An Efficient Distributed Collaborative Camera Actuation Algorithm for Redundant Data Elimination for Event Detection and Monitoring in Wireless Multimedia Sensor Networks

Ahmed Salim
Department of Mathematics,
Faculty of Science,Zagazig University,
Zagazig, P. O. Box 44519, Egypt
Department of Mathematics,
Faculty of Science and Arts, Al-mithnab,
Qassim University, P. O. Box 931,
Buridah 51931, Al-mithnab, Kingdom of Saudi Arabia

Hagar Ramdan
Mathematics Department,
Faculty of Science, Zagazig University,
Zagazig, P. O. Box 44519, Egypt

ABSTRACT

Wireless Multimedia Sensor Network(WMSN) is an extension of Wireless Sensor Network(WSN), where in addition to scalar sensors camera sensors are present. In wireless multimedia sensor networks (WMSNs), a high cost of processing and communicating the multimedia data is required. So it is critical to reduce possible data redundancy. Data redundancy occurs due to overlapping of Field of view (FOV) of camera sensors. Data redundancy affects on the communication cost in terms of bandwidth used, CPU processing etc. increases. Therefore, camera sensors should only be actuated when an event is detected within their vicinity. The scalar sensors first detect the occurrence of an event in the region of interest. Then the scalar sensors reports their corresponding camera sensors regarding the occurrence of event. In this paper, a distributed actuation scheme is proposed which depends to activate the least number of cameras while still preserving the necessary event coverage to avoid possible redundancy in the multimedia data. Even though the camera sensors have heard from scalar sensors about an occurring event, they may not cover the event. These nodes unnecessarily undergo distributed camera actuation scheme and some or all of them are activated. So our objective is to keep such cameras in turned off condition and to activate optimum number of camera sensors while preserving the necessary event coverage. The basic idea of this scheme is the collaboration of camera sensors that have heard from scalar sensors about an occurring event to minimize the possible coverage overlaps and also their FoVs intersect with the event region. This paper also proposes distributed actuation schemes for monitoring the event boundary. Simulation are presented to show the performance of our and other work in terms of coverage ratio, and the number of activated camera sensors under several random deployment schemes.

Keywords

Wireless Multimedia Sensor Network(WMSN), boundary node, event detection, camera actuation, Field of View(FoV), redundancy, scalar sensors.

1. INTRODUCTION

Wireless sensor networks (WSNs) [1] have attracted tremendous attention of the research community in recent years. A vast amount of research work has been conducted to solve the practical and theoretical issues that are still open, which has resulted in a surge of civil and military applications over the last few years. Most deployed WSNs measure scalar physical phenomena like temperature, pressure, humidity, or location of objects. In general, most sensor networks are designed for delay - tolerant and low - bandwidth applications.

The recent availability of inexpensive hardware, for example, CMOS cameras and microphones, which can ubiquitously capture multimedia content from an environment is fostering the development of WMSNs [2], which consist of sensor devices that are interconnected by a wireless communication channel and allow retrieving video and audio streams, still images, and scalar sensor data. With rapid improvement and miniaturization in hardware, a single sensor device can be equipped with audio and visual information collection modules. In addition to the ability to retrieve multimedia data, a WMSN will also be able to store, process in real time, correlate, and fuse multimedia data originated from heterogeneous sources. WMSNs deploy a certain number of multimedia sensors(image and video) [3, 4, 5, 6]in conjunction with a large number of scalar sensors (i.e., temperature, motion, light, etc.) to collect and process multimedia data.

WMSNs can produce a high volume of multimedia data whose handling requires higher bandwidth and excessive CPU processing. So opposed to scalar sensors which can be sensing all the time, multimedia sensors cannot be collecting and transmitting multimedia data all the time as this will quickly drain their battery power. Therefore, it will be wiser to turn them on whenever an event is detected within the vicinity by means of the scalar sensors. When an event takes place in a monitored region, it is first of all detected by the scalar sensors. The scalar sensors inform their corresponding camera sensors regarding the occurring event. Then the camera sensors collaboratively exchange their reading to decide which among them to be actuated by distributed camera actuation based on scalar sensor count(DCA-SC) [7]. [7] shows that counting the number of scalar sensors is a reasonable way to determine the size of the event areas when a randomly uniformly deployed network is considered. Data redundancy occurs due to over lapping of Field of view (FOV) of camera sensors. Due to data redundancy communication cost in terms of bandwidth used, CPU processing etc. increases. Camera sensor's Field of view(FoV) represents the angle at which camera sensor can capture accurate image of an object. Elimination of redundant data is a crucial issue in case of WMSN. As redundancy causes transmission of same data repeatedly, it needs to be eliminated so as to reduce communication cost in terms of unnecessary energy wastage, bandwidth used and CPU processing etc.

In this paper, we propose a modified distributed actuation scheme which strives to turn on the least number of cameras to avoid possible redundancy in the multimedia data while still providing the necessary event coverage. Even though the camera sensors have heard from scalar sensors about an occurring event, they may not cover the event. These nodes unnecessarily undergo distributed camera actuation scheme and some or all of them are activated. As some of the scalars that lie outside of the event boundary sense the occurring event though they lie outside the event boundary, still their sensing range cover the event region. After detecting the event the scalar sensors inform their corresponding camera sensors regarding the occurring event. Being informed from the scalar sensors the camera sensors undergo distributed camera actuation scheme and some or all of them are actuated unnecessarily even though they do not cover the event region. So our objective is to keep such cameras in turned off condition and to activate optimum number of camera sensors for adequate coverage of the event region in such a manner that no event information will be missed.

In simulations, we compared the performance with [7] in terms of the number of activated camera sensors, and the coverage ratio. The results show that our approach decreases the number of actuated(active) camera sensors while maintain the same coverage.

