

## Image Processing Based Data Reduction Technique in WWSN for Smart Agriculture

Jana Koteich<sup>\*1</sup>, Christian Salim<sup>2</sup> and  
Nathalie Mitton<sup>1</sup>

the date of receipt and acceptance should be inserted later

**Abstract** Nowadays, to improve animal well being in livestock farming application, a wireless video sensor network (WWSN) can be deployed to early detect injury and monitor animals. They are composed of small embedded video and camera nodes that capture video frames periodically and send them to a specific node called a sink. Sending all the captured images to the sink consumes a lot of energy on every sensor and may cause a bottleneck at the sink level. Energy consumption and bandwidth limitation are two important challenges in WWSNs because of the limited energy resources of the nodes and the medium scarcity.

In this work, we introduce two mechanisms to reduce the overall number of frames sensed and sent to the sink. The first approach is applied on each sensor node, where the FRABID algorithm, a joint data reduction, and frame rate adaptation on sensing and transmission phases mechanism is introduced. This approach reduces the number of sensed frames based on a similarity method. The aim is to adapt the number of sensed frames based on the degree of difference between two consecutive sensed frames in each period. This adaptation technique maintains the accuracy of the video while capturing frames holding new information. This approach is validated through simulations using real data-sets from video sensors [1]. The results show that the amount of sensed data is reduced by more than 70% compared to a recent algorithm in [2] while guaranteeing the detection of all the critical events at the sensor node level.

The second approach exploits the Spatio-temporal correlation between neighboring nodes to reduce the number of captured frames. For that purpose, Synchronization with Frame Rate Adaptation SFRA algorithm is introduced where overlapping nodes capture frames in a synchronized fashion every  $N - 1$  periods, where  $N$  is the number of overlapping sensor nodes. The results show

---

<sup>1</sup>Inria, France

E-mail: jana.koteich@inria.fr; nathalie.mitton@inria.fr

<sup>2</sup>Junia, Computer Science and Mathematics, F-59000, Lille, France

E-mail: christian.salim@junia.com

more than 90% data reduction, surpassing other techniques in the literature at the level of the number of sensed frames by 20% at least.

**Keywords** Smart Agriculture; Spatio-temporal Correlation; Data Reduction; Data Prediction; WSN

## 1 Introduction

Nowadays, the smart agriculture domain faces a lot of challenges for better usage of its natural resources. However, the agriculture domain includes livestock farming. Understanding the wild animals' behavior would facilitate the means of protection of cattle in places beyond man's control. For this reason, wireless video sensor networks (WVSNs) are deployed in remote sites to monitor livestock that is exposed to threats from wild animals like jackals in South Africa. WVSNs process in real-time and retrieve multimedia data periodically to be sent to a sink.

Different architectures have been studied in the literature. Figure 1 represents the general architecture of a wireless sensor network, where the nodes capture frames from videos with a given frequency (frame rate) and wirelessly send them to the sink.

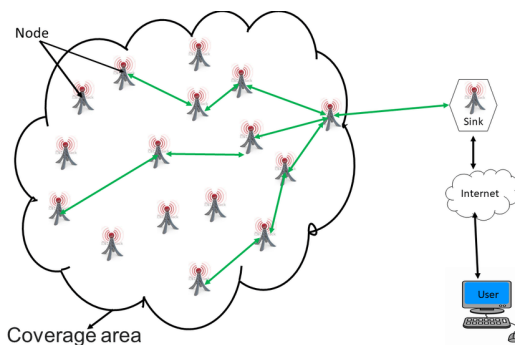


Fig. 1: General architecture of a wireless sensor network

In a WVSN system, nodes capture frames periodically. Due to this periodic cycle, network energy consumption gets highly stressed by the transmission of a huge amount of redundant and unnecessary data [3]. Therefore, a method that can minimize the amount of sensed data at the sensor-node level and transmitted data is required. Thus, to achieve data reduction on each sensor node in the overall system, there are three main phases to be studied: The sensing phase, processing techniques, and the transmission phase.

The main focus of our work is on the sensing and transmission phases. We introduce a Frame Rate Adaptation Based Image Difference algorithm (FRABID), which reduces the images sent in two steps. First at the sensing

level by reducing the number of frame capture, then at the transmission level, by selecting only a part of them to be sent. To reduce the number of captured frames, the nature of motion is predicted by comparing the first two sensed frames in each period.

To be able to set the new frame rate, we apply a comparison based on the L1 norm Euclidean distance. The sum of the absolute differences between two consecutive frames provides basic information on the nature of the motion (new event, slow motion, fast motion...) in a given scenario, then a new frame rate will be assigned to each period depending on the percentage of the difference between the frames. If the difference percentage is high, the frame rate will be set to its maximum. Then, to filter the captured frames and send only the important frames holding new information, the sensor node compares the sensed frame with the last sent frame using L1 norm simple Euclidean distance. If the difference exceeds a predefined threshold, the difference image will be sent to the sink, otherwise, the frame will not be sent and the sensor node will stop capturing frames for the current period. Our contribution bypassed other previous work by 1) assigning a delicate frame rate to each period on the sensor node and 2) reducing redundant sensed data by more than 90%.

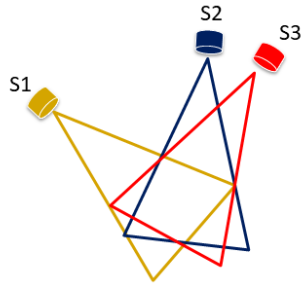


Fig. 2: Overlapping wireless video sensor nodes

We conducted simulations using Python and the results show the validity of our approach, by reducing the amount of sensed frames to more than 70% compared to the STAFRA algorithm in [2], and decreasing the redundancy by reducing the number of similar sensed frames to less than 10%, outperforming other approaches, while guaranteeing the capture of important events.

Our second contribution aims to reduce redundant data between neighboring nodes. Neighboring nodes are defined by their field of view (FoV). Overlapping FoVs in dense networks causes wasting power of the system because of redundant sensing of area [4]. Figure 2 represents three neighboring sensor nodes that have overlapping FoV. To achieve data reduction between overlapping sensors, we proposed a Synchronization with Frame Rate Adaptation (SFRA) algorithm. The main aim of the proposed synchronization method is energy conservation and prolonging network lifetime while preserving all needed information.

