A Low Power Architectural Framework for Automated Surveillance System with Low Bit Rate Transmission

Hanan A. Hosni Mahmoud

Dept. of computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University
Riyadh, 11671 KSA

Abstract: The changed security scenario of the modern time has necessitated increased and sophisticated vigilance of the countries' borders. The technological challenges involved in accomplishing such feat of automated security system are many and require research at the components-and-algorithms as well as the architectural levels. This paper proposes an architectural framework for automated video surveillance comprising a network of sensors and closed circuit television cameras as well as proposing algorithmic/component research of software and hardware for the core functioning of the framework, such as: communication protocols, object detection, data-integration, object identification, object tracking, video compression, threat identification, and alarm generation. In this paper, we are addressing some general topological and routing features that would be adopted in our system. There are two types of data with regard to data communication – video stream and object detection. The network is broken down into several disjoint, almost equal zones. A zone have one or more one cluster. A zone manager is chosen among the cluster heads depending on their relative residual energies. There are several levels of control that could be implemented with this arrangement with localized decision made, to get distributed effect at all levels. A cell tracks each target in its zone. If the target moves out of the range of a cell, the cell manager will send the target description to estimated next cell. The next cell starts tracking the target. If the estimated cell is wrongly chosen, corrections will be made by the cluster heads to get the new target-tracking. We also propose bitrate reduction algorithms to accommodate the limited bandwidth. One of the main feature of this paper is introducing a Low-Power Low-Bit rate video compression algorithm to accommodate the low power requirements at sensor nodes, and the low bit rate requirement for the communication protocol. We proposed two algorithms the ALBR and LPHSME. ALBR is addressing low bit rate required for sensors network with limited bandwidth which achieves a reduction in Average number of bits per Iframe by approximately 60% in case of low motion video sequences and 53% in case of fast motion video sequences . LPHSME addresses low power requirements of multi sensor network that has limited power batteries. The performance of the proposed LPHSME algorithm versus full search and three step search indicates a reduction in motion estimation time by approximately 89% in case of low motion video sequences (e.g., Claire) and 84% in case of fast motion video sequences. The reduced complexity of LPHSME results in low power requirements.

***Keywords*:** Surveillance System, Architectural framework, High-security systems, Low-Power, Low-Bit rate, Motion Estimation, DCT.

1. Introduction

Nowadays the need for securing the borders and the ports of countries has become crucial. Many efforts have been directed towards securing the borders from potential enemies by using video surveillance. There exists a vacuum for a fully-automated system capable of scene understanding and automated surveillance. The action, in Video Surveillance and Monitoring, was always taken by a human. This, though a useful initial measure, was a simplistic solution to a complex problem – avoiding many technical challenges that beset a possible fully-automated solution. Besides, the Central Operator Unit was responsible of coordinating all the Sensor Processing Units, therefore if the former failed the whole system failed.

This paper proposes an automation of tracking objects on the area of observance and to identify threats and raise the alarm. Each Isolated Processing Node (IPN) with a camera attached will surveillance moving objects. The information on each object, which reports to a Collectively Processing Unit (CO-PU) that further analyzes the information received by isolated unites and determines collectively the probability of a threat. This way, each IPN is proposed to track objects while the CO-PU is deciding which objects pose a threat. We address some general topological and routing features that would be adopted in our system [1]-[2]. There are two types of data with regard to data communication – video stream and object detection. The network is broken down into several disjoint, almost equal zones. A zone can have either one cluster or more than one cluster. If there is a single cluster within a zone, the cluster head acts as a zone manager also. In the case where more than one cluster forms a zone, a zone manager is chosen among the cluster heads depending on their relative residual energies. There are several levels of control that could be implemented with this arrangement with localized decision made, to get distributed effect at all levels. Since there are different levels proper hand off are needed. A cell tracks each target. When the target moves it can move out of the range of a cell. The cell manager sends the target description to estimated next cell. The next cell starts tracking the target. If the estimated cell is wrong corrections are made by the cluster heads to get the new target-tracking cell. In other words local distributed decisions are made and the next level of control, corrects errors. In routing two things need to be considered

1) Tuples associated with the message being routed
2) Tuples associated with each node.