This paper also apply the distributed actuation schema in monitoring the event region boundary. We apply the distributed actuation schema on the boundary camera sensors to activate the least number of camera sensors cover the event region boundary. Our work depends on the assumption that the camera sensor is said to be boundary camera sensor(on the event region boundary)if its FoV has a boundary scalar sensors located in it. In simulation, we compare our performance with the maximum coverage (where we compute the coverage when all the camera sensors within the network are actuated) in terms of the number of actuated(active) boundary camera, and the coverage ratio. The results show that the number of actuated(active) boundary camera sensors is decreased while maintain the same boundary coverage.

This paper is organized as follows. In the next section, we summarize the related work. Section 3 states our considered network model, assumptions and problem definition. In Section 4, details the modified distributer actuation algorithm with the pseudo-code. Section 5 includes the simulations to assess the performance of the distributed approach under a variety of conditions. In Section 6, we propose a new approach for event boundary detection and monitoring. In Section 7, we propose the experimental results of event boundary detection approach. Finally, Section 7 concludes the paper.

2. RELATED WORK

The redundant data elimination is a critical issue in case of WMSN. As redundancy causes repeatedly transmission of same data which affects badly on communication cost in terms of unnecessary energy wastage, bandwidth used and CPU processing etc. To extract the redundant data, an overwhelming majority of the studies in WMSNs are related to processing, compression or distributed coding of multimedia data for increased network lifetime. The idea is to combine the data coming from different sources enroute, eliminating redundancy, minimizing the number of transmissions and thus saving energy [1, 8, 9] and to deliver multimedia content with predefined levels of quality of service (QoS). [10, 11].

Art gallery problem can be used to determine the least number of nodes and their locations in order to provide full coverage of the monitored region. The problem can be solved in polynomial time in two dimensional(2D) environment[12]. For art gallery solution, the camera sensors are deployed manually, assuming that the the deployment regions topology for and scalar sensors within the WMSN are known in advance. Then, the locations of camera sensors should be made available to scalar sensors so that they can know which camera sensor to actuate when an event occurs. So the solution for art gallery problem cannot be used for our problem if WMSN is arbitrarily deployed.

Some works consider FoV coverage as a special case of circular coverage used in WSNs and such networks are referred to as directional FoV sensor network. The work in [13] investigates a suite of novel problems related to connected coverage in directional sensor networks where sensors only sense directionally and have a sector-like sensing range. [13] considers a scenario where directional sensors can be precisely deployed at any location within a target sensing area, and address the following two problems, Connected Point Coverage Deployment and Connected Region-Coverage Deployment, which aim to use a minimum number of directional sensors to form a connected network to cover a set of point locations and the entire target sensing area, respectively. The camera connectivity in our problem is not considered. In our problem we have scalar sensors which can be used support the camera connectivity when necessary. Reference [14] presented algorithms for randomly deployed directional sensor networks to identify a minimal set of directions to cover the maximal number of point-locations. In [15], several scheduling algorithms were presented to divide a randomly deployed sensor network into subsets to alternatively cover a set of point-locations so as to prolong network lifetime. In [16], the authors analyzed the probability that a point-location is sensing by a randomly deployed directional sensor networks(not effectively sensed). However, the problems, objectives, and algorithmic approaches presented in this paper are completely different from these earlier research efforts.

[17] focus on the deterministic placement of cameras in a dynamic and occluded scene, aiming at optimal positioning with the minimal number of cameras to cover an area of interest, assuming cameras with unchangeable orientation after deployment. It is considered the presence of obstacles (trees, furniture, columns, moving people, etc.) that generate occlusion in cameras FoV and may interfere in the final coverage of the monitored field.

[18, 19] concerned with the minimal and optimal acquisition of images to cover an area. Such computer vision problems

are relevant for coverage calculation and optimization in video-based wireless sensor networks, but still demand additional computational solutions and novel research to deal with ad-hoc battery-constrained wireless sensor cameras. Another Work [20] which an entropy-based analytical framework is developed to measure the amount of visual information provided by multiple cameras in the network. By contrast, cameras nodes that are selected to transmit data to the sink are that nodes that provide the highest entropy. This works study the correlation characteristics of visual information in WMSNs which was not our concern in this work.

The work in [21] presents the coverage of WMSNs. Instead of dealing with the placement of the cameras, this work deals with the orientation of the cameras. The authors proposes the fully distributed algorithm to adjust the orientations of the multimedia sensors, multimedia sensor nodes update the orientation of multimedia sensors on the fly to increase the multimedia coverage significantly by avoiding overlaps and occlusions. In this algorithm, the cameras are selected to actuate rather than to pan.

The authors in [7] propose a distributed solution to camera actuation problem in WMSNs with almost negligible message overhead on camera sensors. The idea is for each camera sensor to utilize the number of scalar sensors which detected an event within its FoV and exchange this information with the neighboring camera sensors to determine the possible coverage overlaps. This work makes a number of cameras which are not cover the event region unnecessarily undergo distributed camera actuation scheme and some or all of them are activated as illustrated in section 4. So our objective is to keep such cameras in turned off condition and to activate optimum number of camera sensors for adequate coverage of event region in such a manner that no event information will be missed and redundancy can be eliminated. To do that, we add a new restriction to the camera sensor to run a distributed camera actuation scheme. This restriction is that after the camera sensor is informed about the event occurrence, the camera node will undergo distributed camera actuation scheme if its FoV intersect the event region. After that restriction, the actuated cameras are those cameras that event region is located under their FoVs.

3. NETWORK ASSUMPTIONS AND PROBLEM DEFINITION

In our work, we obtain the same assumption in [7]. In this paper, the assumptions can be mentioned as follow:

(1) We assume that the scalar and camera sensors are randomly deployed. Both the cameras and scalars are assumed to have fixed positions as seen in Fig.1.

(2) A Multimedia(camera) sensor node v is represented by the FoV of its camera. The term FoV refers to a directional sensing area of multimedia sensor node, which is approximately hypothesized as an isosceles triangle in two-dimensional space. The 2D model of a video sensor node where the FoV is defined as a isosceles triangle denoted by a 4-tuple v(P,Rs,θ,α). Where P is the position of v, Rs is its sensing range, α is the Orientation angle of the cameras FoV which determines the sensing direction, and θ is the Vertex angle of the FoV as shown in Fig. 2.

