Multimed Tools Appl (2019) 78:5941–5967
https://doi.org/10.1007/s11042-018-6376-8

Similarity based image selection with frame rate adaptation and local event detection in wireless video sensor networks

**Christian Salim[1,2] · Abdallah Makhoul[1] ·
Rony Darazi[2] · Raphaël Couturier[1]**

Received: 25 October 2017 / Revised: 24 April 2018 / Accepted: 3 July 2018 /
Published online: 14 July 2018

Abstract Wireless Video Sensor Networks (WVSNs7unding environmental information. Those sensor nodes can locally process the information and then wirelessly transmit it to the coordinator and to the sink to be further processed. As a consequence, more abundant video and image data are collected. In such densely deployed networks, the problem of data redundancy arises when information are gathered from neighboring nodes. To overcome this problem, one important enabling technology for WVSN is data aggregation, which is essential to be cost-efficient. In this paper, we propose a new approach for data aggregation in WVSN based on images and shot similarity functions. It is deployed on two levels: the video-sensor node level and the coordinator level. At the sensor node level the proposed algorithms aim at reducing the number of frames sensed by the sensor nodes and sent to the coordinator. At the coordinator level, after receiving shots from different neighbouring sensor nodes, the similarity between these shots is computed to eliminate redundancies and to only send the frames which meet a certain condition to the sink. The similarity between shots is evaluated based on their color, edge and motion information. We evaluate our approach on a live scenario and compare the results with another approach from the literature in terms of data reduction and energy consumption. The results show that the two approaches have a significant data reduction to reduce the energy consumption, thus

✉ Christian Salim
christian.salim@univ-fcomte.fr; christian.salim@ua.edu.lb

Abdallah Makhoul
abdallah.makhoul@univ-fcomte.fr

Rony Darazi
rony.darazi@ua.edu.lb

Raphaël Couturier
raphael.couturier@univ-fcomte.fr

[1] Femto-st institute, University Bourgogne Franche-Comté, Belfort, France

[2] TICKET Lab, Antonine University, Hadat-Baabda, Lebanon

our approach tends to overcome the other one in terms of reducing the energy consumption related to the sensing process, and to the transmitting process while guaranteeing the detection of all the critical events at the node and the coordinator levels.

Keywords Wireless video sensor networks · Shot similarity · Video aggregation · Frames similarity · Event detection

1 Introduction

Nowadays, after the development of Wireless Video Sensor Networks (WVSN), the enhancement of the surveillance in terms of monitoring and detecting criticalities and anomalies has set big improvements in different fields (e.g. street, forest, traffic, personal, healthcare, industrial monitoring, etc [1]). Hence, after each anomaly and emergency detection, decisions must be made at the coordinator level. The coordinator may be a normal node or a specific node with greater ressources. It manages a zone of interest, analyzes the data received from several camera sensor nodes and sends the necessary information to the sink which controls the whole network as shown in Fig. 1. Different types of anomalies exist depending on the monitored environment and the predefined criteria and parameters such as quick motion, sound, or scene change, the decisions are made in order to avoid any action that can affect the monitored environment.

The detection of irregularities in any monitored scene is one of the main targets in WVSN. Every scene is permanently filmed using multiple video-sensor nodes. At the sensor node level, the sensor-nodes collect frames and send those frames to the coordinator. The coordinator is responsible for the data aggregation process. The aggregation is limited to either selecting, fusing or deleting the received frames. As a consequence, a significant amount of energy is consumed due to the huge amount of captured frames, which reduces the lifetime of the network. Moreover, the continuous transmission process between all the components of the network (sensor nodes, coordinators and sink) has a big influence on the bandwidth capacity of the network which may cause a bottleneck on the network [2].

Video-sensor nodes operate periodically in WVSN. We define some keywords:

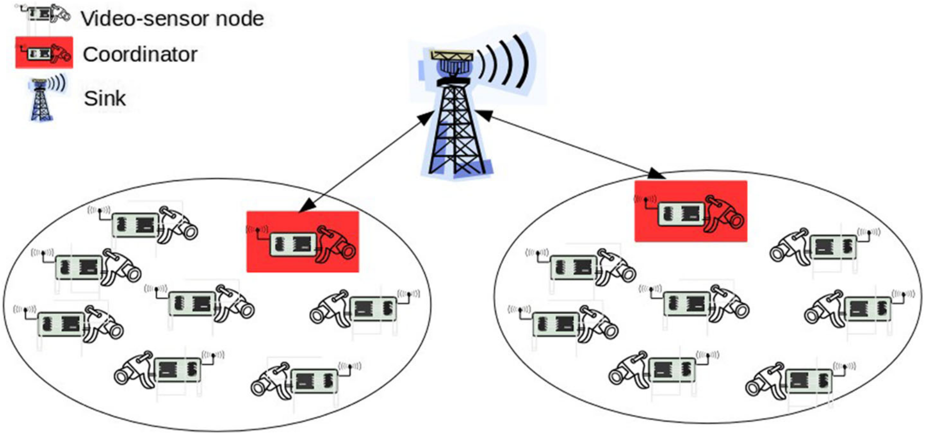


Fig. 1 Architecture of WVSN

A fixed frame rate is defined on every sensor node to film the video accordingly, this frame rate is the number of captured frames per second (fps).

A period is a fixed time length during which frames are captured with a given frame rate.

A video shot is considered as a video sequence taken within a period.

Energy consumption and bandwidth limitation are two important challenges in WVSN. The first one is related to the sensing and transmission modules of the sensor node. The higher the frame rate and the number of frames sent, the more energy is consumed. The second one is related to the transmission module of the sensor node and the coordinator, the greater the number of frames sent on the network is the more bandwidth is used. The energy consumption and bandwidth usage issues on the coordinator's side can be addressed by reducing the amount of sent data from the coordinator to the sink node. In our approach, the data analysis starts at the sensor node level and continues at the coordinator level to match the greatest reduction possible in terms of energy and bandwidth consumptions on both levels. Each video-sensor node compares all the frames in a shot to the last frame sent and computes the similarity between them. Based on the similarity function, only the frames in which an event occurs are sent. The selected frames are called critical frames and are sent to the coordinator. The similarity function at the sensor node level is based on color and edge similarities able to compare frames. This comparison selects the least required number of captured frames to be sent to the coordinator. By applying the similarity function, we reduce the energy consumption related to the *Communication* process by reducing the number of transmitted data.

Alongside the similarity function, the frame rate of each video-sensor node is adapted. A method based on signal frequencies presented in [3] is adopted and applied to WVSN in our approach. This method consists in reducing the number of frames captured by adapting the frame rate of each video-sensor node based on the number of critical frames detected in several consecutive past periods. Consequently, by adapting the frame rate, the *Sensing* process is reduced thus decreasing the energy consumption. At the coordinator level an updated version of the similarity function is implemented in which the motion similarity is added to the color and edge similarities. To avoid comparing all received shots at the coordinator level, a geometric study and a filtering condition are presented. Those conditions consist in reducing the number of possible comparisons. The remainder of this paper is organized as follows. In Section 2, we present the related work to our approach. In Section 3, we describe the proposed method at the sensor node level within its two aspects: the local detection system and the adaptive sampling system, as well as their corresponding algorithms. In Section 4, the data aggregation scheme is described and the proposed geometric method at the coordinator level is introduced. The experimental results and the comparison with another method are given in Section 5. Finally, we conclude in Section 6 with perspectives and future work.

2 Related work

Several research work dealing with data redundancy and energy reduction have been conducted so far [5, 6, 15, 16, 21]. In [16], Akkaya et al. introduced a GPS module into scalar sensors in order to control the cameras. Thus, the system detects which camera should be actuated based on the sensor's position. In [21], Priyadarshini et al. proposed an approach which eliminates redundancies caused by the overlapping of the FOV's (Field Of View) of the video-sensors. To do so, it tends to turn off some cameras and activate the optimal number of cameras at the same time. In [6], Bahi et al. proposed an in-network data aggregation technique at the coordinator level which identifies the nearly duplicate nodes that generate similar data.

In [25], Akkaya et al. discussed the background subtraction (BS) and compression techniques as common data reduction schemes, which have been used for camera sensors to reduce energy consumption.

In [9] and [28], almost all of the studies deal with the physical and network layers. In [9] the authors use a CMOS image sensor where the image is recreated from two outputs, with the details in stationary objects and the suppressed motion in moving objects. It should be noticed that a high frame rate is only applied in the region-of-interest where it matters the most to detect and track any event.

In [18], the authors proposed two new approaches based on the cover set concept to help a node in finding its redundancy level. They proposed an algorithm to schedule the activity of sensor nodes according to the overlapping degree between sensors, and to know for certain if a sensor belongs to the cover set of another sensor.

