Data Reduction and Frame Rate Adaptation in WVSN

Jana Koteich
Inria
France
jana.koteich@inria.fr

Christian Salim
Inria
France
christian.salim@inria.fr

Nathalie Mitton
Inria
France
nathalie.mitton@inria.fr

***Abstract*—Wireless Video Sensor Networks (WVSNs) are becoming one of the most used technologies for surveillance and livestock monitoring. They are composed of small embedded video and camera motes 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 WVSNs because of the limited energy of nodes and the medium scarcity. 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, the greater the number of frames sent on the network the more bandwidth is used. In this paper, FRABID, 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 [3]. 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.**

***Index Terms*—data reduction, frame rate adaptation, wireless video sensor networks, shot similarity, image difference, event detection.**

I. Introduction

Nowadays, wireless video sensor networks are considered an important part of the surveillance field systems, where they are taking great attention to livestock monitoring [8]. Wireless video sensor networks (WVSNs) serve as low-cost monitoring systems. They are deployed in a remote site to monitor livestock that is exposed to threats from wild animals like jackals in South Africa. Understanding the wild animals' behavior would facilitate the means of protection of cattle in places beyond man's control. For that, WVSNs are deployed for livestock monitoring, where they process in real-time and retrieve multimedia data periodically to be sent to a sink.

In a WVSN system, limited energy resource nodes capture frames periodically. This scenario consumes a lot of energy. Maximization of system lifetime and energy conservation is commonly recognized as a key challenge in the design and implementation of WVSNs. So, the main target is to reduce the energy consumption related to the sensing and transmission phases at the sensor node level.

The limited lifetime of the sensor nodes must be taken into account by WVSNs. This periodic cycle leads to a lot of redundant data sent to the sink if no changes occur in the monitored zone of interest, especially when dealing with multimedia. To achieve data reduction at the sensor-node level, we introduce a Frame Rate Adaptation Based on Image Difference algorithm (FRABID), which reduces the images sent in two steps. First at the sensing level by reducing the number of frame captures, 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 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 sensing data by more than 90%.

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 STAFRA algorithm in [2], and decreased the redundancy by reducing the number of similar sensed frames to less than 10%, outperforming other approaches, while guaranteeing the capture of important events.

The remainder of this paper is organized as follows: Section II introduces the state of the art. Section III presents the model of our network. Section IV presents the data reduction technique for sensing and transmission phases. In Section V simulation results are discussed. Finally, conclusions are drawn in Section VI.

II. Related Work

Several research work dealing with data redundancy and energy reduction in WVSNs have been proposed in the literature. In this section, we will browse some of these approaches while focusing on the data reduction at the sensing phase.

Several research work for energy reduction has been proposed to decrease data redundancy: Scheduling methods [4]–[6], [9], [13], Data aggregation [18], Geometrical criteria [14]–[16], prediction techniques [8], frame rate adaptation [1], [2], [7], [10], [11]. In Akkaya et al [17], a GPS module is used to control the cameras and to determine which camera should be activated based on the sensor's location. In [4] the authors divided the region into several clusters using a clustering methodology. In each cluster, to avoid data redundancy for all overlapping cameras, a scheduling approach has been adopted in their method. Authors in [5] divided the region according to the different risk levels of the sensor nodes to form several areas of interest. Each area has its own adaptive scheduling model. This model changes the capture speed of the node based on its risk level and environment. Clustering methods can be a helpful approach to be studied later as a complementary of the work presented in this paper in the case of large infrastructure.

In [8], 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.

Several studies tried to solve the issue of data redundancy by taking into consideration overlapping sensor-nodes. The authors in [1], [2], [7] used geometrical conditions to detect overlapping sensors. After detecting overlapped sensor-nodes, the authors in [1] defined a stable situation, which is the case of the absence of new information in the monitored zone. 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 [1] outperforms the algorithm in [7] where in every period, the video shots are compared using a similarity process, and if the two shots surpasses a predefined threshold then one of the sensor-nodes will send the frame. Both algorithms [1], [7] used to apply the method of overlapping sensor-nodes between two nodes only. Priyadarshini et al. [12] 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. The approach of overlapping sensor-nodes is helpful and could be studied later on as a complementary work of the approach presented in this paper to decrease redundancy between overlapping sensor-nodes.

Different approaches used image comparison to reduce the energy consumption at the transmitting level. In [7] the MASRA algorithm used color-edge similarity method to compare two consecutive frames, and in [2] the STAFRA algorithm used norm L2 simple euclidean distance for similarity method. 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.

In [2] and [7], 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 [7], 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 lot of motion, the frame rate will always be high, since there is 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.