Different approaches have introduced strategies to detect neighboring nodes. After having all overlapping sensor nodes in a stable state (a stable state is defined when no motion is detected), the nodes will start synchronization. Each node will capture frames in a different time slot from its overlapping nodes with the minimum frame rate. The addition of the SFRA algorithm to other approaches presented in the literature for overlapping sensor nodes is the synchronization fashion, which reduces the number of sensed frames while still preserving almost the needed information. This method has shown more than 90% reduction of data compared to other methods and added 33% more data reduction on a sensor node when combined with the FRABID algorithm. In both algorithms (FRABID and SFRA) we are concerned with data reduction at the sensor-node level which affects proportionally energy consumption and bandwidth limitation. The remainder of this paper is structured as follows: Section 2 presents a general review of previous contributions presented in the literature. Section 3 describes the system model. Section 4 presents the algorithm for data reduction on each sensor. Section 5 illustrates the proposed algorithm for data reduction in overlapping sensor nodes. Section 6 experimental results. In Section 7, the results for combining both algorithms is illustrated, and finally, the conclusions are drawn in Section 8 with perspective on future work.

## 2 Background and Related Work

Different techniques and research work have been proposed in the literature to reduce energy consumption and data redundancy in Wireless Video Sensor Networks (WVSNs). In this section, we will browse some of these approaches while focusing on the data reduction at the sensing phase, transmission phase, and between overlapping sensor nodes at the application level.

Several research work for energy reduction has been proposed to decrease data redundancy: Scheduling methods [4–7], Data aggregation [8], Geometrical criteria [9–11], prediction techniques [12], frame rate adaptation [2, 13–16], the usage of edge and fog computing [17–19]. In [12], the authors used the kinematics functions to predict the next location of the intrusion in the area of interest in order to increase the frame rate adaptation of the targeted nodes, this approach comes as a complementary solution to our method to detect the position of the intrusion.

For data reduction at the transmission level, several approaches dealt with frame analysis to achieve this aim. A wireless video sensor-node captures a series of frames to form the video at the receiver side. A frame is a group of bytes represented as pixels sent over the network. Sending all the frames over the network will increase the energy consumption at the sensor node level and congestion problems on the limited bandwidth of the network. Thus the analysis helps send only the different frames to the sink that holds new information. For that, in [14] the authors used image comparison based on color and edge properties to find similarities between frames, to decide which

frame to send, where 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. The color property detects any change in the colors of the scene. While in [2] and [13] each frame is compared to the last frame sent to the sink using norm L2 simple euclidean distance similarity method to compare images. In our approach the adaptation replaces the above methods with L1 norm simple euclidean distance, which will reduce the processing time on each sensor-node.

For data reduction at the sensing phase, different approaches aimed to adapt the frame rate on the sensor node. In [2] and [14], the authors approach reduces the frame rate of each sensor node at the sensing phase according to the event happening in the zone of interest. In [14], the authors worked on 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, while in [2], the authors' approach tends to adapt the frame rate of a period based on the number of critical frames of the previous period. Results show that the second approach gives better results since it changes the frame rate earlier. Unfortunately, these studies manage to adapt the frame rate based on the criticality of the previous periods, so in a scenario with a frequent motion, the frame rate will always be high, since there are always critical frames. In our approach the frame rate is adapted according to the difference between the first two frames in each period, which is more like a prediction to the nature of motion in the current period, and so the frame rate is adapted according to the conditions of each period.

To further reduce the redundancy of captured data, the overlapping field of views (FoVs) of sensor nodes is exploited to achieve data reduction. Nodes can considerably prevent wasting power by avoiding redundant sensing, processing, or sending similar multimedia data. Thus, it prolongs network lifetime, particularly in dense networks that are usually deployed with a high number of low power, low resolution, and inexpensive multimedia nodes randomly [4]. Several approaches tried to solve the issue of data redundancy by taking into consideration overlapping sensor nodes. The authors in [2,13,14] used geometrical conditions to detect overlapping sensors. After detecting the overlapping sensor nodes, the authors in [13] defined a stable situation, where no motion is detected in the monitored area. In the stable situation, the node with less residual energy will decrease its frame rate to its minimum, while the other overlapping sensor node will continue sensing with its initial frame rate. This approach [13] outperforms the algorithm in [14] where in every period, the video shots are compared using a similarity process, and if the two shots surpass a predefined threshold then one of the sensor nodes will send the frame. In [20] Priyadarshini et al. investigated the overlapping method, which reduces redundancies by turning off certain cameras and activating the appropriate number of cameras based on the overlapping FOVs (field of view) of various cameras.

In [17], the authors proposed a multi-cloud to multi-fog architecture and design two kinds of service models by employing containers to improve the resource utilization of fog nodes and reduce the service delay. The algorithm

is based on the transmission energy consumption of terminal devices and uses a dynamic threshold strategy to schedule requests in real time. This algorithm can reduce service latency, improve fog node efficiency, and prolong WSNs life cycle through energy balancing. Such approaches can be complementary to our approach which focuses more on the node side.

Based on the different approaches mentioned above and presented in the literature we introduced two approaches for data reduction at the sensor node level, where the FRABID algorithm a joint data reduction, and frame rate adaptation on sensing and transmission phases mechanism is introduced. This approach reduces the number of sensed frames based on a similarity method that outperforms other techniques since it is based on a real-time prediction in every period. Then a complementary technique for data reduction on sensor node level exploits the condition of overlapping nodes to reduce the redundant sensed data. This approach is based on the synchronization between overlapping nodes and achieved an important reduction at the level of sensed data outperforming other techniques presented in the literature.

### 3 Assumptions and System Model

In our scenario, the wireless video sensor network (WVSN) is composed of two different kinds of nodes: the video sensor nodes and the sink node as shown in Fig 1. In this system model, frames are captured periodically and sent directly to the sink. At the very beginning of the sensing, the initial frame rate is set to its maximum ( $FR_{max} = FR_{init}$ ), then after the activation of our data reduction algorithm, a new frame rate ( $NFR_i$ ) is dynamically computed at every period  $\Delta t_i$ .

We assume in our topology that a wireless video sensor network is homogeneous, where all sensors have the same storage, processing, battery power, sensing, and communication capabilities. In this system model and in a normal situation, all nodes will have the same battery power over time (since they are capturing with the same frame rate). The sensor nodes are deployed outdoors, and they are battery devices. Battery depletion has been identified as one of the primary causes of the lifetime limitation of these networks [21]. Replacing them regularly is impractical in large networks or may even be impossible in hostile environments [21].

