.0681	0.1279	6.9202	0.1875	0.3087
AMBER [42]		Online/Not supervised	97.84%	0.9638	0.9806	0.0194	0.0362	1.9713	0.4724	0.6340
FCFNE [62]		Online/Not supervised	N/A	0.9160	N/A	0.0040	0.0840	0.5486	0.8185	0.8645
LCBC [63]		Online/Not supervised	N/A	0.8270	0.9911	0.0089	0.1730	1.1791	0.6267	0.7131
AAPSA [28]		Online/Not supervised	99.01%	0.6841	0.9976	0.0024	0.3159	0.7946	0.8376	0.7531
SOBS_CF [30]		Online/Not supervised	90.80%	0.9079	0.9098	0.0902	0.0921	9.0211	0.1537	0.2629
CL-VID [64]		Online/Not supervised	88.30%	0.9674	0.8832	0.1168	0.0326	11.5261	0.1300	0.2292
CP3-Online [65]		Online/Not supervised	98.62%	0.5773	0.9956	0.0044	0.4227	1.1844	0.7013	0.6333
SSOBS [66]		Online/Not supervised	93.35%	0.5281	0.9427	0.0573	0.4719	6.4682	0.1425	0.2244
AIRS-SG		Online/Not supervised	98.21%	0.4251	0.9930	0.0070	0.5749	1.6119	0.4989	0.4591

and FCFNE techniques that have better complexity (logarithmic), since they used a block processing method rather than a pixel processing method.

7 Conclusion and perceptive

In this paper we have presented a new approach for background subtraction in dynamic scenes. We have proposed a combination between SG and AIRS in order to overcome the background variation problems which are a deficiency in the most of the subtraction systems proposed in the state of the art. The idea is to exploit the AIRS recognition mechanism which is cable to generate a wide variety of solutions similar to the pixels in the current processing and as a consequence generate several valid and robust models by using only one frame. Indeed, the other methods require several samples to achieve the same purpose. Furthermore, AIRS allows rapid adaptation to the environment through an online update.

On the other hand, the SG approach is very efficient for detecting pixel changes but suffers from problems related to dynamic background. Through the Memory cell identification and ARB generation phase our method predicts and generates new background models. These models are filtered with competition for resources and development of a candidate memory cells process before being selected as new background models.

Evaluations of the proposed method on 6 videos in dynamic background category of CDnet 2014 dataset prove the capacity and precession of our system (AIRS-SG) in the wide variety of pixels. The obtained results are very acceptable in relation to the state of the art, so that our system does not require any human intervention after its launch, thus overcoming the drawbacks of deep learning approaches in the case of a change in environmental infrastructure.

In future work, we propose to increase the number of features used to have additional discriminative power since we have tested only one feature represented by H value of HSV color space. Also we will extend our bio-inspired approach and methodology to other multimedia applications, such as image processing.

References

1. Agatonovic-Kustrin S, Beresford R (2000) J Pharmaceut Biomed Anal 22(5):717
2. Allebosch G, Van Hamme D, Deboeverie F, Veelaert P, Philips W (2015) In: International Joint Conference on Computer Vision, Imaging and Computer Graphics. Springer, pp 433–454
3. Babaee M, Dinh DT, Rigoll G (2018) Pattern Recogn 76:635
4. Bianco S, Ciocca G, Schettini R (2017) IEEE Trans Evol Comput 21(6):914
5. Carranza J, Theobalt C, Magnor MA, Seidel HP (2003) ACM Trans Graph (TOG) 22(3):569
6. Chen Y, Wang J, Lu H (2015) In: 2015 IEEE International Conference on Multimedia and Expo (ICME). IEEE, pp 1–6
7. Cheung SCS, Kamath C (2005) EURASIP J Adv Signal Process 2005(14):726261
8. Clerc M (2005) Hermès-lavoisier février
9. De Gregorio M, Giordano M (2017) In: 25th European symposium on artificial neural networks, computational intelligence and machine learning, pp 453–458
10. Dong E, Han B, Yu X, Du S (2018) In: 2018 IEEE International Conference on Mechatronics and Automation (ICMA). IEEE, pp 1179–1183
11. El Baf F, Bouwmans T, Vachon B (2007) In: SIGMAP, pp 153–158
12. Elgammal A, Harwood D, Davis L (2000) In: European conference on computer vision. Springer, pp 751–767
13. Farou B, Kouahla MN, Seridi H, Akdag H (2017) Eng Appl Artif Intell 64:1