III. 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 Δt_i.

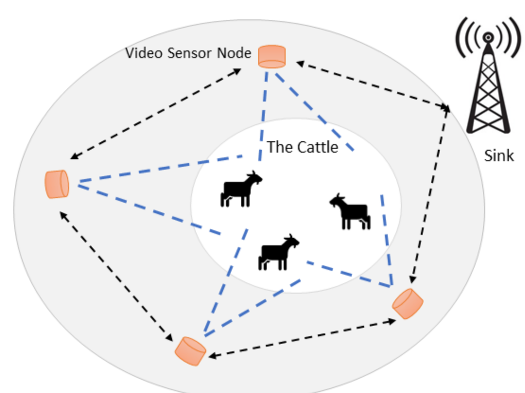


Fig. 1: System Model Architecture

IV. Data Reduction

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 on Image Difference (FRABID) algorithm which 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.

A. Sensing Phase: Frame Rate Adaptation

In this part, inspired from [2] and [7], 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. FRABID algorithm adapts the frame rate of every sensor-node dynamically and independently from the others for every period Δ_i, the time needed to capture frames with a specific frame rate. Figure 2 illustrates the steps in each period for frame rate adaptation. It is based on images comparison. It adapts the frame acquisition rate by comparing consecutive images with the L1 norm. L1 norm is the sum of the absolute values of the pixel-by-pixel difference between the two images. Figure 4 shows what we can achieve from the succession of images of Figure 3.

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, Figure 4a shows the image difference between Frames of Figure 3a and 3b, 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 Frames pictured in Figure 3 are three consecutive frames. We generate the image difference between images 3a, 3b to get image of Figure 4a, and the difference image between frames of Figures 3b and 3c to get the frame in Figure 4b. The difference shown in Figure 4a is small, so we can deduce from the image differences that the man has a slow movement, then he moved faster in the second two frames and so Figure 4b displays a much higher difference.

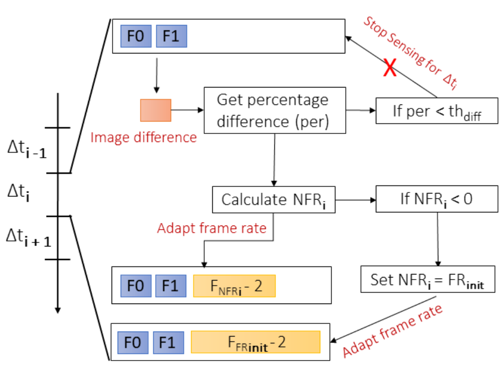


Fig. 2: Frame rate adaptation during period Δt_i

In each period Δ_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:

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

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

$$sum_{diff} = \sum_{p=0}^{nbcomp} \Gamma_{diff_p} \tag{2}$$

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

$$per = \frac{sum_{diff} \times 100}{max_{val} \times nbcomp} \tag{3}$$

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 Δ_i is calculated as follows:

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

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 frame. 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 Δ_i. If $NFR_i < 0$, the frame rate of the period Δ_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 in Section IV-A. In both cases, we need a high frame rate to capture all frames in a critical scenario.

B. 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] 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 IV-A as shown in details on Figure 5. First, difference image will be generated (using Equation 1), then from Equation 3, we will get the percentage difference per between two frames. Figure 4a shows the output of the generated image difference. Based on the value of the percentage difference per the frame will be sent if Equation 5 below holds:

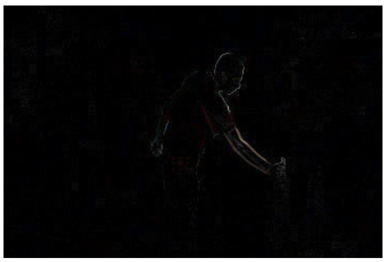
$$per > th_{diff} \tag{5}$$


(a) First frame


(b) Second frame


(c) Third frame

Fig. 3: Frame succession example


(a) Between Figures 3a and 3b


(b) Between Figures 3b and 3c

Fig. 4: Difference between successive frames by using L1 norm

Algorithm 1 FRABID run at every node

```
while True do
  {Sensor-node starts sensing}
  for all Period Δ_i do
    Capture two first frames F_0 and F_1
    Send F_0 to the sink
    Generate img_diff between F_0 and F_1
    Compute per with equation 3
    NFR_i = FR_init − (per × FR_init/v) (Eq. 4)
    if (NFR_i < 0) then
      Set NFR_i = FR_init
    end if
    if per < th_diff then
      Stop capturing frames for period Δ_i
    else
      Send img_diff to the sink
    end if
  end for
end while
```

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 4a and 4b) will be sent to the coordinator instead of sending the original 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 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 5). The frame will be called a similar frame.