In [19], the authors proposed a scheduling network solution to minimize power consumption using the multipath theory in wireless video sensor networks. They proposed an algorithm that transmits packets over multipath according to their importance.

Different strategies has been used to reduce energy consumption and bandwidth usage by using an adaptive video streaming etc. that can minimize the utilization of network bandwidth taking into consideration that bandwidth is the most important ressource in a network [13, 26, 30, 32]. All these works help to increase the lifetime of the network. Increasing the lifetime of the network is also studied in [31] specifically for smart camera network.

Several proposed methods in the literature discuss the similarity of images [20, 24, 29]. In [29], the authors used the L_1–, L_2– and L_∞– distance between two cumulative color histograms to simulate the similarity between two color images. In [20], they are interested in the segmentation techniques to compute the similarity, all the techniques are mainly edge based techniques. In [24], the comparison is achieved through an exercise in determining the lack of spatial correlation between two images.

Many methods have been proposed in the literature concerning the visual information and motion estimation in wireless video sensor networks [10, 11, 17, 27]. In [10], the authors studied the correlation in visual information between different cameras with overlapped field of views (FOVs) where the new spatial correlation model function for visual information is implemented. The joint effect of multiple correlated cameras is taken into consideration in this study. An entropy-based analytical framework is developped to measure the amount of visual information provided by multiple cameras. The authors designed a correlation based camera selection algorithm which reduces the energy dissipation of the communication and the computation. This algorithm requires fewer cameras to report to the sink than a random algorithm.

In [11], Jbeily and al. proposed a new symmetric-object oriented approach for motion estimation in WVSN called SYMO-ME which reduces the high complexity of motion estimation, the authors main objective is to reduce the redundancy between successive frames. They adapt a new motion estimation energy consumption model for block matching algorithms (BMAs) in WVSN. This model depends on the energy consumption value of different executed instructions.

Many previous works focused on the scheduling method [4, 5, 7, 12, 22, 33]. In [5], the authors used a clustering methodology. They managed to make a scheduling approach to all overlapping cameras in the same cluster to avoid redundant data. Jiang et al. in [12] proposed a probability scheduling approach based on the kinematics functions and normal law to study the expected positions of the intrusion depending on the kinematics functions to track its trajectory.

In previous works regarding the similarity process, they do not use a pixel by pixel technique. They use the color histograms for color images [29] which can mislead the comparison if the same color happens to be in another place in the area with the same intensity. None of the mentioned works have proposed a data aggregation method at the coordinator level while taking into consideration data reduction performed at the sensor node level for energy consumption. In this paper, both levels are taken into consideration, the sensor node and the coordinator levels. The reduction in terms of energy and bandwidth consumptions is the main purpose of this paper. On the sensor level, a combination of color and edge techniques is established to do the comparison between several images to send only the appropriate frames to the coordinator. The coordinator is responsible for sending to the sink the non similar frames received from different sensor nodes. A geometrical condition is implemented on the coordinator to select the sensor nodes where the comparison must take place.

3 Local detection system: sensor node level

The proposed method is divided into two sections. The first one consists of a local detection function that detects any change in the frames in order to be sent to the coordinator. This function is introduced in every period of our proposed “Multimedia Adaptive Sampling Rate Algorithm” ($MASRA$). The second section presents $MASRA$ algorithm. This algorithm adapts the sampling frequency of each sensor node based on the monitored area.

3.1 Local detection system

In this section, the frame analysis at the video-sensor node level is introduced. This analysis helps sending only the different frames to the coordinator in order to prevent sending all the frames which costs in terms of energy and bandwidth. In some multimedia applications [34], only the middle frame of a shot is used to represent the shot content. But this solution could represent only static shots without taking into consideration the color similarity between frames in the shot nor the edge similarity or the motion similarity, etc.

Comparing the new approach to the Structural Similarity (SSIM) Index quality assessment index, which is based on the multiplicative combination of the luminance, the contrast and the structural terms, shows that this new approach conserves the information and is less complex than SSIM. Thus, SSIM is not used with tiny sensor-nodes because it drains energy a lot more than two simple low-level similarity metrics (color and edges). To compare between SSIM and Color-Edge function in Multimedia Adaptive Sampling Rate Algorithm (MASRA), we implement both algorithms on raspberry pi 3 using c++ for openCV. For the same images input, the results of the execution time needed are shown in Table 1. The important execution time needed to run SSIM function proves why the SSIM is not used for tiny sensor nodes applications.

The proposed approach uses color and edge properties to find similarities between frames, to decide which frame to send. Below a brief explanation is presented to argument the choice of these two properties together and to prove their complementarity. Those two properties have been chosen for simple reasons: the edge property detects any change in the form of the objects in the area of interest or detects a new object that enters the scene. If a new object enters the scene, this property will represent new edges in the gray-scale format as explained later in the paper.

As for the color property, it detects any change in the colors of the scene, an example of such a case is the change of the luminosity of the monitored scene when a burgler turns the lights off before acting. To conclude, the edge property cannot detect a change in the luminosity of the scene, and the color similarity cannot detect a new overlapping object in the scene if it has the same existing color. Thus, those two properties are complementary and are considered of equal importance in the rest of the paper. They are also equally weighted in the similarity function of the approach.

3.1.1 Color similarity

Each frame is compared to the last frame which has been sent to the coordinator. This comparison includes color similarity between frames. An image is generally a 2D matrix M(n,m). Each pixel is divided into 3 different colors to be able to add the RGB color criteria. To do so, the original matrix of the image is transformed from a 2D to a 1D matrix, each element defining a pixel. Then, each pixel is represented by its 3 colors RGB by column (3 columns are needed). In brief, the RGB colors concentration of every pixel in the image is represented by a 2D matrix where the rows represent pixels and the columns represent the RGB colors concentration as shown in the matrix below:

$$M = \begin{array}{l} \\ pixel_0 \\ pixel_1 \\ pixel_2 \\ \vdots \\ \vdots \\ pixel_{n\times m} \end{array} \begin{array}{c} \begin{array}{ccc} Red & Green & Blue \end{array} \\ \begin{pmatrix} 2 & 3 & 4 \\ 20 & 60 & 40 \\ 5 & 10 & 20 \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ .... & .... & .... \end{pmatrix} \end{array}$$

This color similarity consists in comparing the two frames pixel by pixel. First, it computes the total distance for each color between the two frames as shown in equation 1. Then, it normalizes each distance by dividing it by $n \times m \times 255$, Where $n \times m$ is the number of pixels in the image, 255 is the maximum concentration of a color. Three distances are computed $distance_{red}$, $distance_{green}$ and $distance_{blue}$, each one normalized and $\in [0;1]$. E.g, for an image of 540x360=194400 pixels, each of the 3 main distances is divided by 194400×255. The distance for each color (column) is computed as mentioned below:

$$distance_c = \frac{1}{n \times m \times 255} \times \sum_{i=0}^{n\times m} \sqrt{[M_1(i,c) - M_2(i,c)]^2} \qquad (1)$$

Where c is the color (R,G or B), i is the pixel in comparison. To compute the total distance difference between the two in comparison frames, a normalization of the sum of those 3

Table 1 Execution time comparison for SSIM and color-edge function

Function	Execution time
SSIM	3.7 s
Color-Edge	0.1 s

distances is a must by dividing this sum by 3 so the total distance ∈ [0;1]. The color similarity function Col_sim is the inverse of the total distance and computed as follows:

$$Col\_sim = 1 - \frac{\sum distance_c}{3} \tag{2}$$

The distance is computed alone for every column then it is aggregated to be able to compute Col_sim. In the past equations, c refers to any color vector in "RGB" color space while M_1 and M_2 are two Matrices composed of 3 vectors R,G and B.

3.1.2 Edge similarity

This similarity function is way less expensive in terms of energy consumption compared with the color similarity function. In this function, the compared frames are converted to their gray level format. Comparing the edges via their gray scale pixel values in the frame will not be affected by the absence of color.

The used function takes the grayscale image as an input, and returns a binary image BW of the same size as an output. The output image contains 1's where the function finds edges in the input image and 0's elsewhere using the canny function. As presented in [8], edges are found by looking for local maxima of the gradient in the input image. The gradient is calculated using the derivative of a Gaussian filter. The method uses two thresholds, to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. This method is therefore less likely than the others to be fooled by noise, and more likely to detect true weak edges. We compute all the edges in each frame using this function. When an edge is detected the number of edge points is incremented. The edge points represent the total number of edges in the frame. If both frames represent an edge in the same area, the number of matched edge points between both frames is incremented. Then the percentage of matched data which represents the edge similarity between the two frames is calculated:

$$Total\_points = \sum edge\_points \tag{3}$$

$$Matched\_points = \sum Matched\_edge\_points \tag{4}$$

$$Edge\_sim = \frac{Matched\_points}{Total\_points} \tag{5}$$

Where $Total\_points$ are the number of edge points in a frame, $Matched\_points$ are the number of edges in common between the two frames in comparison. The edge similarity $Edge\_sim$ is the ratio of the $Matched\_points$ over the $Total\_points$ of the first frame.