Decisions made about routing are based on these tuples. Some parameters in these tuples can be interchangeably used depending on what sort of algorithm is being used. The message packet typically contains the source address, destination address, and time-to-live, current sender's address. Tuples typically associate with each node are residual energy, Distance to destination, Deviation angle with destination, Packet forwarding ratio, Number of hops to destination, Minimum amount of energy required to get to destination, Interference levels at each node, level of layer with respect to destination (Fisheye concept) [3].

Algorithms for in-building topological control and synchronization were discussed in [4]. The first protocol creates a connected network that shifts connectivity towards nodes with more residual energy. The routing is primarily done through these energy rich nodes. Their second protocol deals with Relay Synchronization MAC protocol that extends the lifetime of a node by allowing it to sleep when not required. It is specifically tailored towards application with low data rate compared to bandwidth available. Building a strong minimum energy topology is an NP-complete problem as proven in [5], where comparison of Incremental Power Heuristics Algorithm with the Minimum Spanning Tree algorithm is also discussed. A cluster based communication protocol is offered in [6]. Mathematical analysis to prove that the Cone based Topology Control Algorithm works in 3-dimension space is provided in [7]-[8]. Other topological references include [9]-[10] Routing algorithms on sensor network had been studied. In [11], the routing method is analyzed statistically. In [12], the authors introduced a routing algorithm for extending Lifetime of mobile sensor network utilizing connectivity. New ideas in low probability, random selection relaying to route the packets reaching a node were discussed in [13]. In [14], the authors surveyed IoT security evolution discussing the challenges and countermeasures. Zone based routing, providing setup that leads to a collision free routing is discussed in [15]. In [16], the authors described a setup of a surveillance system. And introduced distributed target classification and Tracking especially with regard to sensor networks. It uses a location aware data centric approach. Cells that fall in the estimated track of object/target movement are put on alert. Depending on targets detected cells consisting of a number of nodes are created. These nodes coordinate with each other in cooperation of detection of object within view. Once the target moves out of view, a new cell is activated, and the process is repeated.

Low bit rate video compression is very important in communication Videos over low bandwidth networks. The main challenge is to develop high compression rate mechanisms without much sacrificing the signal quality. Indeed, many sophisticated and successful algorithms for video compression are currently available that achieve impressive results. One important aspect is the durability of energy resources at the constituting nodes of this wireless network. It is well understood that battery technology cannot provide adequate supply of energy, and thus tend to utilize energy saving techniques in the design process. Employing the very LBR compression techniques in such environments is not straightforward and is heavy energy consuming. Some of the low power DCT architectures available in literature can be found in [17]-[18]. One of the first DCT implementations is the one that is based on Chen's Fast DCT Algorithm [19] which is characterized by good performance, small silicon area and low power dissipation at the expense of more complicated design.

2. Description of the proposed framework

In this section we are presenting the proposed framework.

2.1 The proposed architecture

The proposed architecture consists isolated nodes and a collective node that is responsible of the isolated nodes. Each IPN detects objects in the surrounding area, while CO-PU collects information from the isolated nodes and performs image fusion, image fusion were described in [20]-[22].

The following sections will explain in detail the function of each element. In normal operation IPN will only send information on the detected objects to optimize bit rate. The connection between the CO-PU and IPNs should be wireless. Each cluster will cover a specific area. Therefore, all the information on a given cluster will be highly related. The system will consist of more than one CO-PU and several IPNs. The IPNs will form clusters as. Highly correlated data will give us more information on a given object and will positively impact the false alarm ratio. The cluster head will take the information from the entire cluster and send it to CO-PU.

The Isolated Processing Node (IPN) identify objects in the surrounding area. Scene. The IPN schema corresponds to the basic wireless sensor network. The functional diagram of each IPN is depicted in Fig. 1. The IPN captures the video and images and perform some processing such as motion detection. The data from neighbor IPNs will be combined with an IPN's own data. Data integration will take effect only on the cluster head. However, partial processing for combining data will take effect on all IPNs. This partial processing will be explained later.

The detection of object is performed by simple background subtraction, which is minimal in arithmetic operation, thus comprises low power dissipation. The result will be collected by the CO-PU for analysis. The data obtained by the neighbor IPNs will track the object on the scene based on visibility.