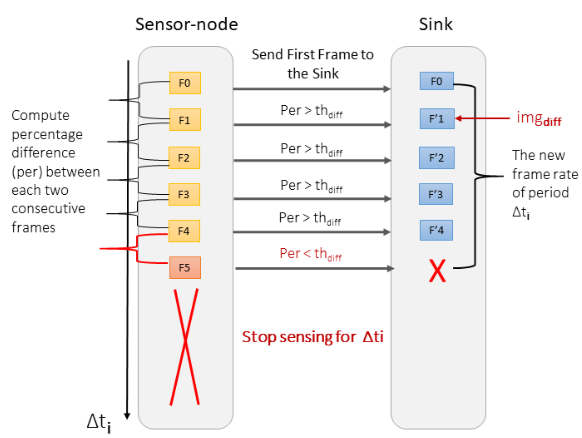


Fig. 5: Video sensor node behaviour during period Δt_i

V. Experimental Results

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. For image comparison, first we used the function from PIL imaging library in Python:

$$img_{diff} = ImageChops.difference(F_0, F_1) \quad (6)$$

The above function (6) will return $image_{diff}$, the difference image 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. Then, we made our simulations on a data-set [3] that provides a realistic, camera-captured, diverse set of videos that cover a wide range of detection challenges.

In our real scenario, we expect continuous motion, with different variations, since the video sensor-nodes are deployed to monitor wildlife and such scenario is exposed to continuous, periodic motion. For that, we have selected from the data-set [3] 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 initial frame rate is equal to $FR_{init} = 30$ frames per second. The threshold to detect critical frames is set to $th_{diff} = 1\%$ and speed convergence is set to $v = 2$.

A. 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. For that, FRABID algorithm showed an important reduction in the number of similar sensed frames, thus reducing power consumption on useless data.

The simulation is done using two data-sets from [3]: The first selected video is called Highway[1], where there is continuous 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.

1) Highway Data-set: The video is captured for 57 periods. Table I shows the initial recorded data in a normal state of the sensor-node when no reduction algorithm is activated.

TABLE I: Initial Highway Data-set Records

	Sensed	***Critical***	***Similar***
Number of Frames	1700	1300	400

For the sensing phase, the frame rate in each sensor node changes independently according to the technique explained in the above sections, where 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 I and II show that, after activating the FRABID algorithm, the number of sensed frames is decreased from 1700 to 971 so a reduction of 43% (Table II). 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 reduction of 98%.

[1]http://changedetection.net/

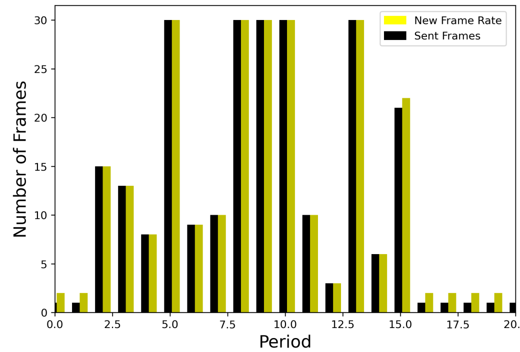


Fig. 6: Frame Rate and Sent Frames in FRABID Algorithm

TABLE II: Records After Applying FRABID Algorithm

	Sensed	***Critical***	***Similar***
Number of Frames	971	963	8

Figure 6 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.

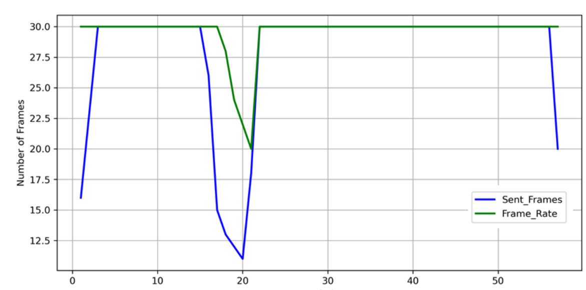


Fig. 7: Frame Rate and Sent Frames in STAFRA algorithm

TABLE III: Records Comparison Between FRABID and STAFRA on Tramstop Data-set

	FRABID	***STAFRA***
Total	3200	3200
Sensed	1254	3200
Sent	1232	2914
Similar	22	286

B. 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.

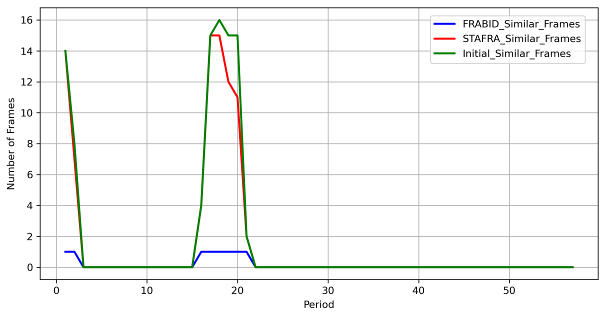


Fig. 8: Initial number of similar frames before and after STAFRA and FRABID activation

Figure 7 shows the frame rate and the sent frames in each period after applying STAFRA algorithm. The new frame rate is in most of the periods still set to its highest value (30 fps). This is due to the high rate of motion in the monitored scenario, which leads to keep the frame rate high.