Definition 1 (**Similarity Function**) It is the combination of the two independent similarities (color and edge). This function is the sum of the product between each similarity and its weighting factor (Col_fact and $Edge\_fact$). It is represented as follows:

$$Sim = Col\_sim \times Col\_fact + Edge\_sim \times Edge\_fact \tag{6}$$

Where $Col\_fact + Edge\_fact = 1$.

As mentioned before, the edge property cannot detect a change in the luminosity of the scene, and the color similarity cannot detect a new overlapping object in the scene if it has the same existing color. Thus, Color and Edge similarities are complementary and each one targets different aspects of the image. For this reason, they are weighted equally in the reminder of this paper.

3.2 Multimedia Adaptive Sampling Rate Algorithm (MASRA)

In this section, we focus on the reduction of the number of sensed frames on every video-sensor node. This reduction is based on adapting the frame rate of every sensor node. Inspired from [3], it consists in reducing the number of sensed frames at the sensor node level.
The term "frame rate FR" in this approach is used in the reminder of this paper as the frame rate per period. A period consists of several seconds depending on the needs of the application.

To add that a condition must be satisfied in order to send a frame to the coordinator. This condition helps reducing the energy and the bandwidth consumption by decreasing the number of sent frames from the video-sensor node to the coordinator. Critical frames are only sent to the coordinator.

The first frame of each period is always sent to the coordinator as described in Algorithm 1.

Definition 2 (Critical frame) A critical frame is defined as a frame that represents a degree of similarity "sim" smaller than a predefined threshold th_{sim} as presented in the LDS function of (Algorithm 1). E.g, if the predefined threshold (least similarity needed) is set to 75%, supposing that $frame_{n-1}$ is sent to the coordinator, if $frame_n$ is similar to $frame_{n-1}$ lesser than 75%, it is also sent.

Our objective in this method is to detect changes that are associated with the number of critical frames Nb_Cr_0 per period, where Nb_Cr_0 is directly related to the minimum sampling frame rate FR, denoted as follows:

$$FR >= 2 \times Nb\_Cr_0 \tag{7}$$

In the proposed $MASRA$ algorithm (Algorithm 1) Nb_Cr_0 is defined as the number of critical frames per period. We define FR as follows:

$$FR = c \times Nb\_Cr_0 \tag{8}$$

Where c is a confidence parameter between 2 and 5 as presented in [3]. In order to detect the variation in the number of critical frames, a user-defined confidence parameter d that represents the minimum detectable change (e.g, if $d = 0.2$ then changes that affect Nb_Cr_0 for more than $0.2 \times Nb\_Cr_0$ must be detected). A change is detected in the process when in the current period the current number of critical frames denoted Nb_Cr_i overcomes one of the following thresholds for h consecutive periods :

$$th_{up} = Nb\_Cr_0 \times (1 + d) \tag{9}$$

$$th_{down} = Nb\_Cr_0 \times (1 - d) \tag{10}$$

In this case the frame rate FR is modified according to the last value of Nb_Cr_i in order to adapt the frame rate as shown in the $MASRA$ algorithm (Algorithm 1).

Algorithm 1 Adapting Sampling Rate Algorithm $MASRA$

```
Store the first period's critical frames in $Nb\_Cr_0$
Set $FR = c \times Nb\_Cr_0$
Set $d$
Compute $th_{up}$
Compute $th_{down}$
Set $h$=number of consecutive detected changes required
Set $h_{up} = 0$
Set $h_{down} = 0$
Set $th_{sim}$ the threshold similarity
Set $Col_{fact}$ and $Edge_{fact}$
Set $sim$ the overall similarity between two frames
while $Energy > 0$ do
    for each period do
        Start LDS
        Takes first frame $f_0$
        Sends first frame $f_0$
        Takes frame $f_i$ at $R_t$ Rate
        Compares $f_i$ to the latest sent frame
        $sim = Col\_sim \times Col\_fact + Edge\_sim \times Edge\_fact$
        if $sim < th_{sim}$ then
            sends critical frame $f_i$
        end if
        End LDS
        $Nb\_Cr_i$=number of critical frames in this period
        if $Nb\_Cr_i > th_{up}$ then
            $h_{up} = h_{up} + 1$
            $h_{down} = 0$
        else if $F_{curr} < th_{down}$ then
            $h_{down} = h_{down} + 1$
            $h_{up} = 0$
        else
            $h_{up} = h_{down} = 0$
        end if
        if $(h_{up} > h)||(h_{down} > h)$ then
            Compute $Nb\_Cr_0 = Nb\_Cr_i$
            Compute $FR = c \times Nb\_Cr_0$
            Compute $th_{up}$
            Compute $th_{down}$
        end if
    end for
end while
```

To sum up, the sensor node starts by sending the first frame to the coordinator and then compares the second sensed frame to the previously sent frame. The comparison is done based on the LDS similarity function presented in Algorithm 1. The second frame is sent to the coordinator based on the output of Algorithm 1. According to the number of sent frames in each period, Algorithm 1 detects if this number exceeds one of the two predefined thresholds th_{up} or th_{down}. If the previous condition is satisfied for h consecutive periods, the frame rate FR changes as follows:

$$FR = 2 \times Nb\_Cr_i \tag{11}$$

Where Nb_Cr_i is the number of frames sent (critical frames) in the last period.

4 Data aggregation scheme: the overlapping method

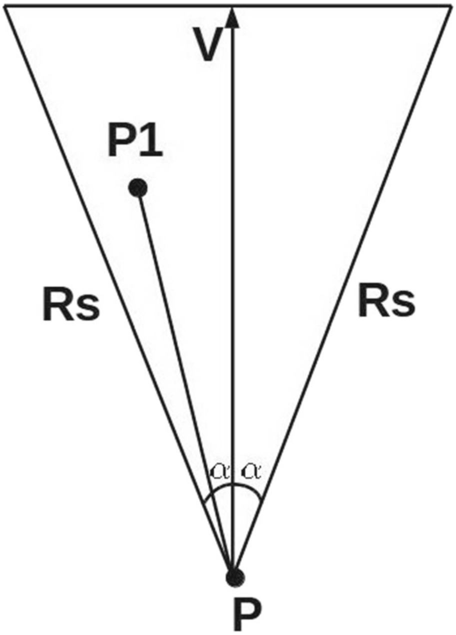
4.1 Video sensing model

A video sensor node S is represented by the FoV of its camera. In our approach, we consider a 2-D model of a video sensor node where the FoV is defined as a sector denoted by a 4-tuple $S(P, R_s, \overrightarrow{V}, \alpha)$. Here P is the position of S, R_s is its sensing range, $\overrightarrow{V}$ is the vector representing the line of sight of the camera's FoV which determines the sensing direction, and α is the offset angle of the FoV on both sides of $\overrightarrow{V}$. Figure 2 illustrates the FoV of a video sensor node in our model. In [18] the authors presented the FOV with 4 points a,b,c and the center of gravity g as shown in Fig. 3 to be able to detect the overlapping areas according to those points.

A point P_1 is said to be in the FoV of a video sensor node S if and only if the two following conditions are satisfied:

1. $d(P, P_1) \leq R_s$, where $d(P, P_1)$ is the Euclidean distance between P and P_1.
2. The angle between $\overrightarrow{PP_1}$ and $\overrightarrow{V}$ must be within $[-\alpha, +\alpha]$.

Fig. 2 FOV



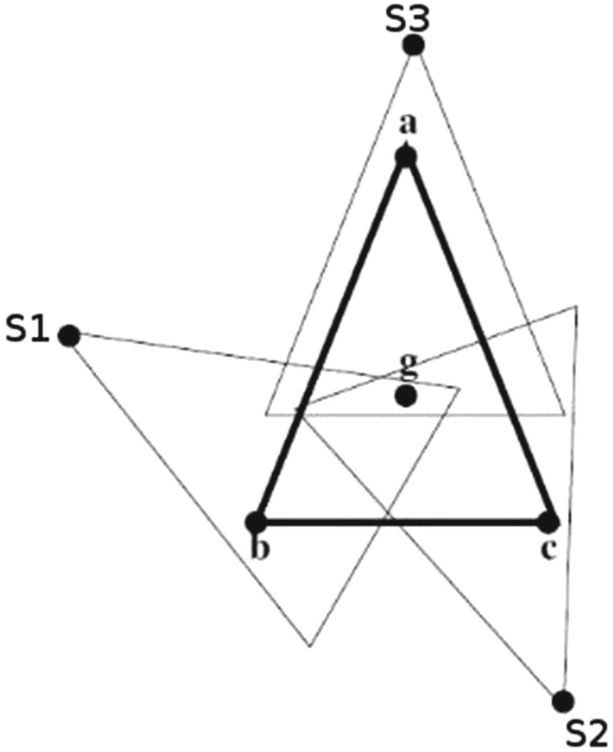


Fig. 3 Video sensing and overlapping model

In other words, these two conditions are met if:

$$\|\overrightarrow{PP_1}\| \leq R_s \tag{12}$$

$$\overrightarrow{PP_1}.\overrightarrow{V} \geq \|\overrightarrow{PP_1}\| \times \|\overrightarrow{V}\| \times cos\alpha. \tag{13}$$

In the remainder of this paper, we consider that all video nodes have the same characteristics: same sensing range R_s and same offset angle α.