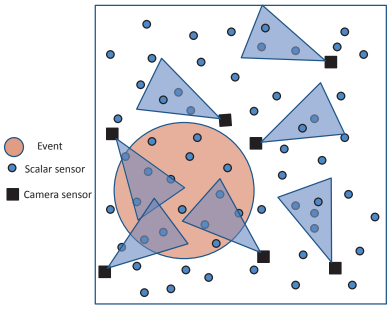

Fig. 1: Network model.

(3) The sensing range and the orientation of FoV of the camera sensor are assumed to be random.

(4) We assume that all of the scalar sensors will detect and broadcast at the same time.

(5) We assume that all sensors know their locations via using one of the techniques from the literature [22].

(6) It is assumed that all the sensors can communicate with each other independent of the type of sensor as long as they are within the transmission range of each other.

(7) one event is assumed to occur at a time. The events are assumed to appear randomly within the monitored area.

(8) Event boundary are assumed to be circular for easy implementation. The event boundary can be represented by a circle or a polygon. Circle shape is assumed for easy implementation in our context.

The event boundary can be determined by scalar sensors sitting around the boundary [23, 24]. The boundary scalar sensor(scalar sensor locates on the boundary of the event region) is assumed to detect the event. While scalar sensors will detect the events, Scalar sensors inform their corresponding camera sensors regarding the occurring event by broadcasting DETECTION message[7]. Multimedia sensors will be turned on to capture a video or image of the event whenever needed. We will refer the multimedia sensors as camera sensors thereafter.
In this paper, it is considered that camera sensors and the scalar sensors broadcast a notification message that contains their id and location information. As a result all the sensors can know position of each other. The actuation algorithm is completely distributed and requires only 1-hop communication for the nodes[7].

4. PROBLEM DEFINITION

The considered problem can be defined as follows: Consider a WMSN with **n** scalar sensors and **m** camera sensors. Initially all the scalar and camera sensors are randomly deployed and they can not move. our goal is to actuate (i.e., turn on) the proper camera sensors when an event is detected in order to minimize the coverage overlaps among them in a low-cost and distributed manner while still striving to maintain the adequate coverage for the event. The considered problem can be defined as follows. Consider Figure 3, where the small circles represent the scalar sensors and black small squares v1, v2, v3, v4, v5, v6, v7, and v8 represent the camera sensors. Their field of view (FOV) is represented by the a triangle. The pink circle represents the event region. The largest dashed

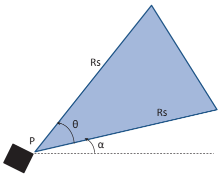

Fig. 2: FoV coverage model.

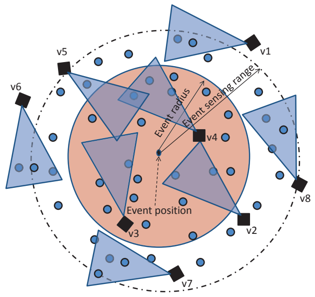

Fig. 3: Proposed example.

circle represents the sensing area of the event(event sensing range represents the distance up to which sensing of event occurs). Four types of tables are maintained at each camera sensor[7]. FSN(FoV Scalar Nodes) table contains the IDs of scalars that lie within the FOV of camera sensor. DSN(detecting scalar Nodes) table contains the IDs of scalars that lie within the field of view of camera which are detecting the event. PCN(priority camera node) table contains the IDs of activated camera sensor(s). ACS (actuated camera's scalars) table contains the IDs of neighboring scalar sensors within FoV of actuated camera sensors so far.

When an event takes place, a scalar sensor will detect the event and broadcast a DETECTION message containing its own ID. A camera sensor will record the ID of the scalar sensor that sent the message in its DSN table only if the scalar sensor is within its FSN table. Each camera sensor will then send an INFORM message to share the number of scalar sensors within its FSN with all of its neighbors. A camera sensor receiving such an INFORM message will add the ID of the camera sensor to its PCN table only if the $|DNS|$ received is greater than its own $|DNS|$. After exchange of INFORM message the camera sensors maintain a priority list that contain the DNS value of each of the camera sensors. So after INFORM message exchange, the $|DNS|$value of each of the camera sensor becomes available with other camera sensors. The camera having maximum number of detected scalar sensors ($|DNS|$) is activated first. The activated camera sensor sends UPDATE message to other camera sensors. UPDATE message contains the IDs of scalar sensors that lie within the DNS table of concerned camera sensors. Based on matching the IDs contained in UPDATE message and the IDs of scalars lying within DNS table of camera, the camera sensors decide which will be activated next. In Figure 3, the nodes v1, v6, v7, and v8 do not cover the event. But here being informed from event detecting scalar sensors. But activation of the nodes v1, v6, v7, and v8 are unnecessary as their FoVs do not cover the actual event region. The nodes v1, v6, v7, and v8 unnecessarily undergo distributed camera actuation scheme and some or all of them are activated. So our objective is to keep such cameras in turned off condition and to activate optimum number of camera sensors for adequate coverage of event region in such a manner that no event information will be missed and redundancy can be eliminated .

5. MODIFIED DISTRIBUTED ACTUATION ALGORITHM

The basic idea of distributed actuation algorithm[7] is to exchange the amount of covered event area among the camera sensors and let them decide to be actuated based on the overlaps among their neighbors and certain predefined thresholds. The size of the covered event area for a camera sensor will be determined based on the number of scalar sensors which detected the event and their positions are located in the FoV area of that camera sensor. Using the count of scalar sensors for determining the event coverage of a camera is intuitive as the ratio of expected number of sensors within the covered event area to the total number of sensors within the FoV of that camera will be same as the ratio of the corresponding areas. The complete pseudo-code for the algorithm is shown in Algorithm 1. Each camera sensor runs this algorithm.