The nodes are prone to failure for any internal or external reasons or die of the battery however stops functioning. By the time, after applying any algorithm that may put some nodes into sleep mode, or power-saving mode, we will get a group of sensor nodes with different battery power percentages. Suppose nodes x, y, and z are homogeneous, and node x is in power-saving mode, then basically its battery lifetime will last more than y and z. These variations and variables will help in the identification of our approaches in the upcoming sections.

## 4 Data Reduction on each Sensor

### 4.1 Sensing Phase: Frame Rate Adaptation

We address the energy and bandwidth reduction by reducing the number of frames first captured and then sent over the air. To achieve this aim, we introduce the Frame Rate Adaptation Based Image Difference (FRABID) algorithm that performs in two steps: 1) it adapts the rate at which the frames are captured, and 2) it selects among the captured frames the pertinent to send. Algorithm 1 details FRABID.

In this part, inspired by [2] and [14], we focus on the data reduction at the sensing phase by reducing the number of captured frames in each period on every video-sensor node. The FRABID algorithm adapts the frame rate of every sensor node dynamically and independently from the others for every period  $\Delta_i$ , the time needed to capture frames with a specific frame rate. Figure 3 illustrates the steps in each period for frame rate adaptation. It is based on image comparison. It adapts the frame acquisition rate by comparing consecutive images with the L1 norm. The L1 norm is the sum of the absolute values of the pixel-by-pixel difference between the two images. Figures 7 and 8 show what we can achieve from the succession of images of Figures 4, 5, and 6.

Our main contribution for frame rate adaptation is related to the generated image difference, since from this difference we can conclude two points: first, the detection of a motion, and second, the nature of the motion in a given scenario. To support these two points, Figures 7 and 8 show the image difference between Frames of Figure 4, 5 and 6, the difference generated in the frame is due to the motion of the man. If the difference generated in the frame is approximately negligible, then we can deduce the absence of motion.

Now, suppose that the Frames pictured in Figures 4, 5 and 6 are three consecutive frames. We generate the image difference between images 4, 5 to get the image of Figure 7, and the difference image between frames of Figures 5 and 6 to get the frame in Figure 8.

The difference shown in Figure 7 is small, so we can deduce from the image differences that the man has a slow movement, then he moves faster in the second two frames and so Figure 8 displays a much higher difference.

In each period  $\Delta_i$ , each sensor node captures the first two frames  $F_0$  and  $F_1$  at frame rate  $FR_{init}$ . The first frame  $F_0$  is sent to the sink, and  $F_1$  is compared to  $F_0$ . We thus generate the image difference ( $img_{diff}$ ) by calculating the absolute difference between image pixels of  $F_0$  and  $F_1$  as shown in the equation below:

$$img_{diff} = abs(F_0 - F_1) \quad (1)$$

According to the definition of the L1 norm, we still need to sum the difference presented in  $img_{diff}$ .  $img_{diff}$  is thus transformed into an array  $\Gamma_{diff}$  containing the value of each pixel of  $img_{diff}$ . Then, we sum the data presented

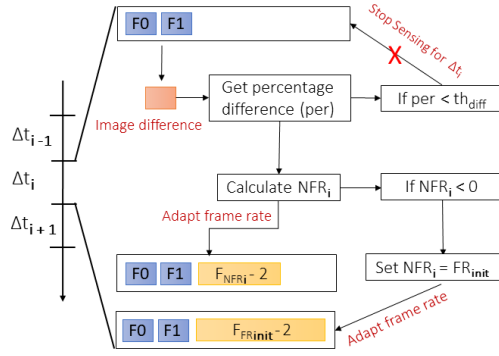
Fig. 3: Frame rate adaptation during period  $\Delta t_i$ 

Fig. 4: First Frame



Fig. 5: Second Frame



Fig. 6: Third Frame

in this array as follows:

$$sum_{diff} = \sum_{p=0}^{nbcomp} \Gamma_{diff_p} \quad (2)$$

where  $\Gamma_{diff_p}$  is the value of pixel  $p$  of  $img_{diff}$  and  $nbcomp$  is the size of the image in the number of pixels. As a result, we get  $sum_{diff}$  that represents the value of the difference between two frames, then the difference percentage  $per$  is computed from the  $sum_{diff}$  as follows:

$$per = \frac{sum_{diff} \times 100}{max_{val} \times nbcomp} \quad (3)$$

Where  $max_{val}$  is the maximum value that can be assigned to a pixel (In a gray-scale RGB image,  $max_{val} = 255$ ).

Now, the new frame rate  $NFR_i$  of Period  $\Delta_i$  is calculated as follows:

$$NFR_i = FR_{init} - \left( \frac{FR_{init}}{v} \times per \right) \quad (4)$$

where  $v$  represents the convergence speed of  $FR_i$ . The higher  $v$ , the quicker the frame rate is adapted but the more likely to miss important frames. Also,



note that we use  $FR_{init}$  and not  $FR_{i-1}$  since images could be very different from one period to another one and we could miss important events.

Yet, the frame rate is adapted based on the value of the percentage difference between the first two sensed frames in each period  $\Delta_i$ . If  $NFR_i < 0$ , the frame rate of the period  $\Delta_i$  is set to  $FR_{max}$ , since this means there is a high difference between two successive frames and so either a new event appears in the frame, or there is a fast motion as explained above. In both cases, we need a high frame rate to capture all frames in a critical scenario.

---

**Algorithm 1** FRABID run at every node
 

---

```

1: while True do
2:                                     # Sensor-node starts sensing
3:   for all Period  $\Delta_i$  do
4:     Capture two first frames  $F_0$  and  $F_1$ 
5:     Send  $F_0$  to the sink
6:     Generate  $img_{diff}$  between  $F_0$  and  $F_1$ 
7:     Compute per with equation 3
8:      $NFR_i = FR_{init} - (per \times FR_{init}/v)$  (Eq. 4)
9:     if ( $NFR_i < 0$ ) then
10:       Set  $NFR_i = FR_{init}$ 
11:     end if
12:     if  $per < th_{diff}$  then
13:       Stop capturing frames for period  $\Delta_i$ 
14:     else
15:       Send  $img_{diff}$  to the sink
16:     end if
17:   end for
18: end while

```

---

#### 4.2 Transmission Phase: Data Reduction

To reduce the energy consumption related to transmission level, we take the advantage of the similarity of consecutive frames as in [2] and [13] to reduce the number of sent frames to the sink. Every sensed frame is compared with the last sent frame using the method described in 4.1 and as shown in details on Figure 9. First, difference image will be generated (using Equation 1), then from Equation 3, we will get the percentage difference *per* between two frames. Figure 7 shows the output of the generated image difference. Based on the value of the percentage difference *per* the frame will be sent if  $per > th_{diff}$ , where  $th_{diff}$  is a predefined threshold set according to the criticality of the scenario monitored. If we are dealing with a delicate situation in which we need to catch even the tiniest movement, the threshold should be set to its minimum.