Figure 8 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. FRABID algorithm sense approximately 2% of the similar frames, while with STAFRA algorithm almost all similar frames are captured. From Table III we can see the number of similar frames sensed with FRABID algorithm activation is 22, while in STAFRA 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 IV: Records Comparison Between FRABID and STAFRA on Highway Data-set

	FRABID	***STAFRA***
Total	1700	1700
Sensed	971	1684
Sent	963	1594
Similar	8	90

VI. Conclusion

In this paper, FRABID, a new data reduction adaptive frame rate algorithm is presented to adapt the frame rate at the sensing phase and transmission phase. This adaptation is based on the difference between frames. Our simulations based on real data-sets show an important reduction in the scenarios with periodic, continuous motion, 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. For future work, we will first investigate the impact of the different parameters such as the convergence speed v of the new frame rate setting and the threshold value th_{diff} above which an action is taken. Then, our approach will be enhanced by getting an inclusion of mobility prediction scheme to follow the motion and "wake up" only concerned sensors and real experimentation will be conducted on real sensor-nodes.

Acknowledgment

This work was partially supported by CPER DATA and LIRIMA AgriNet project.

References

[1] C. Salim, N. Mitton. Image Similarity based Data Reduction Technique in Wireless Video Sensor Networks for Smart Agriculture. AINA 2021
[2] C. Salim, A. Makhoul, R. Couturier. Energy-efficient secured data reduction technique using image difference function in wireless video sensor networks. Multimedia Tools and Applications, 2020
[3] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, CDnet 2014: An Expanded Change Detection Benchmark Dataset, in Proc. IEEE Workshop on Change Detection (CDW-2014), 2014.
[4] M. Alaei and Jose M. Barcelo-Ordinas. A method for clustering and cooperation in wireless multimedia sensor networks. Sensors, 10(1):3145–3169, 2010.
[5] CongDuc Pham, A. Makhoul, and R. Saadi. Risk-based adaptive scheduling in randomly deployed video sensor networks for critical surveillance applications. J. Network and Computer Applications, 34(2):783–795, 2011.
[6] Y. Yao and Georgios B. Giannakis. Energy-efficient scheduling for wireless sensor networks. IEEE TRANSACTIONS ON COMMUNICATIONS, 2005.
[7] C. Salim, A. Makhoul, R. Darazi, R. Couturier. Similarity based image selection with frame rate adaptation and local event detection in wireless video sensor networks. Multimedia Tools and Applications, 2019
[8] C. Salim, A. Makhoul, R. Darazi and R. Couturier, "Kinematics Based Approach for Data Reduction in Wireless Video Sensor Networks," WiMob, 2018
[9] M. Alaei and Jose M. Barcelo-Ordinas. A method for clustering and cooperation in wireless multimedia sensor networks. Patent Application Publication, 2010.
[10] G.B. Tayeh, A. Makhoul,J. Demerjian,D. Laiymani. A new autonomous data transmission reduction method for wireless sensors networks. MENACOMM, 2018
[11] R. Monika, R. Hemalatha, S. Radha. Energy efficient surveillance system using wvsn with reweighted sampling in modified fast haar wavelet transform domain. Multimed Tools Applications, 2018
[12] D. S. Das, S. B. B. Priyadarshini, B. M. Acharya. Redundant data elimination and optimum camera actuation in wireless multimedia sensor network (wmsn), 2013.
[13] Z. Qin, L. Wang, C. Ma, J. Xu, and B. Lu. An overlapping clustering approach for routing in wireless sensor networks. International Journal of Distributed Sensor Networks 2013.
[14] J.M. Bahi, A. Makhoul, and M. Medlej. A two tiers data aggregation scheme for periodic sensor networks. Ad Hoc and Sensor Wireless Networks 2014.
[15] A. Benzerbadj and B. Kechar. Redundancy and criticality based scheduling in wireless video sensor networks for monitoring critical areas. Procedia Computer Science 2013.
[16] S. Benbernou, A. Makhoul, M.-S. Hacid, and A. Mostefaoui. A spatio-temporal adaptation model for multimedia presentations. ISM 2005.
[17] A. Newell and K. Akkaya. Distributed collaborative camera actuation for redundant data elimination in wireless multimedia sensor networks. Ad Hoc Networks 2011.
[18] J.M. Bahi, A. Makhoul, and M. Medlej. An optimized in-network aggregation scheme for data collection in periodic sensor networks. ADHOC-NOW 2014.