In this paper, at the beginning, the camera sensors broadcast CIM (camera information message) and scalars broadcast SIM (scalar information message) which contains their id and location information. Each camera sensor initialize its FSN table to contain the scalar sensors located in its FoV, DNS, PCN, and ACS tables to be empty (see algorithm 1: lines 1-2).

When an event takes place in a monitored region it is first of all detected by the scalar sensors. Scalar sensors inform their corresponding camera sensors regarding the occurring event by broadcasting DETECTION message[7]. In our work, to eliminate the activation(actuation) of a camera sensors which do cover the event region, we assume that the scalar sensor will send in the DETECTION message the information of the event to their camera sensors. After the camera sensor receive the message from the scalar sensors, it will examine if its FoV intersects the event region or not. If the camera sensor intersect the event region it will undergoes to the undergo distributed camera actuation scheme. Otherwise, the camera sensor is not activated and it broadcasts SLEEP message(see algorithm 1: lines 3 - 5).

Once a camera sensor hears from the scalar sensor (receives the DETECTION message contains the scalar sensor's ID), the camera sensor will register the ID of the scalar sensor that sent the DETECTION message in its DNS table if the scalar sensor is located in its FoV(see algorithm 1: lines 8 - 10). We assume that all of the scalar sensors will detect and broadcast at the same time. After the camera sensor constructs its DNS table, it broadcasts the number of such scalar sensors (i.e., the sensors which detected the event) to all of its neighboring camera sensors (see algorithm 1: lines 11 - 13). After this process, the camera sensors create priority lists for being actuated based on the number of scalar sensors. Basically, a camera which has more detected scalar sensors will be given priority in actuation. A large number of scalar sensors indicate that FoV of that camera can possibly capture a large portion of the event. In the case that two camera nodes have the same cardinality of DNS, the node that has the large ID will be given priority in actuation (see algorithm 1: lines 14 - 18).

As soon as a camera sensor decides to be actuated(has the largest cardinality of DNS, or its PCN is empty), it broadcasts another message(UPDATE message) to update its neighbors(see algorithm 1: line 20). This UPDATE includes the IDs of scalar sensors which are within the FoV (DNS elements) of the camera sensor. The cameras with lower priorities should wait for their neighboring cameras having higher priority before they decide to be actuated or not. The camera sensor will update its ACS and PCN tables. Basically, the ACS table is updated by adding new elements of the received ACS and PCN table is updated by removing the ID of the camera sensor which sent the UPDATE message (see algorithm 1: lines 21 - 22). The process will repeat until the PCN table becomes empty. Before, the camera sensor take its decision to be actuated or not. It can check whether its scalar sensors have already been included in the UPDATE message or not. If a significant portion of its scalar sensors has already been included, then the camera sensor may decide not to actuate. A threshold is set based on the application for the number of scalar sensors within FoV of a camera sensors providing exclusive coverage for the event (see algorithm 1: line 25). The smaller this a value, the more likely a camera sensor will decide to be actuated. After that, If the camera sensor satisfies the threshold value, it will be actuated, it will update its ACS table to contain its detected scalar sensors in its FoV(DNS elements). Then, the camera sensor will broadcast the UPDATE message (see algorithm 1: lines 25 - 28). Otherwise it will broadcast a SLEEP message(see algorithm 1: line 30)

5.1 Algorithm outline

6. EXPERIMENTAL RESULTS

In this section, we assess the performance of the our actuation schema under a variety of conditions. We also compare it with [7]. We also assess the performance of boundary detection and monitoring approaches.

6.1 Simulation setup

For simulations, to reduce the impact of randomness, we generated different WMSNs with different simulation seeds while varying the number of camera sensors, the number of scalar, event radius, and the threshold value. Simulations are performed under 20 different topologies and the average of these were recorded. The location of the event is chosen randomly. Scalar and camera sensors are deployed randomly. They are assumed to have fixed positions. Sensors are randomly deployed in 120× 120 square region. Nodes have equal communication and sensing ranges of 20m, an AoV of $\theta = 60^\circ$ random position P and random orientation angle α(direction). The default parameters are set in Table 1 unless otherwise specified. We considered the following performance metrics:

—Number of activated camera nodes

—Coverage ratio percentage: Defined as the percentage of the area of an event which is covered by all activated camera sensors with respect to its total area. To compute it, we set a 10000 number of random points in the event region and compute the points that are covered by the activated cameras' FoVs. The cover ratio is computed as the percentage of the covered point to the total number of points. The point is covered, if it is located in a camera' FoV. One exact method for determining whether a point is inside the FoV triangle is the method described by Moreno in [25]

In our simulation, we evaluate our approach with the approach in [7].

Algorithm 1 Distributed actuation algorithm

This Algorithm is run by each camera sensor, suppose camera sensor i .

```
1:  Initialize Table $FSN_i$ to all scalar sensors within the FOV.
2:  Initialize Tables $DNS_i$, $ACS_i$, and $PCN_i$to be empty.
3:  while receiving DETECTION messages do do
4:    if Intersect(Event, i)= false then
5:      Broadcast the SLEEP message.
6:      break.
7:    else
8:      if DETECTION message received from j, AND j $\in FSN_i$ then
9:        Add j to $DNS_i$ .
10:     end if
11:     if $|DNS_i| \neq 0$ then
12:       Broadcast Inform message.
13:     end if
14:     while receiving INFORM messages do
15:       if INFORM message received from j, and $|DNS_j| < |DNS_i|$ then
16:         Add j to $PCN_i$ table.
17:       end if
18:     end while
19:     while $PCN_i > 0$ do
20:       if Update Message received from j then then
21:         $ACS_i = ACS_i \bigcup ACS_j$.{add neighbor scalars of j to neighbors of i}
22:         $PCN_i = PCN_i - j$.{delete j from $PCN_i$ table}
23:       end if
24:     end while
25:     if Threshold≤ $\frac{|DNS_i - ACS_i|}{|FSN_i|}$ then
26:       Actuation=true.
27:       $ACS_i = ACS_i \bigcup DNS_i$. {add the detected scalar sensors of i to $ACS_i$ table}
28:       Broadcast UPDATE message.
29:     else
30:       Broadcast SLEEP message.
31:     end if
32:   end if
33: end while
```