So, if the difference between two frames exceeds  $th_{diff}$ , the generated frame (like Figure 7 and 8) will be sent to the coordinator instead of sending the orig-

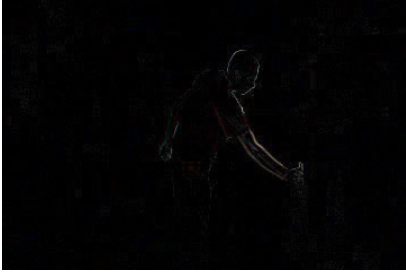


Fig. 7: Image difference between Figures 4 and 5

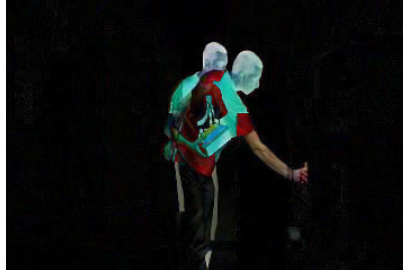


Fig. 8: Image difference between Figures 5 and 6

inal image. In our approach since we are using L1 norm to compare the frames, we are reducing energy consumption for processing time, since the difference image is already generated, while in other approaches like STAFRA algorithm in [2], frames are compared using the L2 norm, then another function is used to generate the difference image. The sent image is called a critical frame in the remainder of this paper because it means that an event is happening in the area of interest [2]. If the percentage  $per$  does not exceed  $th_{diff}$ , the frame will not be sent, and the sensor node will stop sensing for the current period (Figure 9). The frame will be called a similar frame.

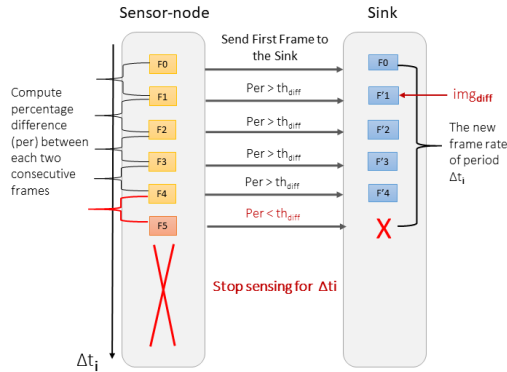


Fig. 9: Video sensor node behavior during period  $\Delta t_i$

## 5 Data Reduction in Overlapping Sensor-nodes

In section 4, we were interested in reducing the amount of sensed and sent frames from each node to the sink by applying the FRABID algorithm at

the sensor node itself. To furthermore reduce the data transmission, we now focus on the spatial correlation between neighboring sensor nodes to reduce the redundant sensed frames between overlapping sensor nodes by applying a new approach based on synchronization.

Before describing the Synchronization with Frame Rate Adaptation (SFRA) approach, we introduce the video sensing model and the characteristics of every video sensor node to proceed with the SFRA algorithm.

### 5.1 Video Sensing Model

We consider a 2-D model of a video sensor node where  $z = 0$  (XOY plane) and all the captured frames are compared as 2-D images not taking into account the third dimension. A video sensor node  $S$  is represented by the Field of View (FoV) of its camera. A FoV covers only a part of the surrounding area of a video sensor. A FoV is a vector of 4-tuple  $S(P; R_s; \vec{V}; \theta)$  where  $P$  is the position of  $S$ ,  $R_s$  is its sensing range,  $\vec{V}$  is the vector representing the line of sight of the camera's FoV e.g. its sensing direction, and  $\theta$  is the offset angle of the FoV as shown in Figure 10.

As mentioned before, we assume that all video sensor nodes are identical with fixed lenses providing a  $\theta$  angle FoV thus the same sensing range  $R_s$ , densely deployed randomly.

Each node  $S_i$  covers a sector area in its FoV. We define  $U_{S_i}$  as the set of sensors that have an intersecting coverage zone (FoV).

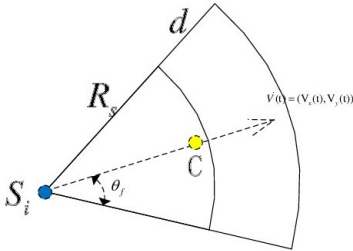


Fig. 10: Video sensor node behavior during period  $\Delta t_i$

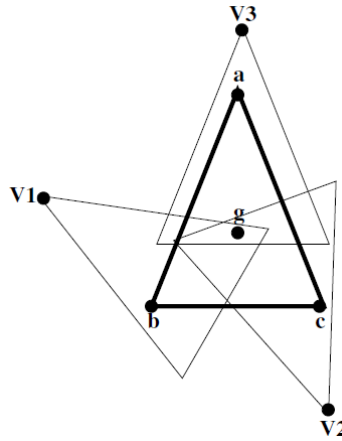


Fig. 11: FoV Coverage [5]

## 5.2 The Overlapping Method and Coverage

As mentioned above, in our topology sensor nodes are deployed randomly. This may increase the spatial correlation between neighboring sensor nodes, so the sensing range of two or more sensor-nodes may overlap (Figure 11). In such a scenario, the sensors capture typically redundant data of a given target since the same event may be captured by multiple sensors. Several approaches studied how to detect overlapping sensor-nodes [2,4,13,14]. In our work we will consider that the overlapping nodes are already detected using the geometric condition method presented in [13]. Suppose  $S_1$ ,  $S_2$  and  $S_3$  are selected as overlapping sensor nodes with a stable state to test the synchronization with frame rate adaptation algorithm (SFRA) as explained in the next section.

## 5.3 SFRA Algorithm

The idea behind synchronization is to reduce the number of sensed frames while preserving all needed information. As mentioned earlier, the synchronization phase starts when all overlapping sensor nodes ( $U_{s_i}$ ) are in a stable state.

Inspired by [13], a stable situation is a state where the sensor node is not capturing new information. Using the equations from 1 and 3 the sensor will compute the similarity between the captured frames. If two consecutive sensed frames are estimated as similar (no new information is represented in the second frame), the node does not send the second frame to the coordinator. The node counts the number of consecutive similar frames, if it surpasses  $nb$  (the required number of consecutive similar frames), the state of the area of interest monitored by the node is considered as stable (situation=1).