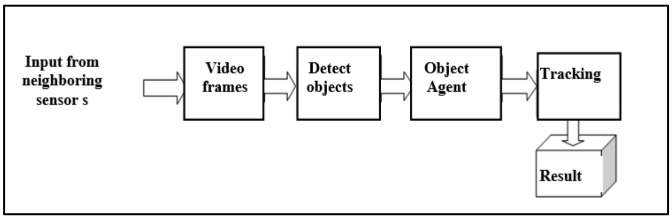

Figure 1. Basic function perform at CO-PU

The interface Outline is depicted as follows:

Camera: image sensor sends out video frames of the scene under observation and receives movement parameters from camera action block.

Object Detection: receives video frames and detect object by means of background subtraction and foreground detection.

Object Agent: gets object detected and assigns a software agent to each detected object for data integration block.

Data Integration: receives information of detected objects from neighbor IPNs and combines it with its own information. If more than one agent is tracking the same object, the agents are merged and sent to the CO-PU.

Tracking: Object agents track the given objects, if the camera has to move to follow the objects, then movement information is sent to camera action block.

Camera Action Blocks: receives movement information and controls the camera based on received data

Video Compression: receives video frames and compress them in order to be sent out to the CO-PU.

Object detection will include background subtraction [23]-[24] under different lighting and foreground identification, as follows:

Preprocessing – consists of changing the raw input video, to reduce camera noise a temporal or spatial smoothing is used. One technique is to use luminance intensity, which is one scalar for each pixel. However color image such as RGB or HSV is more preferable in background subtraction algorithm.

Background Model – uses video frames to calculate and update the background model and capture the entire background scene.

Foreground Detection – it identifies foreground pixels that are different than background pixels. Using the following measure:

$$|I_t(x,y) - B_t(x,y)| > T \quad (1)$$

Where, It(x,y) corresponds to the luminance pixel intensity at time t and Bt(x,y) its background estimate. The threshold T is determined experimentally

Data validation – examines the candidate mask and eliminates those pixels that do not corresponds to the actual moving objects and outputs the final foreground mask

Such information will be sent to CO-PU for collective processing

Frame numbers which indicated start of the moving objects will be send to the motion estimation unit: MEU at the current IPN.

In our approach detection of object must be done on IPNs that are restricted in memory. Consequently, the technique must be suitable for real-time implementation. Our initial study of background subtraction indicates that Mixture of Gaussian (MG) and Wronskian Change Detector Model (WCD) are good options [25]. The first method guarantee accuracy and has been proved that can be implemented in real-time operation but is susceptible to change in illumination, while WCD is less complex and offers good performance and is invariant to illumination. An approach to combine desired feature will help us to research and build our new method.

Low Power High Speed motion estimation: LPHSME will be performed after background subtraction takes place, and is done only on frames where an object is detected. This will lead to reduced complexity and hence lower power consumption. Comparison of the suggested LPHSME with full motion estimation performed on all frames are done showing the resemblance of constructed frames by the two methods, at the receiver site, are almost the same. The average PSNR ratio between the LPHSME to the Full motion estimation is 96:100.

Agent Object Detection

The agent framework binds set of tasks related to a particular object or aspect of the whole problem

Permits and asynchronous, multi-platform instantiation.

For each camera an area agent will control the creation of agent dedicated to collect as much information as possible of the dedicated object and detects the border-region, if it is approached by the object; a switch of agent is in order. The Area agent will detect best neighbor for proper handoff.

Area agent creates Object Agent responsible to track a given object. After the object detection has been performed, foreground regions are detected and object agents are assigned to these regions. The agent will develop an object model consisting of velocity, acceleration, heading based on several frame information. This information must be stored on a Tracking Database associated with agents.

2.2. Dynamic memory architecture

Our approach will consider use of stationary and non-stationary cameras. Non-stationary camera presents a more challenging task since movement parameters must be available for object detection and movement parameters depends on object parameters. To overcome the delay associated to this dependence scheme, our approach will use Dynamic Memory Architecture [26]. A system that tracks multiple objects with an active camera can be decomposed into two modules, visual perception and camera action. In our scheme, visual perception and action module would run in parallel and dynamically exchange information via a specialized shared memory (i.e., the dynamic memory) The dynamic memory will store the camera parameters. When a visual perception module tries to obtain camera parameters and they are not available, the dynamic memory gives an estimate on the parameters based on previously stored values. The same steps are followed when action module tries to read object location parameters and they are not available.