Table 1. : Simulation Parameters

m	n	R_c	R_s	Threshold	Event Radius
100	1500	20	20	0	25

6.2 Effect of varying number of camera sensors

We varied the number of camera sensors keeping all other parameters constant. We varied the number of camera sensors to see its effect on the number of activated camera nodes and the coverage ratio. The results in Fig.4 show that, as expected, the number of activated camera nodes increases when the number of camera sensors is increased. As the number of camera nodes increases, the number of camera nodes deployed in the event region will be increased which lead to increasing the number of activated camera nodes. As shown in the Fig.4, the number of cameras activated in our work are found to be less than that of other work in all the cases. This is because we add a new restriction to the camera node to be active that its FoV must intersect with the event region which extract the nodes that have a detect scalar nodes but its FoV does not cover the event region. The results in Table 2 show that, as expected, the coverage increases when the number of camera sensors is increased. The coverage ratio of the proposed work is the same with the other work because in our work, we extract the camera nodes that its FoV intersect to the event region while in the other work the camera node that has a number of detect scalar nodes satisfies the threshold rule will be active even if its FoV does not cover the event region. Therefore it is concluded that proposed approach is found to be better than the other work.

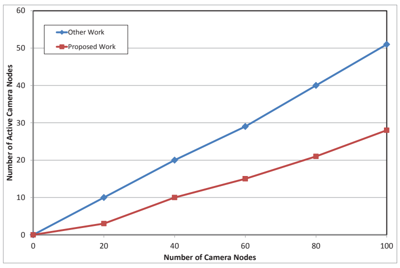

Fig. 4: Number Of Camera Nodes Vs. Number Of Cameras Activated.

Table 2. : Coverage Ratio Comparison

Number of Camera Nodes	20	40	60	80	100
Other work	13.487	27.266	39.825	43.661	67.697
Proposed Work	13.487	27.266	39.825	43.661	67.697

6.3 Effect of varying number of scalar sensors

We varied the number of scalar sensors keeping all other parameters constant. We varied the number of scalar sensors to see its effect on the number of activated camera nodes and the coverage ratio. Consider other work and proposed work, the results in Fig.5 show that in both the cases the number of cameras activated remains semi constant. This is because of the threshold rule which made the increasing of the number of scalars does not effect on the number of activated cameras. The coverage ratio percentage is constant in all cases and the same on both works and equal to 67.7.

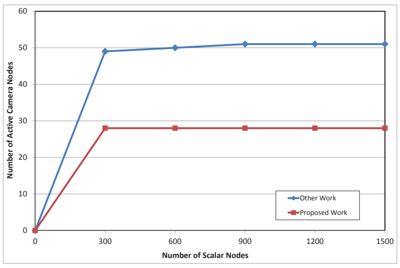

Fig. 5: Number Of Scalar Nodes Vs. Number Of Cameras Activated.

6.4 Effect of varying event radius

We varied the event radius keeping all other parameters constant. We varied the event radius to see its effect on the number of activated camera nodes and the coverage ratio. Consider other work and proposed work, the results in Fig.6 show that in almost cases with increase of event radius the number of cameras activated increases. As the event radius increases the number of cameras located in the event region will be increased which leads to increasing the number of activated cameras. The results in Table 3 show that the coverage ratio of the two works as the event radius increases.Both works have the same coverage ratios. The coverage ratio initially clearly decreases as the event radius increases and then it remains almost constant or lightly increases in both the cases. As the event radius increases, the area of event to cover is also increases. So the increasing of the number of activated cameras does not affect purely on the coverage ratio.

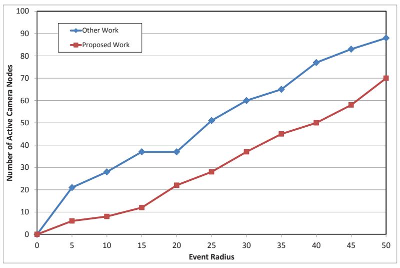

Fig. 6: Event Radius Vs. Number Of Cameras Activated

6.5 Effect of varying Threshold value

We tried three different values threshold as we varied the number of camera sensors to see its effect on the number of activated cameras and the coverage ratio. The results in Fig. 7 showed that, the number of activated camera nodes decreases as the value of threshold increases. Note that when threshold is set to 0.5 and 1, the number of activated cameras is almost constant.

The results in Fig. 8 show that, the coverage ratio decreases as the value of threshold increases. The coverage is highest and when threshold= 0 as expected. Note that when threshold is set to 0.5 and 1, the coverage ratio is almost constant. This mean that the cameras that are activated when in threshold=0.5 are activated also when threshold=1. because these nodes are the nodes that satisfy the threshold rule and its FoV intersects with the event region.

Table 3. : Coverage Ratio Comparison

Event Radius	5	10	15	20	25	30	35	40	45	50
Other work	99.993	83.921	62.591	59.469	67.697	71.516	70.861	71.351	70.529	69.576
Proposed Work	99.993	83.921	62.591	59.469	67.697	71.516	70.861	71.351	70.529	69.576

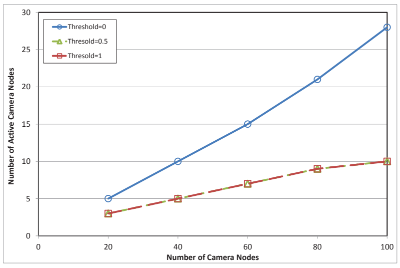

Fig. 7: Number of activated camera nodes of varying camera sensors with different thresholds.