To activate the SFRA algorithm, each node should know:

- The set of overlapping nodes ( $U_{s_i}$ ) with its FoV
- The state of each node that belongs to  $U_{s_i}$

When all overlapping nodes that belong to  $U_{s_i}$  are in a stable state (situation = 1), they will set their frame rate to its minimum ( $FR_{min}$ ) and synchronize capturing frames between each other as illustrated in Figure 12.

One of the main challenges in the SFRA algorithm is to guarantee that the nodes are well synchronized. We assume that the system is well synchronized, where the clock drift which is the result of the clock skew (the difference between the two clocks frequency [22]) is approximately negligible.

The capturing phenomenon will be as follows:

In each period one sensor node that belongs to  $U_{s_i}$  will be capturing frames with ( $FR_{min}$ ). The other nodes will be in sleep mode. So, each node will sense at period  $\Delta_i$  then sleeps for  $N - 1$  periods, where  $N$  is the number of sensor nodes that belong to  $U_{s_i}$ .

Figure 12 represents the behavior of sensor-nodes when starting the synchronization.

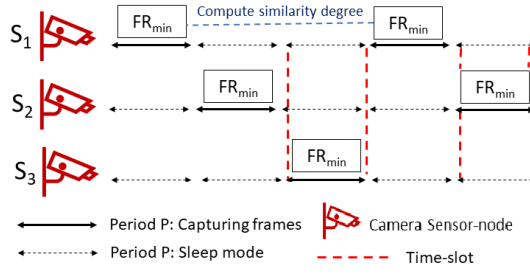


Fig. 12: Synchronization behaviour

Each captured frame will be compared with the last sent frame to the sink ( $F_{last}$ ) using the method explained in section 4.2, which is based on the L1 norm. If the difference surpasses a predefined threshold  $th_{diff}$ , the sensor-node will return to its normal state with frame rate  $FR_{max}$  and will notify the other nodes that belong to  $U_{s_i}$  to deactivate SFRA algorithm, since the newly captured frame represents a new event as explained in algorithm 2.

---

**Algorithm 2** SFRA run at node  $S_1$ 


---

```

1: Get  $situation_2$  and  $situation_3$ 
2: Get  $F_{last}$  the last sent frame to the sink
3: Set  $N$  the number of overlapping nodes
4: while  $situation_1 = 1$  and  $situation_2 = 1$  and  $situation_3 = 1$  do
5:    $FR_{S_1} = 1$  fps
6:   sleep ( $N - 1$ )
7:   Capture frame  $F_0$ 
8:   Generate  $img_{diff}$  between  $F_0$  and  $F_{last}$ 
9:   Compute  $per$  with equation 3
10:  if  $per > th_{diff}$  then
11:    Stop synchronization phase
12:    Set  $FR_{S_1} = FR_{max}$ 
13:    Notify all overlapping nodes
14:    Send  $img_{diff}$  to the sink
15:  end if
16: end while

```

---

## 6 Experimental Results

### 6.1 FRABID Algorithm

In this section, we present the results that validate our approach and compare them to the STAFRA algorithm in [2].

We implement the algorithms (FRABID and STAFRA) using Python Imaging Library (PIL), that has light image processing tools. First we get the difference image  $image_{diff}$  between frames  $F_0$  and  $F_1$ . Then we used Numpy

library to transform images into arrays and get the sum of the generated image difference. We made our simulations on a data-set [1] that provides a realistic, camera-captured, diverse set of videos that cover a wide range of detection challenges.

In our real scenario, we expect frequent motion, with different variations, since the video sensor-nodes are deployed to monitor wildlife. This scenario is exposed to continuous, periodic motion. For that, we have selected from the data-set [1] the videos that are captured outdoors, with different variations of motion, as detailed later.

The initial frame rate to capture the video is set to 30 frames per second ( $FR_{init} = 30$  fps) for the sensor-node, which is the maximum frame rate ( $FR_{max} = FR_{init} = 30$ ). Each period is  $\Delta_t = 1s$  and the threshold to detect critical frames is set to  $th_{diff} = 1\%$ , and speed convergence is set to  $v = 2$ .

## 6.2 Data Reduction: Sensing Phase

The main purpose in this work is to sense the frames that represent the critical situations and decrease the number of similar sensed frames.

For similar frames, they are only sensed and not sent to the sink, so we are capturing useless information. So, the FRABID algorithm decreased the number of similar sensed frames, thus reduced the power consumption on useless data. Note that a full compressed image size is equal to 30KB and the difference image  $img_{diff}$  size is 20KB. The thresholds and all the parameters can be adapted according to the application and the QoS required.

The simulation is done using two data-sets from [1]: The first selected video is called **Highway**<sup>1</sup>, where there is frequent motion with a slight dynamic background. The second selected video is called **Tramstop**, it presents more challenges for having different variations of motion and speed.

## 6.3 Highway Data-set

The video is captured for 1700 periods. Table 1 shows the initial recorded data in a normal state of the sensor-node when no reduction algorithm is activated.

Table 1: Initial Highway Data-set Records

	<b>Total</b>	<b>Critical</b>	<b>Similar</b>
Number of Frames	1700	600	400

For the sensing phase, the frame rate in each sensor node changes independently according to the technique explained in the above sections, where

<sup>1</sup> <http://changedetection.net/>

the difference between the first two sensed frames in each period will be the reference to select a new frame rate for the current period. From 1700 frames, all frames are sensed and sent to the coordinator.

Tables 1 and 2 show that, after activating the FRABID algorithm, the number of sensed frames decreased from 1700 to 971 so a reduction of 43% (Table 2). The number of similar frames sensed in a normal state without the activation of any data reduction algorithm is 400, while after activating FRABID algorithm, it decreased to 8, so a reduction of 98%.