In this part, the frame analysis at the coordinator level is introduced. This analysis works when two or more video-sensor nodes are sensing the same area of interest, the algorithm implemented helps sending only the different shots to the sink node in order to prevent sending all the shots which is costly in terms of energy and bandwidth.

4.2 Camera's overlapping filtering

We introduced, in the above sections, the functionalities of our similarity function. This function, when applied at the coordinator level, selects some video shots to be sent to the sink. To select a video instead of another one, the similarity function between the two must exceed a given threshold. A naive solution to find all similar shots is to compare each pair of shots. This method is obviously prohibitively expensive for video sensor networks, as the total number of comparisons is extremely high. We apply a geometric condition on the sensor nodes to select the appropriate comparison to be done and to reduce data latency.

This geometric condition is a combination of the angle condition between the FOVs of the nodes and the ratio of the overlapped area between them.

4.2.1 The angle condition

The angle between two neighbouring sensor nodes is defined as the angle between the vectors of their FOVs. Our idea is that if a wide angle is established between two sensor nodes FOVs, these two nodes can not take part in the similarity comparison function at the coordinator level. In this case, they are not sensing the same area of interest. A shot from two different perspectives can be widely different. To be able to define two sensor nodes as candidates for the similarity function, the angle between their FOVs must not surpass a certain angle threshold. In order to determine the angle between the two vectors ($\mathbf{V}$ and $\mathbf{V}'$) of the sensor nodes S and S' respectively as shown in Fig. 4, the scalar product method between those sensor nodes has been proposed. Both sensor nodes having the same dimensions (angle, FOV, energy ressources,...), so both vectors $\mathbf{V}$ and $\mathbf{V}'$ in Fig. 4 have the same length l. The scalar product can be calculated in two formats. The first one according to their coordinates (x and y) where $\mathbf{V} = (X_V, Y_V)$ and $\mathbf{V}' = (X_{V'}, Y_{V'})$:

$$\mathbf{V}.\mathbf{V}' = X_V \times X_{V'} + Y_V \times Y_{V'} \tag{14}$$

The second format is given according to the length of each vector and to the angle between both, as follows:

$$\mathbf{V}.\mathbf{V}' = l^2 \times \cos(\mathbf{V}, \mathbf{V}') \tag{15}$$

Where $l = \|\overrightarrow{V'}\| = \|\overrightarrow{V}\|$.

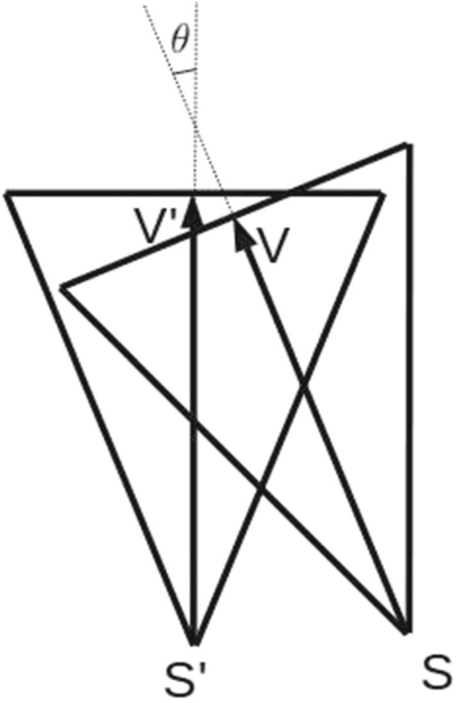


Fig. 4 Two overlapping sensor nodes S and S'

Below we define the equation where the angle θ between the two vectors can be calculated according to both formats of the scalar product :

$$\theta = \arccos((X_V \times X_{V'} + Y_V \times Y_{V'})/l^2) \tag{16}$$

e.g, if an angle threshold th_{angle} is defined as 30 degrees, the angle between $\mathbf{V}$ and $\mathbf{V}'$ must remain less than 30 degrees so the two sensor nodes S and S' can proceed to the next step (the two points strategy), to be able at the end to take part in the similarity function process at the coordinator level.

4.2.2 The two points condition

Inspired from [18] we present below the two points condition for overlapping filtering. A node S' satisfies the two points condition with another node S if g (the center of gravity of abc) and any other point between a, b and c from S's FOV, belong together to the FOV of S as shown in Fig. 3. S_1, S_2 and S_3 satisfy this condition seperately with S. In this scenario each sensor node can be a candidate alongside S to apply the similarity function between them.
Our method is used to chose the candidates that can take part in the comparison process at the coordinator level. Two camera-sensor nodes S_1 and S_2 are chosen as candidates if S_2 and S_1 satisfy together the angle and the two points conditions as shown in Algorithm 2. After chosing the candidates cameras, two cases are taken into consideration, the low similarity process and the high similarity process.

Algorithm 2 Candidate Selection Algorithm CSA

```
Set $S_1$ and $S_2$ two sensor-nodes
Set $ABC$ and $DEF$ the two FOV triangles for $S_1$ and $S_2$
Set $G_1$ and $G_2$ the two Gravity centers for $ABC$ and $DEF$
Set $V_1$ amd $V_2$ the two ligne of sight vectors for $S_1$ and $S_2$
Set $\alpha_{th}$ the angle threshold required
while $Energy > 0$ do
    if $angle V_1, V_2 < \alpha_{th}$ then
        Go to the 2 points condition
        if (A,G)$\in$(DEF) $OR$ (B,G)$\in$(DEF) $OR$ (C,G)$\in$(DEF) then
            $S_1$ and $S_2$ are candidates
            RUN $SSA$ Algorithm
        end if
    end if
end while
```

Definition 3 (**Low Similarity**) When the similarity between both compared video shots does not surpass the predefined similarity threshold percentage β_{sim} between shots, the coordinator works normally and sends both shots to the sink without any modification after each period, assuming that the similarity process is computed between both shots (all the frames sent from both sensor nodes take part in this similarity process) on a complete period and each period only represents one shot composed of several frames.

Definition 4 (**High Similarity**) If the similarity between those shots surpasses the threshold, in this case the coordinator must chose one of these two shots to be sent to the sink node. The coordinator selects the video shot where there are more variations within the shot, in other words, where the number of critical frames is greater as shown in Algorithm 3.

Algorithm 3 Shot Selection Algorithm SSA

```
Set $Shot_1$ and $Shot_2$
Set $sim$ the overall similarity between two frames
Set $sim_c$ the overall similarity between two shots
Set $\beta_{sim}$ the threshold similarity between two shots
Set $Col\_fact_c$, $Edge\_fact_c$ and $Mot\_fact_c$
Set $CR\_F_1$ the number of critical frames in $shot_1$
Set $CR\_F_2$ the number of critical frames in $shot_2$
while $Energy > 0$ do
    for each period do
        Run the LDS
    end for
    Compute $Col\_sim_c$
    Generate $Edge\_sim_c$
    Generate $Mot\_sim_c$
    Compute similarity $sim_c$
    $sim_c = Col\_sim_c \times Col\_fact_c + edge\_sim_c \times Edge\_fact_c + Mot\_sim_c \times Mot\_fact_c$
    if $sim_c > \beta_{sim}$ then
        Run the LDS on sensor node 1 to Generate $CR\_F_1$
        Run the LDS on sensor node 2 to Generate $CR\_F_2$
        if $CR\_F_1 > CR\_F_2$ then
            Send $shot_1$
        else
            Send $shot_2$
        end if
    end if
end while
```

Definition 5 (**similarity threshold percentage β_{sim} between shots**) This similarity between shots from overlapping sensor nodes, is the aggregation of all the similarities between the frames of these two shots, it can vary according to the application. For example and for military reasons, β_{sim} can reach 100% to be sure that the system does not miss any information.