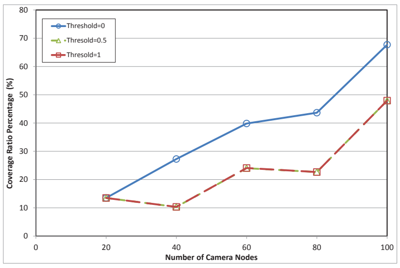

Fig. 8: Coverage ratio of varying camera sensors with different thresholds.

7. EVENT BOUNDARY DETECTION AND MONITORING

Wireless sensor networks can be used to monitor many hazardous environmental events, such as mud flows, forest fires, and oil spills. These events are located in geographic regions, and in many application scenarios it is not possible to continuously collect data from every node in the event region; the monitoring of the event region boundary is often sufficient. For example, in forest fire monitoring, we may be interested in where the fire expanded to or the location of major boundaries of fire[26]. An important application of WSNs is to monitor, detect, and report the occurrences of events of interest [27, 28, 29, 30]. For some large-scale spatial phenomena, such as forest fire, environment temperature, and chemical spills, detecting the event boundary is sufficient or of more importance than collecting measurement over the entire event area. Sensor nodes are expected to collaborate with each other based on each own local view and provide a global picture for spatially distributed phenomena with greatly improved efficiency.

In this section, we introduce event boundary detection and monitoring approach by the camera sensors. We restrict our work on the assumption that the camera sensors is said to be a boundary camera sensor to the event if its FoV contains a scalar sensors that are lied on the event boundary. To determine the boundary scalar sensors, there has been a lot of works to determine the boundary sensors for an event with localized algorithms for Wireless sensor network [23, 24]. We use the approach in [23] which can be detailed as follow: Once an event of interest is detected, senor nodes first exchange their measurements (i.e., 0/1) among neighbors and each node suppresses its own (possibly) faulty measurements following a majority rule. A nonparametric statistical boundary detection model further operates on the error-adjusted measurement results, and identify boundary nodes.
In this section, we use the distributed actuation schema introduced in algorithm 1 to actuate a optimum number of boundary camera sensors to cover the event region boundary. In this Work, only scalar sensors which sit at the boundary of the event can inform the camera sensors about the event. In simulation we compare between two methods:

—the method in which we actuate the camera sensors by using the actuation algorithm in [7](Method 1).

—the method in which we actuate the camera sensors by using the our actuation algorithm 1 (Method 2).

8. EXPERIMENTAL RESULTS

In this section, we evaluate Boundary detection works, the simulation parameters are shown in Table. 1. We considered the following performance metrics:

—Number of activated boundary camera nodes: the boundary camera node is the camera that has a boundary detect scalar nodes(that are located in the boundary of the event region) in its FoV.

—Boundary coverage ratio percentage: Defined as the percentage of the area of an event which is covered by all activated camera sensors with respect to its total area. To compute it, we set a 10000 number of random points in the event boundary and compute the points that are covered by the activated cameras' FoVs. the cover ratio is computed as the percentage of the covered point to the total number of points. One exact method for determining whether a point is inside the FoV triangle is the method described by Moreno in [25]. We compare the values boundary coverage ratio of the works to the **maximum coverage** value which is the percentage of the boundary points that are covered by all boundary camera nodes.

8.1 Effect of varying number of camera sensors

We varied the number of camera sensors keeping all other parameters constant. We varied the number of camera sensors to see its effect on the number of activated camera nodes and the coverage ratio. The results in Fig.9 showed that, as expected, the number of activated camera nodes increases when the number of camera sensors is increased. As the number of camera nodes increases, the number of camera nodes deployed in the event region will be increased which lead to increasing the number of activated camera nodes. As shown in the Fig.9, the number of cameras activated of Method2 are found to be less than that of Method1 in all the cases. This is because of a restriction rule (camera's FoV must intersect with the event region)to the camera node to be active which extract the nodes that have a detect boundary scalar nodes but its FoV does not cover the boundary of the event region. The results in Table 4 show that, as expected, the coverage increases when the number of camera sensors is increased. The coverage is highest and equal to the maximum coverage when threshold = 0. The coverage ratio of Method1 is the same with the Method2 because in Method2, the camera nodes are activated if its FoV intersects to the event region boundary while in the other work the camera node that has a number of detect boundary scalar nodes satisfies the threshold rule will be active even if its FoV does not cover the event region boundary. Therefore it is concluded that method2 is found to be better than the method1.

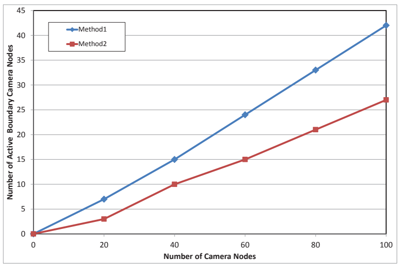

Fig. 9: Number Of Camera Nodes Vs. Number Of Boundary Cameras Activated.

Table 4. : Boundary Coverage Ratio Comparison

Number of Camera Nodes	20	40	60	80	100
Maximum Coverage	20.9	45.5	54.3	72.9	82.6
Other work	20.9	45.5	54.3	72.9	82.6
Proposed Work	20.9	45.5	54.3	72.9	82.6

8.2 Effect of varying number of scalar sensors

We varied number of scalar sensors keeping all other parameters constant. We varied the number of scalar sensors to see its effect on the number of activated camera nodes and the coverage ratio. Consider METHOD1 and method2, the results in Fig.10 showed that in both the cases the number of cameras activated remains semi constant. This is because of the threshold rule which made the increasing of the number of scalars does not effect on the number of activated cameras. The coverage ratio percentage is constant in all cases and the same on both works and equal to 82.6.

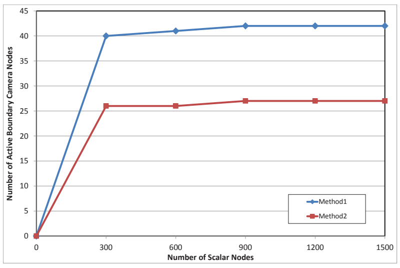

Fig. 10: Number of scalar nodes Vs. Number of boundary cameras activated.