14. Friedman N, Russell S (1997) In: Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., pp 175–181
15. Haq AU, Gondal I, Murshed M (2010) In: 2010 IEEE Symposium on Computers and Communications (ISCC). IEEE, pp 529–534
16. He W, Yong K, Kim W, Ko HL, Wu J, Li W, Tu B (2019) IEEE Access 7:92329
17. Hongwei X, Qian C, Weixian Q (2016) Laser Optoelectron Progress 4(1):3
18. Hou N, He F, Zhou Y, Chen Y (2019) Front Comput Sci 19:1
19. Ishibuchi H, Tsukamoto N, Nojima Y (2008) In: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). IEEE, pp 2419–2426
20. Işık Ş, Özkan K, Günal S, Gerek ÖN (2018) vol 27
21. Jiang S, Lu X (2017) IEEE Transactions on Circuits and Systems for Video Technology
22. Jianzhao C, Victor OC, Gilbert OM, Changtao W (2017) In: 2017 9th International Conference on Modelling, Identification and Control (ICMIC). IEEE, pp 133–138
23. Khan SA, Ishtiaq M, Nazir M, Shaheen M (2018) J Comput Sci 28:94
24. Krungkaew R, Kusakunniran W (2016) In: 2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). IEEE, pp 1–6
25. Laugraud B, Piérard S, Van droogenbroeck M (2018) J Imaging 4(7):86
26. Lee SH, Kwon SC, Shim JW, Lim JE, Yoo J (2018) arXiv:1805.09277
27. Li L, Huang W, Gu IYH, Tian Q (2004) IEEE Trans Image Process 13(11):1459
28. Li H, He F, Liang Y, Quan Q (2019) Soft Comput 24(9):1–20
29. Li H, He F, Yan XH (2019) Appl Math-A J Chin Universities 34(1):1
30. Liang D, Hashimoto M, Iwata K, Zhao X et al (2015) Pattern Recogn 48(4):1374
31. Lim LA, Keles HY (2018) arXiv:1801.02225
32. Lim LA, Keles HY (2018) arXiv:1808.01477
33. López-Rubio E, Molina-Cabello MA, Luque-Baena RM, Domínguez E (2018) vol 28
34. Lu X (2014) In: 2014 IEEE International Conference on Image Processing (ICIP). IEEE, pp 3268–3271
35. Ma C, Liu D, Peng X, Li L, Wu F (2019) J Vis Commun Image Represent 60:426
36. Maddalena L, Petrosino A (2012) In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. IEEE, pp 21–26
37. Maddalena L, Petrosino A (2010) Neural Comput Appl 19(2):179
38. Martins I, Carvalho P, Corte-Real L, Alba-Castro JL (2017) In: Iberian Conference on Pattern Recognition and Image Analysis. Springer, pp 50–57
39. Men Y, Zheng J, Meng L (2016) Video Eng 24(4):5
40. Miron A, Badii A (2015) In: 2015 International Conference on Systems, Signals and Image Processing (IWSSIP). IEEE, pp 273–276
41. Mondéjar-Guerra V, Rouco J, Novo J, Ortega M (2019) In: British machine vision conference (BMVC), Cardiff
42. Ramírez-alonso G, Chacón-murguía MI (2016) Neurocomputing 175:990
43. Sajid H, Cheung SCS (2017) IEEE Trans Image Process 26(7):3249
44. Sehairi K, Chouireb F, Meunier J (2017) J Electron Imaging 26(2):023025
45. Srivastava S, Ng KK, Delp EJ (2011) In: 2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, pp 60–65
46. St-Charles PL, Bilodeau GA, Bergevin R (2015) In: 2015 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, pp 990–997
47. St-Charles PL, Bilodeau GA, Bergevin R (2015) IEEE Trans Image Process 24(1):359
48. Stauffer C, Grimson WEL (1999) In: Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149), vol 2. IEEE, pp 246–252
49. Toyama K, Krumm J, Brumitt B, Meyers B (1999) In: Proceedings of the Seventh IEEE International Conference on Computer Vision, vol 1. IEEE, pp 255–261
50. Wang HY, Ma KK (2003) In: 2003. ICIP 2003. Proceedings. 2003 International Conference on Image Processing, vol 1. IEEE, pp I–153
51. Wang R, Bunyak F, Seetharaman G, Palaniappan K (2014) In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 414–418
52. Wang B, Dudek P (2014) In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 395–398
53. Wang K, Gou C, Wang FY (2018) IEEE Access 6:15505
54. Wang Y, Jodoin PM, Porikli F, Konrad J, Benezeth Y, Ishwar P (2014) In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 387–394
55. Wang Y, Luo Z, Jodoin PM (2017) Pattern Recogn Lett 96:66

56. Watkins A (2001) Airs: a resource limited artificial immune classifier. Ph.D. thesis, Mississippi State University Mississippi
57. Watkins A, Timmis J, Boggess L (2004) Genet Program Evolvable Mach 5(3):291
58. Wren CR, Azarbayejani A, Darrell T, Pentland AP (1997) IEEE Trans Pattern Anal Mach Intell 19(7):780
59. Yang XS (2010) In: Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, pp 65–74
60. Yang XS (2010) arXiv:1003.1409
61. Yong JS, He F, Li H, Zhou WQ (2019) Appl Math-A J Chin Univ 34(4):480
62. Yu H, He F, Pan Y (2019) Multimed Tools Appl 36(2):1–23
63. Yu T, Yang J, Lu W (2019) IEEE Access 7:14671
64. Zhang S, He F, Ren W, Yao J (2018) Vis Comput 79(9):1–12
65. Zhao M, Bu J, Chen C (2002) In: Multimedia systems and Applications V, vol 4861. International Society for Optics and Photonics, pp 325–333
66. Zivkovic Z (2004) In: 2004. ICPR 2004. Proceedings of the 17th International Conference on Pattern Recognition, vol 2. IEEE, pp 28–31