Collectively Processing Unit

Collectively Processing Unit will analyze if the present scene contains a threat pattern. The overall schematic of this unit is illustrated in Fig. 2. The CO-PU has the capability of object classification. The IPN send the object information available at a given time t. Then the object will be decomposed and used on threat identification. Statistical techniques for object classification offer a low complexity approach, independence of orientation and resilience to noise [27]. Therefore, our approach will be centered on Statistical techniques which offer also a low power dissipation feature.

2.3 Communication protocol

In this section, the Communication protocol issues are dealt with. We propose a distributed system in which a certain amount of processing is done in the sensing node itself. This brings about the problem of coordination among nodes to sense, track, and transfer data among the IPNs, and between the IPNs and the CO-PUs. In our system there are two types of sensor combinations. The first is when we deal with only large sensors. These include video cameras, radar. These sensors typically do not have energy constraint or communication bandwidth problems. The second combination is the deployment of a combination of large and small sensors. Small sensors typically include temperature, chemical sensors. These sensors have the problem of energy and bandwidth constraint. Small sensors are deployed by random scattering. A combination of large and small sensor would reduce the energy and bandwidth burden of the small sensors. It would benefit the large sensors network in the ability to cover a larger area without indefinite powering means.

Some issues are:

Sensor Network Robustness: Some sensor may die before others. This phenomenon should not lead to holes in the network and communication breakdown. Since large sensors (LS) are few in number they have a limited ability to compensate in case of a node failure. Therefore, the process of detection and replacement should be prioritized.

Methods of energy conservation in case of small sensors (SS) and LS: In order to have a larger lifetime, energy needs to be conserved. Having an event driven system would extend lifetime.

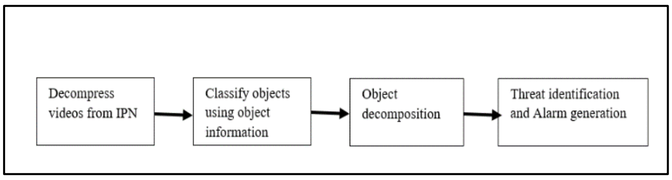

Figure 2. Functional diagram of Threat Identification Task.

Data can have different rates and can be of different types: Collision free environment should be ensured. If there is a collision, proper steps have to be taken to avoid lost data

Sensor Control: Different sensors have different range of views. In order to detect and track multi-target, sensor coordination is required.

Route Generation: Control commands are sent to the various nodes. Nodes send back, detected target information. All these communication typically require efficient route setup.

Priority data detection and priority data route generation: Certain data like LS malfunction; specific object detection could be identified as priority data. This data needs to be detected and sent as fast as possible.

2.4 Uniqueness of the proposed approach

• We approach the whole system with distributed approach. Nodes are deployed in both random and ordered way. We try to maintain connectivity in spite of some amount of randomness.

• We also achieve low power consumption through the different algorithms. This would lead to higher lifetime of a node. A node will be "active" as long there is an event. The rest of the times it is in "sleep" mode that is in a low powered state. That helps in conserving energy.

• Different sensors/nodes have different range of views. The views need to be considered. Each node can detect multi-targets. The nodes can pan, tilt, and zoom for better detection of objects. However since they are detecting more than one object, the pan, tilt and zoom of a node needs to be optimized to detect all objects in their respective tracks.

• Targets/Object once detected, need to be identified with comparison to a given database. However the database is at the CO-PU. To take advantage of the memory of the nodes, a small, distributed database is proposed for certain critical target detection.

2.5 Topological Control with Nodal and Global Energy Optimization

We consider a cluster-based topology. Let us denote the intra-cluster network with Vector set $\{u_i, v_i\}$ and edge set E= $\{e_1, e_2, e_3, \ldots e_n\}$. Let u_i denote the source node with v_i denoting the node that u_i can connect to. If N_u – set of neighbors of a node, D_u - list of directions sorted according to their angles, $dir_u(v)$ – node 'v' direction stored at node 'u'. The following steps are ceased after the source node's incremental energy reaches the maximum possible value, or all cone sectors subtended from the node have a reachable node of v_i. The transmission power of u_i is initially kept at the lowest possible value. It should be noted at this point that there are several source nodes simultaneously executing the algorithm.

1. The transmission power of the source node is augmented incrementally.
2. If any node comes within its transmission radius, the source node broadcasts a "Hello" to that node, sending it its ID and transmission power value to reach that node.
3. Next, the source node includes the discovered node's coordinates (N_u) and which direction (cone sector it was discovered in (D_u, $dir_u(v)$)), in its own routing table.