8.3 Effect of varying event radius

We varied the event radius keeping all other parameters constant. We varied the event radius to see its effect on the number of activated camera nodes and the coverage ratio. The results in Fig.11 show that in almost cases with increase of event radius the number of cameras activated increases. As the event radius increases the number of cameras located in the event region will be increased which leads to increasing the number of activated cameras. The results in Table 5 show that the coverage ratio of the two works as the event radius increases. The coverage ratio initially clearly decreases as the event radius increases and then it remains almost constant or lightly increases in both the cases. Both works have the same coverage ratios.

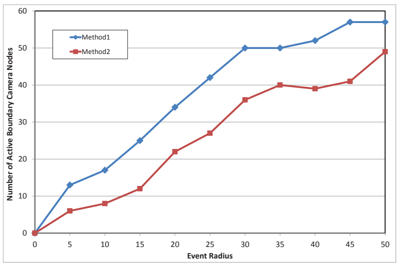

Fig. 11: Event radius Vs. Number of boundary Cameras Activated.

8.4 Effect of varying Threshold value

We tried three different values threshold as we varied the number of camera sensors to see its effect on the number of activated cameras and the coverage ratio. The results in Fig.13 showed that, the number of activated camera nodes decreases as the value of threshold increases. Note that when threshold is set to 0.5 and 1, the number of activated cameras is almost constant in Method2 because of the intersection rule(camera's FoV must intersect the event region).
The results in Fig.15. showed that, the coverage ratio decreases as the value of threshold increases. The coverage is highest and when threshold= 0 as expected. Note that when threshold is set to 0.5 and 1, because of the intersection, the coverage ratio is almost constant in Method2. This mean that the cameras that are activated when in threshold=0.5 are activated also when threshold=1.

Table 5. : Boundary Coverage Ratio Comparison

Event Radius	5	10	15	20	25	30	35	40	45	50
Maximum Coverage	98.7	88.5	80.8	70.05	82.6	79.8	67.6	55.8	64.9	63.9
Other work	98.7	88.5	80.8	70.05	82.6	79.8	67.6	55.8	64.9	63.9
Proposed Work	98.7	88.5	80.8	70.057	82.6	79.8	67.6	55.8	64.9	63.9

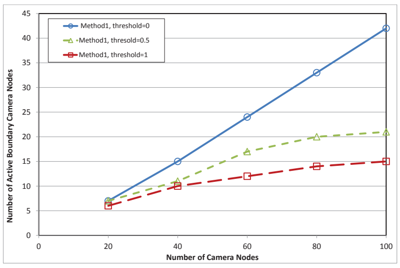

Fig. 12: Method1: Number of activated camera nodes of varying camera sensors with different thresholds.

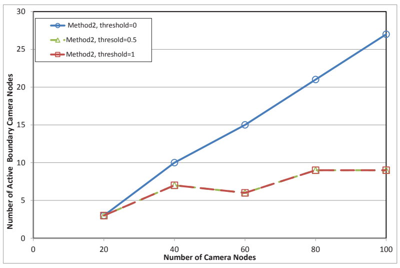

Fig. 13: Method2: Number of activated camera nodes of varying camera sensors with different thresholds.

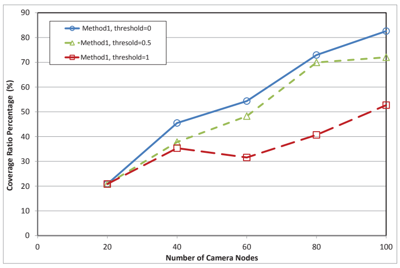

Fig. 14: Method1: Coverage ratio of varying camera sensors with different thresholds.

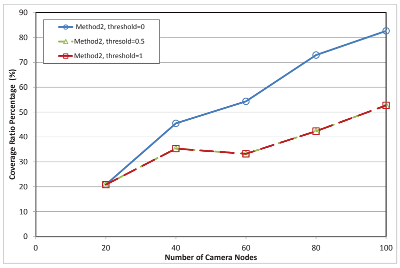

Fig. 15: Method2: Coverage ratio of varying camera sensors with different thresholds.

9. CONCLUSION

Elimination or reduction of the redundant data is one of the critical issues in decreasing the transmission costs of the multimedia data as much as possible. Distributed camera actuation achieves redundant data elimination in which a proper number of camera sensors actuated for adequate coverage of event region. This paper aim to decrease the number of actuated camera sensors as much as possible to prevent the waste of retransmission of the same data(redundant data) by prevent the overlapped camera sensor to be actuated simultaneously. This paper also prevent the camera nodes which are located in the event sensing region and its FoV are not cover the event region to be actuated by add a new restriction to the actuation process. this restriction is that the camera node is actuated if its FoV intersects the event region. When the case is considered of sensing of occurring event at outside, up to certain distance of event region, using the proposed approach only required optimum number of camera sensors in the event region are activated and all other cameras those lie outside event region but inside the sensing range of event are keeping in turned off condition. By studying different cases such as varying number of scalars, number of cameras, and event radius and studying their effect on number of cameras activated, it is observed that the number of cameras activated in proposed approach in most of the cases is found to be less than the number of cameras activated in the other work while persevering the certain coverage. The simulations results have shown that a good coverage performance when all the cameras within the region or within the vicinity of the event are actuated(i.e., threshold = 0) will come with a lot of overlaps which results in unnecessary multimedia data transmission from the nodes. In boundary detection approaches, the coverage ratio is equal to the maximum coverage when threshold=0 (when all the boundary cameras within the region or within the vicinity of the event are actuated), it is observed that Method2 is more efficient than Method1 in terms of the number of actuated boundary camera sensors.

10. REFERENCES

[1] I. F. Akyildiz , W. Su , Y. Sankarasubramaniam , Y. , and E. Cayirci , Wireless sensor networks: A survey , Computer Networks (Elsevier) Journal , vol. 38 , no. 4 , Mar. 2002 , pp. 393 422 .