4.3 Shot selection algorithm

In this section we discuss the SSA (Shot Selection Algorithm): After choosing the 2 candidates that meet the overlapping condition, this algorithm is implemented at the coordinator level to compare received frames from different sensor nodes sensing the same area of interest. This comparison is based on a similarity function that consists of edge, color and motion similarities as follows:

4.3.1 Motion similarity

To evaluate the motion content in a shot, we use a function related to the color similarity function by generating the mean of the sum of the inverse of the color similarity for each frame of a complete shot (period). Inspired from [23] and based on the color similarity function from $MASRA$ algorithm, this motion content mot_u of a shot u is computed and normalized as follows:

$$mot_u = \frac{1}{b-a} \sum_{f=a}^{b-1} (1 - Col\_sim(f, f+1)) \tag{17}$$

Where $mot_u \in [0, 1]$, a and b are the first and last frames sent from the sensor node to the coordinator in a period respectively and f, $f+1$ the two frames from $Shot_u$ which are sent by a sensor node.

The motion similarity between two shots mot_sim associated to two shots $Shot_u$ and $Shot_v$ from two different sensor nodes is defined as follows:

$$mot\_sim = 1 - |mot_u - mot_v| \tag{18}$$

In the last equation $mot\_sim \in [0, 1]$, if closer to 1 it marks that the two shots are similar in motion, an when this value is close to 0, the two shots are motionly different. In our approach we consider that the cameras in sensor nodes are fixed and not rotatable. Hence, the motion content value of the shots is much higher when an event is detected. Therefore, it is important to use this motion content in shots similarity estimation.

4.4 Shots similarity estimation

As explained below, shots sent from neighboring nodes to the same coordinator often have similar visual (color and edge) and/or action (motion) contents. Usually, in WVSNs, the motion content of shots depends on the event detection in the zone of interest. Therefore, when no event is detected the visual correlation between shots from candidates video-sensor nodes becomes higher. In our paper, we compute the similarities between shots as a function of their visual and motion content features. The color and edge similarities comparing two shots at the coordinator level are equal to their means all over the period, to be able to add them to the motion similarity at the end of each period. A solution for the synchronization problem is given later in this paper. The similarity between shots from different sensor nodes after each period is represented as follows:
we consider:

$cf = Col\_fact_c; cs = Col\_sim_c$
$ef = Edge\_fact_c; es = Edge\_sim_c$
$mf = Mot\_fact_c; ms = Mot\_sim_c$

$$SIM = (cf \times cs) + (ef \times es) + (mf \times ms) \tag{19}$$

Where Col_fact_c, $Edge\_fact_c$ and Mot_fact_c are considered as weights for color, edge and motion similarities on the coordinator level respectively, such that:
$Col\_fact_c + Edge\_fact_c + Mot\_fact_c = 1$.

In this approach, if two shots have similar motion contents, their Mot_sim_c function have a higher value. Note that Col_sim_c, $Edge\_sim_c$ and Mot_sim_c are in the range of [0,1].

4.4.1 Different frame rates solution

In this scenario, a synchronization problem is faced when two candidates sensor nodes $S1$ and $S2$ have two different frame rates FR_1 and FR_2 respectively, or when different critical scenes are sensed on each sensor node. At this point, the similarity process at the coordinator level can be broken, e.g, at time $t = 1$, $S2$ sends a frame to the coordinator but $S1$ does not send a frame, due to a criticality difference or to a frame rate difference between sensor nodes. To solve this problem, the comparison must take place between the frame received from $S1$ and the last frame received from $S2$ (if $S2$ did not send a frame at the same time) and vice versa. E.g, at time $t = 1$, $S1$ and $S2$ send two frames f_{11} and f_{21} to the coordinator respectively. At time $t = 2$, $S2$ sends a frame f_{22} to the coordinator but $S1$ does not send a frame. The comparison process continues by comparing Frame f_{22} with the last frame sent from $S1$ which is f_{11}. In other words, this can be a solution because a sensor node does not send a new frame to the coordinator when there is no new event in the scene. In this case, we consider the last frame sent by a sensor node as the actual frame of that sensor node.

5 Experiments

In this section, several experiments have been conducted to validate our approach at the sensor node and the coordinator levels, aiming to minimize the energy consumption and bandwidth usage by reducing the number of data (sensed and transmitted) all over the network. We compare our approach with Jiang et al. [12]. We have used a Matlab based simulator in our experiments. First of all, we introduce a scenario as shown in Fig. 5 where 6 video-sensor nodes S1,S2,S3,S4,S5,S6 are deployed to monitor the same area of interest from different perspectives. The main purpose in our work is to send to the coordinator the frames that represent the critical situations. The coordinator reacts accordingly. We have used 6 Microsoft LifeCam VX-800 cameras to film a short video of 600 seconds, each camera is connected to a laptop to do the processing via a Matlab simulator. In our study an intrusion has been detected in the sensor-nodes at the following time-intervals:

S1: 40 seconds from 75 to 115.

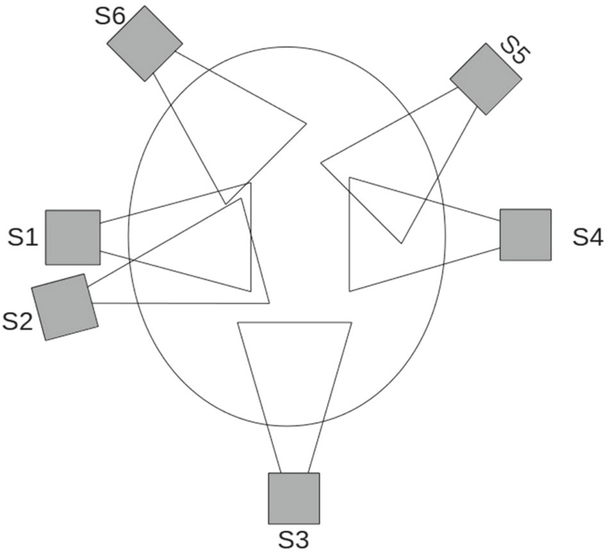


Fig. 5 The setup of the video sensor nodes

Table 2 Weights of small similarities at both levels

Level	Col_fact_c	$Edge\_fact_c$	Mot_fact_c
Sensor node	0.5	0.5	0
Coordinator	0.25	0.25	0.5

S2: 40 seconds from 80 to 120.
S3: 200 seconds from 120 to 320
S4: 160 seconds from 300 to 460
S5: 40 seconds from 450 to 490
S6: 0 seconds. We have run our $MASRA$ and SSA algorithms for 600 periods, each period consists of 1 second, with a frame rate equal to 30 frames per second. The frame rate in each sensor node changes independantly according to the number of critical frames related to its sensor node. In each period, every sensor node senses a certain number of frames according to the assigned frame rate. The minimum frame rate is set to $FR = 1$ frame per period. We consider the initial and maximum frame rate $FR = 15$ frames per period. In this case the sensor node senses 15 frames from the 30 ones in the period.

As for the parameters at the sensor node level we used a color factor and edge factor equal to 50%. At the coordinator level : we used a color factor and edge factor equal to 25% each and the motion factor is equal to 50%. As shown in Table 2 the motion factor Mot_fact_c has a higher weight at the coordinator level. A frame received from a sensor node is known to be a critical frame, so an information about the motion is more important at the coordinator level to be sent to the sink.

Then, we implemented the PPSS approach in [12], and we did run the same video sequence. This algorithm adopts the normal law of probability and the kinematics rules. Its role is to schedule the monitoring time of the sensor-node depending on the trajectory of the intrusion and the time needed to reach its FOV and the sensor-node sends all the sensed frames to the coordinator while the intrusion is in its FOV, and then it goes back to the sleep mode. But after several experiments, this approach tends to lose information up to 15% due to probability errors. This loss of data in PPSS is shown in Figs. 6 and 7 for sensor S1 in our scenario when the intrusion passes by its FOV.