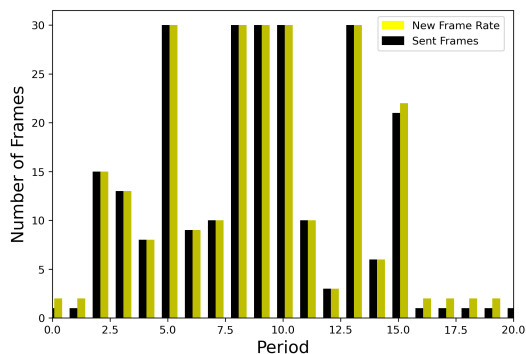


Fig. 13: Frame Rate and Sent Frames in FRABID Algorithm

Table 2: Records After Applying FRABID Algorithm

	Sensed	Critical	Similar
Number of Frames	971	963	8

Figure 13 shows the frame captured and sent in each period. The difference between both values never surpasses 5 frames and in the rest of the periods the number of sensed frames is equal to the frame rate. This affirms the validity of our approach in sensing only frames with critical events.

#### 6.4 Comparison

Our approach is compared to the STAFRA algorithm in [2], where the number of critical frames in each period affects the frame rate of the sensor-node for the next period.

Figure 14 shows the frame rate and the sent frames in each period after applying the STAFRA algorithm. The new frame rate is in most of the periods

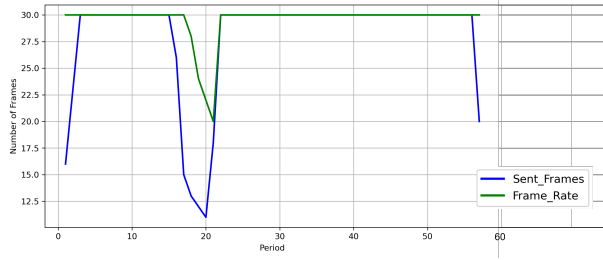


Fig. 14: Frame Rate and Sent Frames in STAFRA algorithm

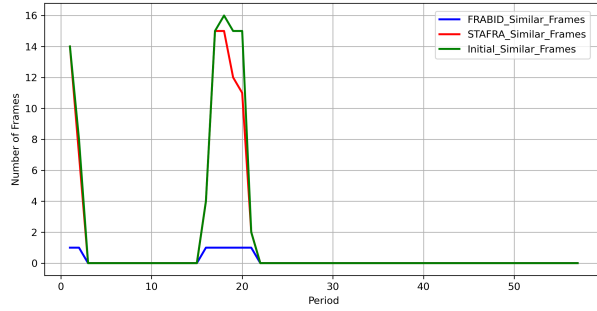


Fig. 15: Initial number of similar frames before and after STAFRA and FRABID activation

Table 3: Records Comparison Between FRABID and STAFRA on Tramstop Data-set

	<b>FRABID</b>	<b>STAFRA</b>
Total	3200	3200
Sensed	1254	3200
Sent	1232	2914
Similar	22	286

still set to its highest value (30 fps). This is due to the high rate of motion in the monitored scenario, which leads to keeping the frame rate high.

Figure 15 shows the initial number of similar frames (the frames that do not hold new information) and the number of similar frames sensed by the sensor-node in both algorithms FRABID and STAFRA. The FRABID algorithm senses approximately 2% of the similar frames, while with the STAFRA algorithm almost all similar frames are captured. From Table 3 we can see the number of similar frames sensed with FRABID algorithm activation is 22, while in STAFRA it is 286. The number of similar frames sensed in FRABID is reduced by 92% compared to STAFRA. This is related to the FRABID algorithm's idea to stop sensing if two frames are similar. So, data reduction



is achieved at the sensing level, and the power consumption at the processing phase is reduced.

Table 4: Records Comparison Between FRABID and STAFRA on Highway Data-set

	<b>FRABID</b>	<b>STAFRA</b>
Total	1700	1700
Sensed	971	1684
Sent	963	1594
Similar	8	90

### 6.5 SFRA Algorithm

In this section, we present the results that validate our approach and compare them to the INS algorithm in [13]. We implement the algorithms (SFRA and INS) using Python Imaging Library (PIL) that has light image processing tools. First we used the function from PIL imaging library in Python for image comparison to generate  $image_{diff}$ , the difference image between frames  $F_0$  and  $F_1$ .

We made our simulations on a data-set named ToCaDa [23], containing two sets of 25 temporally synchronized videos corresponding to two scripted scenarios. 25 cameras were scattered around the university campus of Paul Sabatier University in Toulouse. 8 among the cameras were located in front of the building and filmed it with large overlapping fields of view as it can be seen in the figure 16. Cameras C4, C5 and C7 are selected for our simulation to test the SFRA algorithm. Due to the lack of datasets from the smart agriculture domain, this dataset is considered as a convenient substitute for our simulation since the selected scenario includes fast motion of cars and slow motion of people with variation of sizes of the moving object, plus the presence of sensitive cases like trees that moves with the wind.

The initial frame rate to capture the video is set to 30 frames per second ( $FR_{init} = 30$  fps) which is the maximum frame rate ( $FR_{max} = FR_{init} = 30$ ), and the minimum frame rate is set to 1 frame per second ( $FR_{min} = 1$  fps) for each camera. Each period is  $\Delta_t = 1s$ , and initial frame rate is equal to  $FR_{init} = 30$  frames per second. The threshold to detect critical frames is set to  $th_{diff} = 1.3\%$ . The aim is to reduce the number of captured frames in a stable state by reducing the frame rate and applying synchronization between the overlapping nodes.



Fig. 16: Scenario

### 6.6 Data Reduction: Sensing Phase

The selected videos are captured for 10204 periods. Table 5 presents the initial recorded data in a normal state of the three cameras C4, C5, and C7 when no reduction algorithm is activated, where Initial Event Detected represents the number of the frame where the intrusion is detected.

A stable state is detected at the following time intervals:

C4: 20s from 900 to 1512

C5: 20s from 900 to 1500

C7: 21s from 900 to 1540

C5 first detected the event at frame number 1500, then C4 at frame number 1512, and for C7 the event appears in period 5 at frame number 1540.

A stable state is defined after having 60 consecutive similar frames ( $nb = 60$ ) which means 2 seconds of stability. The thresholds and all the parameters can be adapted according to the application and the QoS required.

The SFRA algorithm is activated after satisfying the conditions explained in 5.3. Synchronization starts at period 33 (frame number 990), since the first 3sec are dedicated to detecting the stable state and exchanging information needed to start synchronization. So, the SFRA algorithm is tested on the three overlapping cameras (C4, C5, and C7) in the stable state from periods 30 to 50. There are different ways to start the synchronization depending on the selected order for the nodes to capture frames. For that we made different simulations taking into consideration the different order possibilities of camera sensor nodes to achieve the synchronization phase and observe the results under different conditions.