While this algorithm is being executed at different source nodes, another algorithm is simultaneously executed. The algorithm starts its execution after a delay of t+µ units. This is done, as the second algorithm is an optimizing algorithm that needs network edges to work with. These edges are available after t+µ time units, 't' being the time at which the first algorithm starts execution. The steps are depicted below in Algorithm Topological Control.

Algorithm Topological Control

Start

1. Initialization: S: be a subset of nodes considered so far during execution of the heuristics, P: Total assigned power of all nodes in S, V: is the nodes that are got from the previous algorithm execution at time t+µ, p_u: power expenditure in node u. Initially P = 0.
2. Let S' = V – S. Find u Є S and v Є S' such that connecting u and v needs minimum incremental power δ_p. Check if v falls in the same sector as any node in S.
3. Yes → do not consider it.
4. No → $\delta_p = p_v + \delta_{pu}$, where δ_{pu}, p_v is the increased power required to reach from u to v and v to u, respectively. Set S = S U {v}, P = P + δ_p
5. If S =V, output P and p_v for all v ε V, and then stop; otherwise repeat from Step 2.

Stop

The second algorithm is applied after a delay time µ. The algorithm has a complexity of $O(n^2)$ per node. Reducing the number of edges, reduces the time to a great extent. Since the first algorithm, give a limit on the number of edges at time t+µ, n is reduced at any time instant. Since the two algorithms are implemented simultaneously, the time required to get a strong topologically connected net with minimizing energy, is reduced. The numbers of nodes being considered at a time is decreased, the two algorithms are first applied to each cluster, to get an intra cluster strong topology. Next it is applied at the inter-cluster head level. In conclusion the above algorithms not only try to minimize each sensor transmit power but also tries to minimize the total transmit power and at the same time maintain strong topology.

In order to reduce data traffic and consecutive, energy, if the scene of interest does not change, data is not sent back. In other words the data transfer is event-driven. Having the nodes to go into three different states – sleep, idle and active could address this problem. This would save energy in the case of SS and eliminate unnecessary activity in the case of LS. A metric Forwarding Ratio could be used to change states. Let the Forwarding Ratio be γ where $\gamma_{i,j} = f(n,t)$. $f(n,t)$ is a function that reflects the relationship between scene changes between time i and j having 'n' objects with 't' be a function representing the background. Since we are assuming changing backgrounds, 't' could change and should be taken into consideration in the state changes. The following rules are applied at each node, where thresholds are set by experience.

If $\gamma_{i,j} < \gamma_0$, then the node will switch to *Idle* state.

If $\gamma_{i,j} < \gamma_1$, then the node will switch to *Sleep* sate.

2.6 Camera Control

Different sensors can have different view range. A trade off exists between getting better coverage of targets by multiple sensors and having a manageable number of nodes within a cluster. Asynchronous sensor orientation control management is required since tracking process takes place at a much faster rate than changing the orientation of a sensor node. Prediction of future target position is required. Assuming linear velocity, the target's future position is estimated by having present information by the objects velocity, acceleration in a particular direction or in a random direction. In the case of known direction with a certain velocity the target future position is estimated as follows

$$P_{n+1} = P_n + v_n * \Delta t$$

The uncertainty in the position is assumed to be the original target uncertainty plus the velocity uncertainty, grown as a function of time.

$$\delta P_{n+1} = \delta P_n + \delta v_n * \Delta t$$

When we consider multi-target detection, we need to consider estimated multi-target positions to adjust the camera pan, tilt and zoom. The estimated positions with variations could be thought of sphere with the center being the estimated position. Given the sphere dimensions, the next cameras zoom, pan and tilt could be adjusted to have the cone of view to cover all the spheres. Camera range of view are shown in Fig.3.