5.1 The sensor node level

5.1.1 Number of frames

The biggest challenge in WSN is the energy consumption due to the limited resources of the sensor nodes and to the big number of frames on the network. When no specified or adapted frame rate is implemented, the amount of sensed frames remains at 30 for each period. In terms of energy consumption and bandwidth usage, sending all the frames is costly while a lot of frames are identical and do not represent any criticality. Sending frames with a time difference inferior to 0.03 seconds in a video surveillance does not represent any additional information. For this reason, we set the initial and maximal frame rate to $FR = 15$ frames sensed per period. The $MASRA$ algorithm is implemented on every video-sensor node to reduce the number of frames sensed and sent to the coordinator. For every sensor node, the frame rate is adapted after two periods where $P = 1$ second. Every sensor node sends the first frame of each period. For sensor node $S1$, as seen in Fig. 7, the $MASRA$ algorithm only

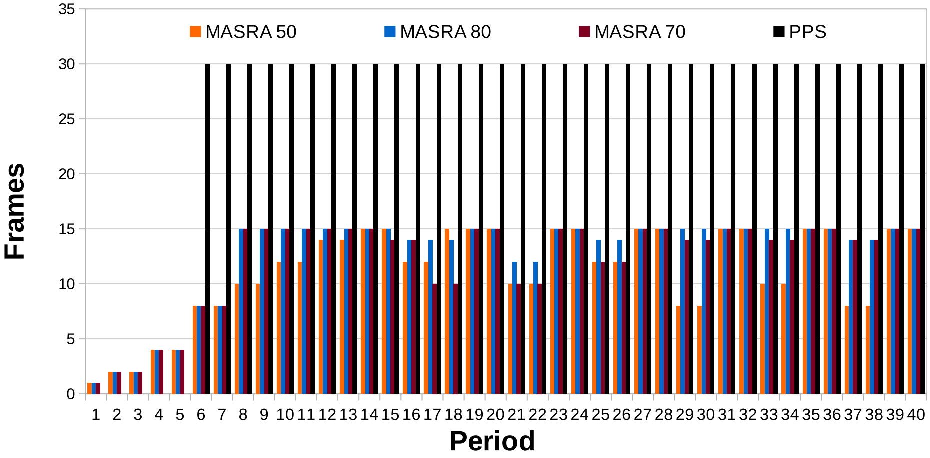


Fig. 6 Difference between MASRA and PPSS on the sensing phase

sends the critical frames to the coordinator according to a predefined threshold of similarity as explained in the upper sections, this threshold varies from 50% to 70% to 80%. In the latter stages we chose a threshold equal to 70% as a mean to all other values. The number of frames sent in each period is the parameter that influences the frame rate. The frame rate variation seen in Fig. 6 validates our frame rate adaptation method in the active mode of sensor S1, when an intrusion is detected.

In Fig. 7, we can see the number of critical frames sent to the coordinator via $S1$, this variation in the number of critical frames per period is proportional to the adaptation of the frame rate. Figures 6 and 7 present a slight difference when the threshold changes from 50 to 70 to 80. Thus, the choice of 70% is validated. As seen in Tables 3 and 4 for S1 and in Tables 5 and 6 for S3, adapting the frame rate reduces the sent data by more than 90%. Then, applying our similarity function causes the degradation of the number of sent frames

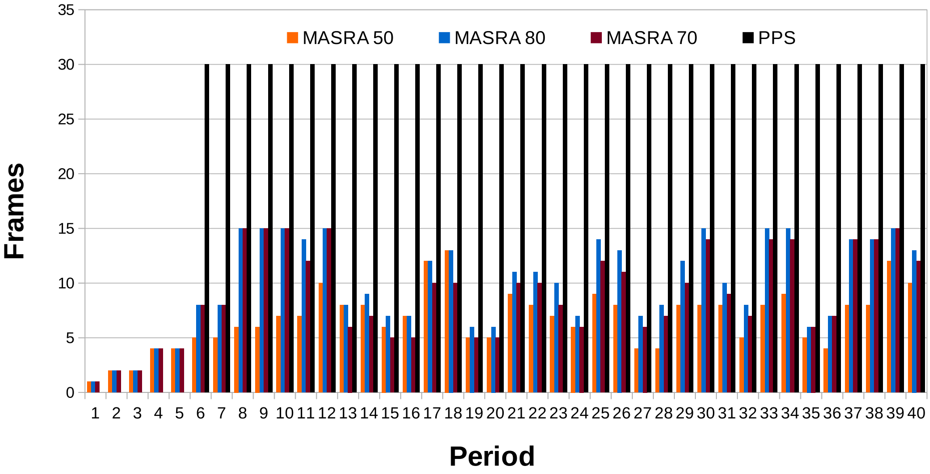


Fig. 7 Difference between MASRA and PPSS on the transmission phase

Table 3 The difference in terms of number of frames for S1 over 40s

Nb of periods	All frames	Sampled frames	Critical frames
40	1200	490	360

by 94% from 14700 frames to 818 frames. Reducing the number of sensed frames via the adaptation of the frame rate, and reducing the number of frames sent to the coordinator by using our similarity function at the sensor node level prove that our algorithm reduces the number of frames in terms of sensing and transmitting as detailed in Tables 7, 8 and 9 for all the network.

By comparing these numbers to the number of frames in Tables 10, 11, 12 and 13, while applying PPSS algorithm, we can conclude that the efficiency of our algorithm for the sensing and transmission process surpasses the PPSS algorithm. And this gain grows furthermore when the time interval of the active mode of the sensor grows, as shown for sensor-node S3. For probability reasons, the first sequence of frames for every sensor is lost in PPSS, once the intrusion opts in the FOV of the sensor node. Tables 7, 8 and 9 show the efficiency of our approach sensor by sensor and on the overall network regarding the number of sensed and transmitted frames.

5.1.2 Bandwidth consumption

The bottleneck issue is a problem caused by the limited ressources in terms of bandwidth capacity and by the huge number of frames sent all over the network. For the same algorithm ($MASRA$) as we can see in Table 14 for the network, the size of the sent frames varies and is by far reduced. At the sensor node level, the frame rate adaptation and the similarity function applied are responsible for this reduction by only sending the critical frames to the coordinator which reduces the size of the total number of frames sent within a period as shown in Table 14. The size of the video filmed in total is equal to $300MB$, this number is cut by 90% to reach $19MB$ when we send all the frames by adapting the frame rate, and from $300MB$ to $15MB$ if we implement our algorithm with all its functionalities as mentioned in Table 14.

Sending $15MB$ in 490 seconds is equivalent to having a bit rate equal to $31KB/s$ which is a very small bit rate which will avoid causing a bottleneck problem even if we have a big number of video-sensor nodes in the network. In this case a capacity of $100MB$ can serve more than 2,000 sensor-nodes at the same time.

In [12], they reduce the bandwidth usage, but depending on the similarity function presented in our paper, the bandwidth reduction is better by 5% from 90% in PPSS to 95% in MASRA algorithm as mentioned in Tables 14 and 15.

5.2 The coordinator level

The SSA algorithm is implemented on the coordinator. As seen in Fig. 5, and based on angle and position conditions, only video-sensor nodes $S1$ and $S2$ satisfy the overlapping method

Table 4 The difference in terms of number of frames for S1 over 490s

Nb of periods	All frames	Sampled frames	Critical frames
490	14700	938	818

Table 5 The difference in terms of number of frames for S3 over 200s

Nb of periods	All frames	Sampled frames	Critical frames
200	6000	2940	2055

Table 6 The difference in terms of number of frames for S3 over 490s

Nb of periods	All frames	Sampled frames	Critical frames
490	14700	3245	2360

Table 7 Sensor by sensor evaluation in terms of number of frames in active and passive modes for MASRA algorithm

	Passive mode	Passive mode	Active mode	Active mode
Sensor (active time)	Sensed	Transmitted	Sensed	Transmitted
S1 (40s)	448	448	490	360
S2 (40s)	448	448	520	380
S3 (200s)	305	305	2940	2055
S4 (160s)	333	333	2340	1710
S5 (40s)	448	448	500	400
S6 (0s)	490	490	0	0
Total	2472	2472	6790	4905

Table 8 Sensor by Sensor evaluation in terms of number of frames in active and passive modes for PPSS Method

	Passive mode	Passive mode	Active mode	Active mode
Sensor (active time)	Sensed	Transmitted	Sensed	Transmitted
S1 (40s)	0	0	1080	1080
S2 (40s)	0	0	1102	1102
S3 (200s)	0	0	5134	5134
S4 (160s)	0	0	4120	4120
S5 (40s)	0	0	1065	1065
S6 (0s)	0	0	0	0
Total	0	0	12501	12501

Table 9 Comparison between MASRA and PPSS in terms of number of frames on the overall Network

	MASRA Sensed	MASRA Transmitted	PPSS Sensed	PPSS Transmitted
Total frames	9262	7377	12501	12501

Table 10 The difference in terms of number of frames for S1 over 40s PPSS

Nb of periods	All frames	Sampled frames	Critical frames
40	1200	1080	1080

Table 11 The difference in terms of number of frames for S1 over 490s PPSS

Nb of periods	All frames	Sampled frames	Critical frames
490	14700	1080	1080

Table 12 The difference in terms of number of frames for S3 over 200s PPSS

Nb of periods	All frames	Sampled frames	Critical frames
200	6000	5134	5134

geometric conditions so their frames can be compared at the coordinator level by the SSA algorithm. The coordinator will send the frame of the more critical video-sensor node to the sink with respect to a predefined similarity metric threshold β_{sim}. In our experiments $S1$ and $S2$ send 938 and 968 frames to the coordinator respectively, which gives $S2$ the edge to be the more critical node. In this case when a comparison takes place between two frames, if the similarity exceeds the predefined β_{sim}, the frame from $S2$ is sent to the sink and the other one from $S1$ will be rejected. Otherwise both frames are sent to the sink.