After activating the SFRA algorithm, the results presented in tables 6, show a reduction of 98% of the captured frames in a stable state. The simulations show the different scenarios of capturing frames which are based on the order of cameras selected to start synchronization.

Scenario 2 in Table 6 shows the results of the number of captured frames by the overlapping nodes, where C5 was first selected to start synchronization then C4 and C7 respectively. C5 first detected the new event in period 51

Table 5: Data-set Records for overlapping cameras C4,C5 and C7 before activating any algorithm

	Total Captured Frames	Initial Event Detected
Camera 4	612	1512
Camera 5	600	1500
Camera 7	640	1540

Table 6: Number of sensed frames after activating SFRA algorithm in different synchronization scenarios

Scenario	Order	Sensors	Sensed	First Frame Detected
1	C4,C5,C7	C4	8	1531
		C5	8	1561
		C7	8	1591
2	C5,C4,C7	C4	8	1561
		C5	8	1531
		C7	8	1591
3	C7,C5,C4	C4	12	1861
		C5	8	1561
		C7	9	1621
4	C4,C7, C5	C4	8	1531
		C5	7	1500
		C7	8	1591

(frame number 1531). The impact of synchronization delayed the detection of the intrusion by one period at maximum for all overlapping nodes (at period 52, all the overlapping nodes will set their frame rate to  $FR_{max}$  for each period returning to their initial state). While based on the records of scenario 3, where the order of camera selection is C7, C5, and C4 respectively show that C5 first detected the event at period 52, which is a delay of at most 2 periods from detecting the event on time.

Scenario 4 in Table 6 presents the results of the best case which is detecting the new event on time. As we can see C5 first detected the event at period 50 (frame number 1500), which is the first event captured by all the overlapping sensor nodes as shown in table 5. For that, the best case will be when the synchronization technique selects C5 to be the last node to capture.

## 6.7 Comparison

The SFRA algorithm is compared to the INS algorithm in [13]. In the INS algorithm one out of the selected overlapping nodes that has more residual energy is selected to sense frames with  $FR_{max}$ , and the other nodes will go to transmission idle mode where the frame rate is set to its minimum ( $FR_{min} = 1$  frame per second).

The simulation is done in the stable state from period 33 to period 50 on the three selected overlapping cameras C4, C5, and C7. We assume that C4 has more residual energy, so for the INS algorithm, C4 will capture frames with  $FR_{max} = 30$  frame per second while C5 and C7 with  $FR_{min} = 1$  frame per second.

In INS we can guarantee the capture of the new event in time, while in SFRA we may miss a few frames due to the impact of synchronization. The difference between those approaches in terms of data reduction is shown in Table 7. The results show that the SFRA algorithm outperforms the INS algorithm [13] in terms of sensing data reduction. INS achieved 65% data reduction while SFRA 98%. So, the SFRA algorithm added 33% more data reduction than the INS algorithm.

Table 7: SFRA and INS comparison

	Initial Captured Frames	INS	SFRA
C4	510	510	8
C5	510	17	8
C7	510	17	8
Total	1530	544	24

The overall power consumption reduction cannot be specified since there are different levels to be studied like the cost of synchronization which is based on the mechanism used to achieve this aim. The cost of using GPS, type of the used hardware. Our main focus is on data reduction, which can be achieved regardless of other constraints.

## 7 Combining FRABID and SFRA algorithms

In this section, we have tested the impact of activating both algorithms FRABID and SFRA on the sensor node.

Simulation is done on the same data set in 6.6. The algorithms are tested on the three Cameras C4, C5, and C7 from period 29 to period 70 (41 sec). Table 8 shows the results after running the FRABID algorithm and then activating the SFRA algorithm when a stable state is detected with the overlapping sensor nodes. In the initial state, before activating any algorithm the total number of frames captured by the three sensors is 3690 frames. When running the FRABID algorithm the total captured frames decreased to 209. Then when activating both algorithms (FRABIB + SFRA) on the three sensor nodes the number of sensed frames decreased to 153, so we added 27% more data reduction in all three overlapping nodes while guaranteeing the detection of critical events. This is due to the synchronization fashion that occurred in the stable state from period 33 to period 50.

Table 8: FRABID and SFRA data reduction

	Initial	FRABID	FRABID + SFRA
C4	1230	83	53
C5	1230	68	54
C7	1230	58	46
Total	3690	209	153

## 8 Conclusion and Future Work

In this work we have tackled the problem of energy consumption and bandwidth limitation in WWSNs for smart agriculture especially for livestock farming applications. Two complementary algorithms are proposed dedicated for data reduction at the level of each sensor node and between overlapping wireless video sensor nodes to monitor livestock in areas where man has no control.

The first approach is FRABID algorithm. Simulations based on real data sets show an important reduction in the scenarios with periodic, continuous motion when activating the FRABID algorithm since the frame rate is adapted according to the variation of speed in the monitored zone, plus a reduction in the number of sensed similar frames. Thus, it reduces the energy consumption needed for the sensing process. The algorithm creates a difference image between two frames, in case the frames are similar, the second frame will not be sent to the sink, otherwise, the difference image will be sent which is 30% smaller than the original image. This approach reduced energy consumption at the processing level, and for the transmission process on the sensor-node level by reducing the number and the size of sent frames to the sink.

To further reduce the redundant sensed data, we exploit the condition of overlapping sensor nodes by proposing the SFRA algorithm. When a stable state is detected in the monitored area, the overlapping nodes will decrease their frame rate to the minimum and capture frames in a synchronized fashion. When one of the overlapping nodes captures an event, all nodes must return to their initial state with maximum frame rate. Simulations were conducted on a real data set. The results showed a reduction up to 98% in stable states. The selected order of nodes to achieve synchronization will even lead to a best case, where the event will be detected in time, or intrusion detection will be delayed at maximum  $N$  periods which is considered the worst case. The evaluation of the results depends on the criticality of the use case. In our scenario, missing a few critical frames is not considered a big loss, since the case is not delicate and the nature of motion is not of high speed.

In this work we have selected datasets that resembles the real-life situation of livestock monitoring due to the unavailability for the moment of datasets collected in livestock monitoring use case, thus as a future work we want to adjust our algorithms on a real-life situation. Then we will investigate in the performance of our algorithms on different use cases such as remote surveillance (property intruders, forest fire), smart cities, etc. to expand the usage

of the algorithms. Then we will test the impact of the different parameters such as the convergence speed  $v$  of the new frame rate setting, the threshold value  $th_{diff}$  above which an action is taken, and the value which detects the stable state. Then, a general study of the energy consumption taking into consideration the real cost of processing at the sensor node level and the cost of synchronization. Finally our approach will be enhanced by getting an inclusion of a mobility prediction scheme to follow the motion and "wake up" only concerned sensors and real experimentation will be conducted on real sensor nodes.