[2] I. F. Akyildiz , T. Melodia , and K. R. Chowdhury , A survey on wireless multimedia sensor networks , Computer Networks (Elsevier) , vol. 51 , no. 4, Mar. 2007 , pp. 921 960 .

[3] M. Rahimi, R. Baer, O.I. Iroezi, J.C. Garcia, J. Warrior, D. Estrin, M. Srivastava, Cyclops: in situ image sensing and interpretation in wireless sensor networks, in: ACM Conference on Embedded Networked Sensor Systems (SenSys), San Diego, CA, 2006.

[4] S. Hengstler, D. Prashanth, S. Fong, H. Aghajan, Mesheye: a hybridresolution smart camera mote for applications in distributed intelligent surveillance, in: Information Processing in Sensor Networks (IPSN-SPOTS), Cambridge, MA, April 2007.

[5] Crossbow IMote2. ¡http://www.xbow.com¿.

[6] A. Rowe, A. Goode, D. Goel, I. Nourbakhsh, CmuCam3: An Open Programmable Embedded Vision Sensor, in: Technical Report, RI-TR- 07-13, Carnegie Mellon Robotics Institute, 2007.

[7] Andrew Newell, Kemal Akkaya.,"Distributed collaborative camera actuation for redundant data elimination in Wireless Multimedia Sensor Networks", AdHoc Networks, vol 9, pp. 514-527, 2011.

[8] K. Akkaya, M. Demirbas, R.S. Aygun, The impact of data aggregation on the performance of wireless sensor networks, Wireless Communications and Mobile Computing Journal 8 (2008) 171193.

[9] Z. Xue, K. Loo, J. Cosmas, P. Yip, Distributed video coding in wireless multimedia sensor network for multimedia broadcasting, WSEAS Transactions on Communications 7 (2008) 418427.

[10] T. Melodia, I.F. Akyildiz, Cross-layer quality of service support for uwb wireless multimedia sensor networks, in: IEEE INFOCOM, Mini- Conference, Phoenix, AZ, April 2008.

[11] L. Savidge, H. Lee, H. Aghajan, A. Goldsmith, Qos-based geographic routing for event-driven image sensor networks, in: Broadband Advanced Sensor Networks (BaseNets), Boston, MA, October 2005.

[12] J. ORourke, Open problem from art gallery solved, International Journal of Computational Geometry and Applications 2 (1992) 215 217.

[13] X. Han, X. Cao, E. Lloyd, C.-C. Shen, Deploying directional sensor networks with guaranteed connectivity and coverage, in: 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2008, SECON 08, 2008, pp. 153160. doi:10.1109/SAHCN.2008.28.

[14] J. Ai and A. Abouzeid, Coverage by directional sensors in randomly deployed wireless sensor networks, Journal of Combinatorial Optimization, vol. 11, no. 1, pp. 2141, 2006.

[15] Y. Cai, W. Lou, M. Li, and X.-Y. Li, Target-oriented scheduling in directional sensor networks, in Proceedings of IEEE INFOCOM, 2007.

[16] H. Ma and Y. Liu, Some problems of directional sensor networks, International Journal of Sensor Networks, vol. 2, no. 1/2, pp. 4452, 2007

[17] Mittal, A.; Davis, L. Visibility Analysis and Sensor Planning in Dynamic Environments. In Proceedings of 8th European Conference on Computer Vision, Prague, Czech Republic, May 1114, 2004; pp. 175-189.

[18] Pito, R. A Solution to the Next Best View Problem for Automated Surface Acquisition. IEEE Trans. Patt. Anal. Mach. Int. 1999, 21, 1016-1030.

[19] Mayer, J.; Bajcsy, R. Occlusions as a Guide for Planning the Next View. IEEE Trans. Patt. Anal. Mach. Int. 1993, 15, 417-433.

[20] R. Dai, I. Akyildiz, A spatial correlation model for visual information in wireless multimedia sensor networks, Multimedia, IEEE Transactions 11 (6) (2009) 11481159, doi:10.1109/TMM.2009.2026100.

[21] N. Tezcan, W. Wang, Self-orienting wireless multimedia sensor networks for occlusion-free viewpoints, Computer Networks 52 (2008) 25582567.

[22] article/cuj geometry article.html, accessed 4/07/2015. N. Patwari, J.N. Ash, S. Kyperountas, A.O. Hero, R.L. Moses, N.S. Correal, Locating the nodes: cooperative localization in wireless sensor networks, Signal Processing Magazine, IEEE 22 (4) (2005) 54 69.

[23] K. Ren, K. Zeng, W. Lou, Fault-tolerant event boundary detection in wireless sensor networks, in: IEEE GLOBECOM, vol. 7, 2006, pp. 354 363.

[24] M. Ding, D. Chen, K. Xing, X. Cheng, Localized fault-tolerant event boundary detection in sensor networks, vol. 2, 2005, pp. 902913. doi:10.1109/INFCOM.2005.1498320.

[25] C. Moreno, http://www.mochima.com/articles/cuj geo metry

[26] Zhong C, Worboys M (2007) Energy-efficient continuous boundary monitoring in sensor networks. Technical report.

[27] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, Wireless Sensor Networks for Habitat Monitoring, ACM WSNA02, Atlanta GA, September 2002.

[28] N. Xu, A Survey of Sensor Network Applications, http://enl.usc.edu/ ningxu/papers/survey.pdf

[29] B. Krishnamachari and S. Iyengar, Distributed Bayesian Algorithms for Fault-Tolerant Event Region Detection in Wireless Sensor Networks, IEEE Transactions on Computers, Vol. 53, No. 3, pp. 241-250, March 2004.

[30] M. Ding, D. Chen, K. Xing, and X. Cheng, Localized Fault-Tolerant Event Boundary Detection in Sensor Networks, IEEE INFOCOM 2005, 13-17 March 2005.