In our experiments, the coordinator receives a sum of 1906 frames from $S1$ and $S2$ combined. By modifying the threshold β_{sim} from 50% to 80%, the number of frames sent to the sink changes. The changes are recognised, the number of sent frames and β_{sim} are proportional. Table 16 summerizes the coordinator behavior by showing the percentages of reduction that degrades from 48% for β_{sim}=50 to reach zero when β_{sim}=80. For β_{sim}=50, the 48% reduction in terms of number of frames sent from the coordinator to the sink added to the 90% reduction at the sensor node level increases the lifetime of the network by reducing the number of frames and the bandwidth usage due to transmission reduction on both levels. As for PPSS, they do send every frame received by the coordinator to the sink node, disregarding the correlation of several sensor-nodes and the similarity of their frames.

As shown in Tables 16 and 17, our algorithm on the coordinator level helps to reduce furthermore the number of frames sent to the sink by more than 32% if $\beta_{sim} < 60\%$ for the correlated sensor-nodes.

6 Energy consumption study

In this section, our energy consumption comparison study is based on the energy model proposed in [14]. The consumed energy as in [14] is divided into two parts, the radio energy for the transmission of the data on the radio and the computational energy for the in-node processing. as shown in the equation below:

$$E = E_{radio} + E_{comp} \tag{20}$$

Table 18 shows the different parameters to compute the energy consumption while considering:
I_{TX} and I_{RX} the electric power needed for sending and receiving by the radio respectively.
T_{TX} and T_{RX} the corresponding operating time over 1 byte.
V be the constant voltage supply throughout the transmission.

$$E_{radio}(k) = k.I_{TX}.V.T_{TX} + k.I_{RX}.V.T_{RX} \tag{21}$$

Table 13 The difference in terms of number of frames for S3 over 490s PPSS

Nb of periods	All frames	Sampled frames	Critical frames
490	14700	5134	5134

Table 14 The ultimate bandwidth total reduction MASRA

Nb of periods	All frames	Sampled frames	Critical frames
490	300 MB	19 MB	15 MB

Table 15 The ultimate bandwidth total reduction PPSS

Nb of periods	All frames	Sampled frames	Critical frames
490	300 MB	29 MB	29 MB

Table 16 The coordinator behavior

β_{sim}	Received Fr	Sent Fr	Sent(%)	Reduction(%)
50	1906	1000	52%	48%
60	1906	1296	68%	32%
70	1906	1640	86%	14%
80	1906	1906	100%	0%

Table 17 The Coordinator Behavior PPSS

β_{sim}	Received Fr	Sent Fr	Sent(%)	Reduction(%)
50	1906	1906	100%	0%
60	1906	1906	100%	0%
70	1906	1906	100%	0%
80	1906	1906	100%	0%

Table 18 Parameters of the energy model

Parameter	Value
I_{TX}	17.4 mA
I_{RX}	19.7 mA
T_{TX}	3.2×10^{-5} s
T_{RX}	3.2×10^{-5} s
V	3.3 V
I_{cpu}	31 mA
f_{cpu}	48 MHz
ϵ_{add}	2.13 nJ
ϵ_{mul}	6.39 nJ
ϵ_{cmp}	2.13 nJ
ϵ_{sht}	4.26 nJ

Table 19 Energy consumption comparison for MASRA and PPSS

Energy	MASRA	PPSS
E_{Comp}	49 J	0.1 J
E_{radio}	28.9 J	276.13 J
E_{total}	77.9 J	276.23 J

Taking into account that k is the number of bytes sent from a specific sender to a specific receiver. For the computational energy consumption:
$\epsilon_{add}, \epsilon_{mul}, \epsilon_{cmp}, \epsilon_{sht}$ are the basic operations (shift,addition,comparison,multiplication, etc...), Table 18 shows the required energy for each operation. To compute this energy consumption, we only needs to count the number of each basic operation in the algorithm:

$$E_{comp} = N_{add} \times \epsilon_{add} + N_{sht} \times \epsilon_{sht} + N_{mul} \times \epsilon_{mul} + N_{cmp} \times \epsilon_{cmp} \tag{22}$$

In order to compare both approaches, we calculate the energy consumption of both the processing and the transmission tasks of a wireless sensor node equipped with a CC2420 radio transceiver and an ARM7TDMI microprocessor. Table 18 displays the parameters that are used in the calculations and which are found in the data sheets of the node's components [14].

6.1 Sensor node level

In our experiments, when running the MASRA algorithm, 9262 frames were sensed and compared using the similaritiy function. For a 640 × 480 frame size, 307200 pixels exist in each frame. Each similarity takes into account all the pixels in every frame. The MASRA algorithm consists of 2 additions, 2 multiplications and 1 comparison. We can compute the computational energy for E_{comp} for 9262 similarities as follows:

$$E_{comp} = 9262 \times 640 \times 480 \times (2 \times \epsilon_{add} + 2 \times \epsilon_{mul}) + 9262 \times \epsilon_{cmp} \tag{23}$$

In this case, $E_{comp,masra}$=49 J.

If we apply PPSS, $E_{comp,ppss} = 0.1$ J. To move on to the transmission phase, where our network for the MASRA algorithm transmits 7377 frames = 15 MB, comparing to the 12501 frames = 29 MB for PPSS. In the MASRA algorithm, the sensors only send the frames to the coordinator, but in PPSS when a sensor-node detects a frame, the node sends a message to its neighboors containing several information such as the id of the sensor, the position of the intrusion .... Adding to the 29 MB of frames that has been sent on the network, the sensor-nodes in PPSS in our experiments send to each other 25600 messages of 4 KB for each message. which means 100 MB to be added in the received data.

$$E_{radio,masra} = 15 \times 1024 \times 1024 \times 17.4 \times 10^{-3} \times 3.3 \times 3.2 \times 10^{-5} = 28.9J \tag{24}$$

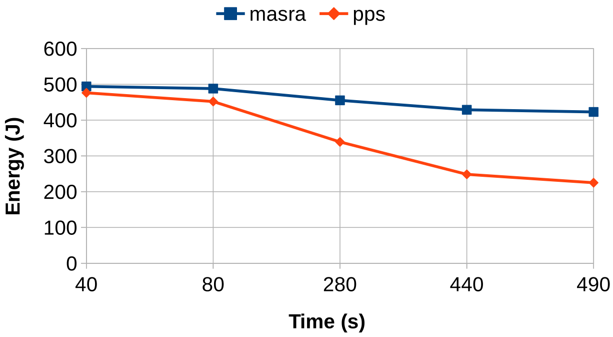


Fig. 8 Energy consumption comparison for MASRA and PPSS

$$\begin{aligned}E_{radio,ppss} &= 29 \times 1024 \times 1024 \times 17.4 \times 10^{-3} \times 3.3 \times 3.2 \times 10^{-5}\\ &\quad +100 \times 1024 \times 1024 \times 19.7 \times 10^{-3} \times 3.3 \times 3.2 \times 10^{-5}\\ &= 276.13J\end{aligned}$$

To compute the total energy consumption consumed by the network on the sensor node level, the energy consumptions related to the computation and to the transmission must be added to each other.

$$E_{masra} = E_{comp,masra} + E_{radio,masra} = 49 + 28.9 = 77.9J \tag{25}$$

$$E_{ppss} = E_{comp,ppss} + E_{radio,ppss} = 0.1 + 276.13 = 276.23J \tag{26}$$

As shown in Table 19, while comparing with PPSS, our algorithm consumes more energy on the computational level, but reduces much more energy on the transmission level. Figure 8 compares both approaches in terms of energy consumption over time on the overall network while considering a start energy of 500 J for the network. The gain of our approach is positive, PPSS and as shown in Fig. 8 consumes more energy than our approach in our experiments.

7 Conclusion

In this paper, we introduced an adaptive frame rate algorithm with a similarity detection function for wireless video sensor nodes. Also, a Shot Selection algorithm is implemented at the coordinator level. The proposed work allows a dynamic frame rate control of each video-sensor node. The conducted experiments show that the proposed algorithms did not miss any event in the recorded video sequence. Thus, the algorithms send the minimum required frames to the sink node by using a similarity detection function at the sensor node and coordinator levels. The selected frames are transmitted by the sensor nodes to the coordinator and by the coordinator to the sink without missing any required information. The results show that the size of the transmitted data in each period is reduced and the energy consumption is decreased, thus, preventing any bottleneck problem regarding the bandwidth limitation issue.