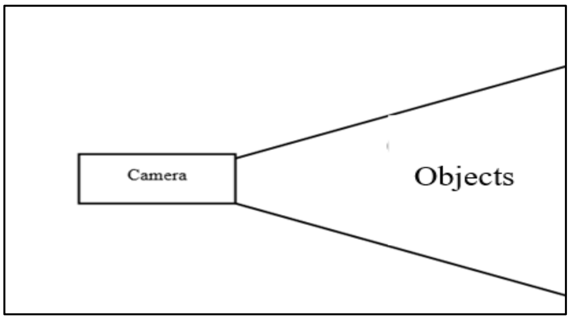

Figure 3. Camera range of view

2.7 Distributed database in a cluster for priority object

The need to generate alarms immediately after detection requires distributed memory scheme. Here an innovative virtual dynamic distributed memory is discussed. Each node has a certain amount of memory. Certain targets need to be detected immediate to generate automatic alarms. To distribute the load of the database among the various nodes, each node detects or looks for only a certain a number of targets. A local database present at each node is called a "page". A "page" can be dynamically changed to insert new targets or delete old ones. A page ID identifies each unique page. Next, each unique page is duplicated among various nodes at regular intervals keeping the page ID the same. Once a target tracking information is checked by a certain page, its ID is tagged to the target information message passed to the next node. If the estimated track of the target covers each of the small databases, a check is made for new updates and the process repeats itself, else the target is tagged with a "complete" ID.

Viruses can be detected by comparison of virus sequences. These sequences could be stored in the local database of each node as explained above. Since virus attacks could bring down a network or a part of network, virus sequence detection database needs to be included within each node. These databases can be updated dynamically including new sequences if needed. While each packet of message is passed from node to node, a virus detection comparison takes place. The cluster head could add or change a new/existing virus sequence page in the estimated path if required. Since the databases are small in each node several copies of each "sequence page" is required.

Priority track generation for high priority data – A priority track is selected using message passing. The message is done to select the shortest path between a cluster head and the CO-PU. A message is send by each cluster head along all its out-going edges. Once the message reaches the CO-PU, it is sent back. The message keeps a track of the number of hops taken and cluster list hopped through and their residual energy (E_{ires}) Assuming that the initial energy that each node starts off with is E_{i0}, the ratio of the residual energy to initial energy is calculated - E_{ires} / E_{i0}. Going in the ascending order of number of hops per route -

If $E_{ires} / E_{i0} < T_{clus}$, choose an alternative route.

If $T_{clus} < E_{ires} / E_{i0} < T_{limit}$, send a "warn" message to the respective cluster head.

If $E_{ires} / E_{i0} > T_{limit}$, select the route.

If the ratio is below a certain threshold (T_{clus}), it indicates that the cluster head may not remain a cluster head for long. The thresholds are calculated from past experience average rate in which a cluster have lost energy. T_{limit} indicates that the cluster can be a part of the priority path only if it does selective transmission of data. A warning message is send to the cluster informing of it being in a priority path. Any value above T_{limit} is a good choice of route. Priority path is kept for priority and emergency data.

3. Proposed Video Compression Technique

In this section we are presenting low bit rate coding to achieve high compression rate for fast transmission. Also, we are expediting the video compression phase by introducing a new fast and efficient search algorithm for motion estimation.

3.1 Low bit rate coding

Coding for low bit rate (LBR) video applications has gained a special interest among the video coding community especially with the emergence of new applications such as *surveillance and monitoring*, videoconferencing, and video telephony. In each case, video and audio information are transmitted over telecommunications links, including networks, telephone lines, ISDN and radio. The main challenge is to develop high compression rate mechanisms without much sacrificing the signal quality. Indeed, many sophisticated and successful algorithms for video compression are currently available that achieve impressive results. One important aspect is the durability of energy resources at the constituting nodes of this wireless network. It is well understood that battery technology cannot provide adequate supply of energy, and thus many researchers have gone into addressing the energy dilemma by utilizing many energy saving techniques in the design process. Employing the very LBR compression techniques in such environments is not straightforward and is heavy energy consuming.

Video processing is computation intensive, and operations such as motion estimation/ compensation and DCT represent most of the computation load, and the major bulk of operations go through these two units [28]. When a current frame is processed, the outcome from a compression structure is motion vectors, DCT-transformation of residual frame, and intra-frame data. In general motion is tracked between two consecutive frames, and the corresponding motion vectors are computed, and the residual image [29]-[30]. In case of a screen cut or new objects appearing in the frame, motion estimation is replaced with intra-based coding. Based on these different compression outputs, the techniques proposed in this work fall into two categories: low bit rate and low power techniques [29]-[31]. The following subsections present the proposed techniques for each category.

The technique for low bit rate depend on choosing only a set B_{sent} of DCT blocks that will be sent to the receiver after quantization. The other blocks will be compared to the x around DCT blocks in B_{sent} and will send the index of the one with the least error.