## Acknowledgements

This work was partially supported by a grant from CPER DATA and by LIRIMA Agrinet project.

## Conflict of Interest Statement

We have no conflicts of interest to disclose. All authors declare that they have no conflicts of interest.

## References

1. Yi Wang, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar. Cdnet 2014: An expanded change detection benchmark dataset. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 393–400, 2014.
2. Christian Salim, Abdallah Makhoul, and Raphaël Couturier. Energy-efficient secured data reduction technique using image difference function in wireless video sensor networks. *Multimedia Tools and Applications*, 79(3):1801–1819, 2020.
3. Waleed M. Ismael, Mingsheng Gao, Asma A. Al-Shargabi, and Ammar Zahary. An In-Networking Double-Layered Data Reduction for Internet of Things (IoT). *Sensors (Basel, Switzerland)*, 19(4):795, February 2019.
4. Amrita Ghosal, Subir Halder, and Mauro Conti. Disc: A novel distributed on-demand clustering protocol for internet of multimedia things. In *2019 28th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9. IEEE, 2019.
5. Junaid Akram, Hafiz Suliman Munawar, Abbas Z. Kouzani, and M A Parvez Mahmud. Using adaptive sensors for optimised target coverage in wireless sensor networks. *Sensors*, 22(3), 2022.
6. Walid Osamy, Ahmed M Khedr, Ahmed Salim, Amal Ibrahim AlAli, and Ahmed A El-Sawy. Recent studies utilizing artificial intelligence techniques for solving data collection, aggregation and dissemination challenges in wireless sensor networks: a review. *Electronics*, 11(3):313, 2022.
7. Zhenquan Qin, Can Ma, Lei Wang, Jiaqi Xu, and Bingxian Lu. An overlapping clustering approach for routing in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 9(3):867385, 2013.
8. Saurabh Kumar and Hyungwon Kim. Energy efficient scheduling in wireless sensor networks for periodic data gathering. *IEEE access*, 7:11410–11426, 2019.

9. Radhakrishnan Maivizhi and Palanichamy Yogesh. Spatial correlation based data redundancy elimination for data aggregation in wireless sensor networks. *2020 International Conference on Innovative Trends in Information Technology (ICITIIT)*, pages 1–5, 2020.
10. Ali Benzerbadj and Kechar Bouabdellah. Redundancy and criticality based scheduling in wireless video sensor networks for monitoring critical areas. *Procedia Computer Science*, 21:234–241, 12 2013.
11. James T Thorson, Mayumi L Arimitsu, Lewis AK Barnett, Wei Cheng, Lisa B Eisner, Alan C Haynie, Albert J Hermann, Kirstin Holsman, David G Kimmel, Michael W Lomas, et al. Forecasting community reassembly using climate-linked spatio-temporal ecosystem models. *Ecography*, 44(4):612–625, 2021.
12. Christian Salim, Abdallah Makhoul, Rony Darazi, and Raphaël Couturier. Kinematics based approach for data reduction in wireless video sensor networks. In *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–8, 2018.
13. Christian Salim and Nathalie Mitton. Image Similarity based Data Reduction Technique in Wireless Video Sensor Networks for Smart Agriculture. In *AINA 2021 - 35th International Conference on Advanced Information Networking and Applications*, Toronto, Canada, May 2021.
14. Christian Salim, Abdallah Makhoul, Rony Darazi, and Raphael Couturier. Similarity based image selection with frame rate adaptation and local event detection in wireless video sensor networks. *Multimedia Tools and Applications*, 78(5):5941 – 5967, 2019.
15. Gaby Bou Tayeh, Abdallah Makhoul, JACQUES DEMERJIAN, and David Laiymani. A new autonomous data transmission reduction method for wireless sensors networks. In *Middle East and North Africa COMMUNICATIONS Conference*, Jounieh, Lebanon, April 2018.
16. R. Monika, R. Hemalatha, and S. Radha. Energy efficient surveillance system using wwsn with reweighted sampling in modified fast haar wavelet transform domain. *Multimedia Tools Appl.*, 77(23):30187–30203, December 2018.
17. Juan Luo, Luxiu Yin, Jinyu Hu, Chun Wang, Xuan Liu, Xin Fan, and Haibo Luo. Container-based fog computing architecture and energy-balancing scheduling algorithm for energy iot. *Future Generation Computer Systems*, 97:50–60, 2019.
18. Arslan Rafi, Gulsayyar Ali, Junaid Akram, et al. Efficient energy utilization in fog computing based wireless sensor networks. In *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pages 1–5. IEEE, 2019.
19. Ding Xu, Qun Li, and Hongbo Zhu. Energy-saving computation offloading by joint data compression and resource allocation for mobile-edge computing. *IEEE Communications Letters*, 23(4):704–707, 2019.
20. Sushree Bibhuprada B. Priyadarshini, Biswa Mohan Acharya, and Debapriya Soumyesh Das. Redundant Data Elimination & Optimum Camera Actuation in Wireless Multimedia Sensor Network (WMSN). *International Journal of Engineering Research & Technology*, 2(6), June 2013.
21. Mohammed Abo-Zahhad, Mohammed Farrag, Abdelhay Ali, and Osama Amin. An energy consumption model for wireless sensor networks. In *5th International Conference on Energy Aware Computing Systems Applications*, pages 1–4, 2015.
22. Dennis Krummacker, Christoph Fischer, Khurshid Alam, Michael Karrenbauer, Sergiy Melnyk, Hans Dieter Schotten, Peng Chen, and Siyu Tang. Intra-network clock synchronization for wireless networks: From state of the art systems to an improved solution. *2020 IEEE 2nd International Conference on Computer Communication and the Internet (ICCCI)*, pages 36–44, 2020.
23. Thierry Malon, Geoffrey Roman-Jimenez, Patrice Guyot, Sylvie Chambon, Vincent Charvillat, Alain Crouzil, André Péninou, Julien Pinquier, Florence Sèdes, and Christine Sénac. Toulouse campus surveillance dataset: scenarios, soundtracks, synchronized videos with overlapping and disjoint views. *Proceedings of the 9th ACM multimedia systems conference*, pages 393–398, 2018.