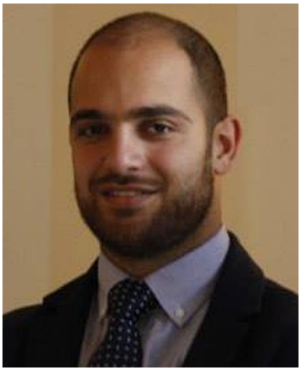
Comparing our approach with PPSS algorithm in terms of data reduction and energy consumption, helps us to find out that our algorithm outperforms PPSS, and reduces the number of data for more than 40% than PPSS. Thus, PPSS consumes 4 times more energy than our approach on the sensor node level. As future works, first of all we need to do some real experimentations on real sensor-nodes in the near future. Then, and after examining the amount of energy needed to do the processing, we aim to extend this work by including a study which further reduces the computational energy consumption at the sensor node level.

Acknowledgements This project has been performed in cooperation with the Labex ACTION program (contract ANR-11-LABX-0001-01) and this work is partially funded with support from the National Council for Scientific Research in Lebanon CNRS-L, the Hubert Curien CEDRE programme n°40283YK, and the Agence Universitaire de la Francophonie AUF-PCSI programme.

References

1. Akyildiz IF, Melodia T, Chowdhury KR (2008) Wireless multimedia sensor networks: applications and testbeds. Proc IEEE 96(10):1588–1605
2. Akyildiz IF, Melodia T, Chowdhury KR, Kaushik R (2007) A survey on wireless multimedia sensor networks. Comput Netw 51(4):921–960
3. Alippi C, Anastasi G, Francesco M, Roveri M (2010) An adaptive sampling algorithm for effective energy management in wireless sensor networks with energy-hungry sensors. IEEE Trans Inst Meas 59(2):335–344
4. Alaei M, Barcelo-Ordinas JM (2010) A method for clustering and cooperation in wireless multimedia sensor networks. Patent Appl Publ 10:3145–3169
5. Alaei M, Barcelo-Ordinas JM (2010) A method for clustering and cooperation in wireless multimedia sensor networks. Sensors 10(1):3145–3169
6. Bahi JM, Makhoul A, Medlej M (2014) An optimized in-network aggregation scheme for data collection in periodic sensor networks. ADHOC-NOW 11:153–166
7. Benzerbadj A, Kechar B (2013) Redundancy and criticality based scheduling in wireless video sensor networks for monitoring critical areas. Procedia Comput Sci 21:234–241
8. Canny J (1986) A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell PAMI-8(6):679–698
9. Choi J, Han S, Kim S, Chang S, Yoon E (2007) A spatial-temporal multiresolution cmos image sensor with adaptive frame rates for tracking the moving objects in region-of-interest and suppressing motion blur. JSSC 42(12)
10. Dai R, Akyildiz IF (2009) A spatial correlation model for visual information in wireless multimedia sensor networks. IEEE Trans Multimedia 11(6):1148–1159
11. Jbeily T, Alkubeily M, Hatem I (2015) A new symmetric-object oriented approach for motion estimation in wireless multimedia sensor networks. IJSR 4(11)
12. Jiang B, Ravindran B, Cho H (2013) Probability-based prediction and sleep scheduling for energy-efficient target tracking in sensor networks. IEEE Trans Mobile Comput 12(4):735–747
13. Kyrkou C, Laoudias C, Theocharides T, Panayiotou CG, Polycarpou M (2016) Adaptive energy-oriented multitask allocation in smart camera networks. IEEE Embed Syst Lett 8(2):37–40
14. Liang Y, Peng W (2010) Minimizing energy consumptions in wireless sensor networks via two-modal transmission. SIGCOMM Comput Commun Rev 40(1):12–18
15. Luo W, Lu Q, Xiao Q (2012) Distributed collaborative camera actuation scheme based on sensing-region management for wireless multimedia sensor networks. Distrib Sens Netw 12:1–14
16. Newell A, Akkaya K (2011) Distributed collaborative camera actuation for redundant data elimination in wireless multimedia sensor networks. Ad Hoc Netw 45(4)
17. Nguyen HT, Worring M, Dev A (2000) Detection of moving objects in video using a robust motion similarity measure. IEEE Trans Image Processing, pp 1–4
18. Pham C, Makhoul A, Saadi R (2011) Risk-based adaptive scheduling in randomly deployed video sensor networks for critical surveillance applications. JNCA 34(2)
19. Politis I, Tsagkaropoulos M, Kotsopoulos S (2008) Optimizing video transmission over wireless multimedia sensor networks. IEEE GLOBECOM, pp 1–6
20. Prieto SM, Allen AR (2003) A similarity metric for edge images. IEEE Trans Pattern Anal Mach Intell 25(10):1265–1273
21. Priyadarshini SB, Acharya BM, Das DS (2013) Redundant data elimination and optimum camera actuation in wireless multimedia sensor network (wmsn). IJERT 2(6)
22. Qin Z, Wang L, Ma C, Xu J, Lu B (2013) An overlapping clustering approach for routing in wireless sensor networks. IJDSN 2013:867385
23. Rasheed Z, Shah M (2005) Detection and representation of scenes in videos. IEEE Trans Multimedia 7:1097–1105
24. Sahasrabudhe N, West JE, Machiraju R, Janus M (1999) Structured spatial domain image and data comparison metrics. pp 97–515
25. Sarisaray-Boluk P, Akkaya K (2015) Performance comparison of data reduction techniques for wireless multimedia sensor network applications. Hindawi Publ Corp 15:1–15
26. Shahab MB, Usman MA, Shin SY (2017) Bandwidth adaptation by squeezing idle traffic in browsers an active window detection based approach for next generation networks. IEEE Commun Lett 21(2):310–313
27. Spagnolo P, Orazio T, Leo M, Distante A (2006) Moving object segmentation by background subtraction and temporal analysis. Image Vis Comput 24(5):411–423

28. Stewart R, Trahan K, Chesavage D, Casey S, Rome M, Kokinakes C (2003) Surveillance system and method with adaptive frame rate. Patent Appl Publ 21:234–241
29. Stricker MA, Orengo M (1995) Similarity of color images. Proc SPIE 2420:381–392
30. Usman MR, Usman MA, Shin S (2015) Subjective quality assessment for impaired videos with varying spatial and temporal information 9:1574–1579,07
31. Usman MA, Usman MR, Shin SY (2016) A no reference method for detection of dropped video frames in live video streaming. pp 839–844
32. Usman MA, Usman MR, Shin SY (2018) An intrusion oriented heuristic for efficient resource management in end-to-end wireless video surveillance systems. pp 1–6
33. Yao Y, Giannakis GB (2005) Energy-efficient scheduling for wireless sensor networks. IEEE Trans Commun 53(8):1–10
34. Zeng X, Hu W, Li W, Zhang X, Xu B (2008) Keyframe extraction using dominant-set clustering. IEEE Int Conf Multimedia Expo 1(1):1285–1288



Christian Salim is currently a Ph.D. student at University of Franche-Comté (UFC), France co-directed by the Antonine University, Lebanon. He received the M.S degree in computer science and telecommunication engineering in 2015 from the Antonine University, Lebanon. He reveived the M.S. degree in Distributed and Mobile Computing in 2015 from University of Franche-Comté (UFC), France. His research interests include wireless sensor networks, multisensor data fusion and data aggregation.

Abdallah Makhoul received the PhD degree in computer science from the University of Franche-Comté (UFC), France, in 2008. Since 2009, he has been an Associate Professor with the University of Franche-Comté. In 2006, he joined as associate professor the DISC department (department of computer science and complex systems) of FEMTO-ST, Belfort (France). His research focuses upon the following areas: Internet of things, e-health monitoring and real-time issues in wireless sensor networks. He has a record of about 70 scholarly publications. He has been a TPC chair and member of several networking conferences and workshops and a Reviewer for several international journals. He participated in several national and international research projects.

Rony Darazi (SMÍ6) received the M.S. degree in Computer and Telecommunications engineering from Antonine Univerity (UA), Lebanon in 2005, and the Ph.D. degree in engineering sciences from the Université catholique de Louvain (UCL), Louvain-la-Neuve, Belgium, in 2011. His PhD was entitled "Towards a combining scheme for compression and watermarking for 3D stereo images". He is currently an Associate Professor at UA. He was a Researcher in the ICTEAM Institute at UCL from 2006 until 2012, and is a Member of the TICKET Lab at UA since 2010. His research interests include information security and digital watermarking, digital 2D and 3D image processing, sensor networks, and e-health. Dr. Darazi is an IEEE senior member, he received a grant research project from the National Council for Scientific Research in Lebanon (CNRS-L) in 2016 and a research fund from the Francophone University Agency (AUF) in 2017. He co-chaired the International Conference on Applied Research in Computer Science & Engineering (ICARÍ5), sponsored by IEEE in 2015, and has been actively involved as a Reviewer in several conferences and journals. In 2009, he received the Best Paper Award, second prize by the Digital Watermarking Alliance and the IS&T/SPIE International Conference on Media Forensics and Security XII.

Raphaël Couturier is a professor in computer science in University Bourgogne Franche Comté, France (ICIP).