The technique for low bit rate is summarized as follows:

1. All DCT blocks are quantized
2. We send some blocks only, which are the set of blocks $B_{sent} = \{BLOCK_i, where\ i=j, j+x, j+2x, j+3x, \ldots\ldots, last\}$ see Fig. 4 , 5 and 6.

3. For $BLOCK_x$ which is not sent and does not belong to the set B_{sent} will be calculated as functions of the distance between different blocks and send the nearest index *j* instead of the block itself. For better accuracy, the indices of the best four DCT blocks will be sent. The block with the nearest distance is determined by computing SAD between $BLOCK_x$ and each block in B_{sent}.
4. All blocks in B_{sent} are distinguished by an index number and for an 8x8 quantized-DCT block we transmit its index number instead of transmitting the original block which constitutes much more bits to be transmitted than the index.

A frameblock is added to B_{sent} after it has been sent and marked as sent in the receiver site with an index equal to the last index number plus one.

Several consideration are done for the motion estimation algorithm LPHSME.

First: Assuming the LPHSME will be performed after background subtraction takes place, and is done only on frames where an object is detected.

Second: we had to consider wide search area since a moving object that is considered a threat might be a fast moving one.

3.2 New Three Step Search for Fast Motion

We consider a new three step search algorithm that have two start search step in parallel, the first one with step search S equals to 4 and the other with step search S equals to 8. The first one is to accommodate normal motion, while the second accommodate fast motion. The second one allows the search to go out of the boundary of the search window and diverge in the search if no match has occurred. The total number of points searched is equal to 33 points instead of 25 points in the classical three step search. But the accuracy of finding fast motion is captured in our new algorithm as will be shown in the simulation results in section 5. The new algorithm is illustrated below. The algorithm gives an accuracy compatible as full search but with less computation which leads to less power dissipation. (Declarative step search is shown in Fig. 7)

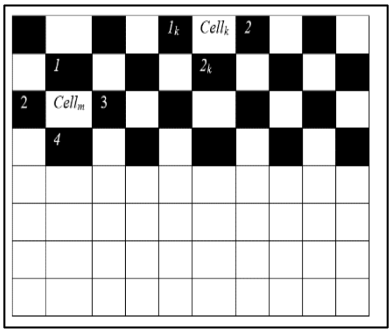

Figure 4. The technique for low bit rate x=2

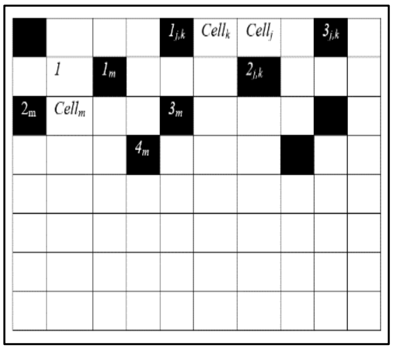

Figure 5. The technique for low bit rate x=4

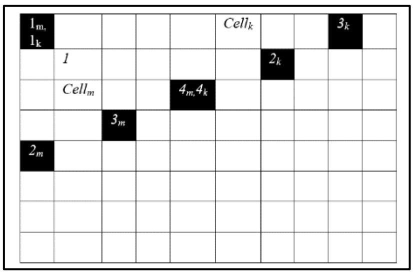

Figure 6. The technique for low bit rate x=8

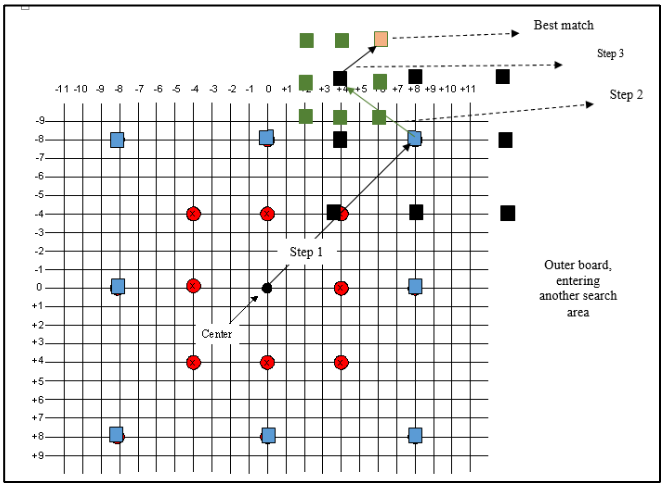

Figure 7. Declarative step search for LPHSME

LPHSME*: New Three Step Search for fast Motion*

Start

1. *Set the search area to be 16 x 16 pixels*
2. *Start with search location at center*
3. *Start with search location at center*
4. *Search the location (0,0) if match found then no movement happened : Set the new search origin to the above picked location and Stop search*
5. *Set step size S = 4 and search parameter p = 7*
6. *Search 8 locations +/-S pixels around location (0,0)*
7. *Pick among the 8 locations searched, the one with minimum cost function* LOC_1
8. *Set step size S = 8 and search parameter p = 7*
9. *Search 8 locations +/- S pixels around locations (0,0)*
10. *Pick among the 8 locations searched, the one with minimum cost function* LOC_2
11. *Pic LOC =* $min(LOC_1, LOC_2)$
12. *if* $(LOC_1 < LOC_2)$ *Then S=4 else S=8*
13. *Set the new search origin to the above picked location*
14. *Set the new step size as S = S/2*
15. *Repeat the search procedure for 2 times.*

End

4. Simulation Results

Simulation of the proposed architecture has been performed to illustrate the performance and the effectiveness of the proposed system. We emphasized on two components namely the low bit rate algorithm, and the *CO-PU* algorithm where it receives different object moving from one region to the other. We used multiple scene of the same videos where movement of an object is detected by the at least two IPNs. The performance of the low bit rate algorithm performed by the IPN was measured by PSNR and the compression ration versus the DCT component of BBME, and the time to decode the sent DCT frames through IDCT. The new three step search for fast motion is simulated and compared to the original three step search and the full search algorithms, with respect to accuracy, and PSNR. The full search is extended to 20x20 search window to accommodate for the out of boundary search window devised in our new algorithm. While the performance of assembling objects wandering at different regions, which is done by CO-PU, was measured by the accuracy of detection, and time required to confirm the movement and the threat there after.

4.1 Verifying the time savings of the proposed algorithms

The proposed algorithms are first tested individually using the conventional block based motion estimation: BBME algorithm as the ground truth. Also comparison of our search algorithm component in the LPHSME is measured against three step search algorithm, as well as full search algorithm. We consider set of video sequences with fast motions such as Table Tennis and Football, with moderate or slow motions such as Claire, MissAmerica, Foreman, Mother&Daughter, Salesman and Carphone.

1. The *ALBR* is compared with the DCT component of BBME. The comparison of the total number of bits transferred per frame. Compression rate versus PSNR for different videos are depicted in table 1 and 2 at different level of compression for ALBR.
2. *LPHSME* is compared with the full search component and the three step search of BBME. The comparison the time required for both to send the motion vectors of all frames of a scene or scenes, from sender to receiver in terms of number of bits transferred per second and the total number of bits send and the time in seconds.

4.2 Discussion

The simulation results are indicated in tables I, II, and III. Also Fig. 8 and 9 depict performance of the proposed algorithm against FS and 3SS. Table I illustrates the performance of the proposed ALBR algorithm against the DCT component of BBME. Results in table I indicate that there is a reduction in Average number of bits per Iframe by approximately 60% in case of low motion video sequences (e.g., MissAmerica) and 53% in case of fast motion video sequences (e.g., Table Tennis) This great savings in DCT time is due to reducing the number of DCT blocks transferred. The performance of the proposed ALBR algorithm against the DCT component of BBME is illustrated in Table II. Different levels of compression, in terms of PSNR and CR, are depicted. From x=2 to x=64, where x, is the denominator of (Total number of DCT blocks/x), which are the actual number of DCT blocks to be transferred, the others will be indexed. In table III illustrates the performance of the proposed LPHSME algorithm versus the ME component in BBME (Fs and 3SS) It is clear that there is a reduction in motion estimation time by approximately 89% in case of low motion video sequences (e.g., Claire) and 84% in case of fast motion video sequences (e.g., Football) This great savings in motion estimation time is due to reducing the number of search points in the search area which consequently reduces the computations required for each MV for each current block. Also PSNR of the proposed algorithm resembles FS and is much better than 3SS.
In fig. 8 and 9, the superiority of the proposed algorithm LPHSME is shown. Motion estimation time of LPHSME against FS is less by 87% on average. Also LPHSME needs only 18% more time than 3SS but with PSNR 8% less than FS as opposed to 3SS which is 20% less than FS.

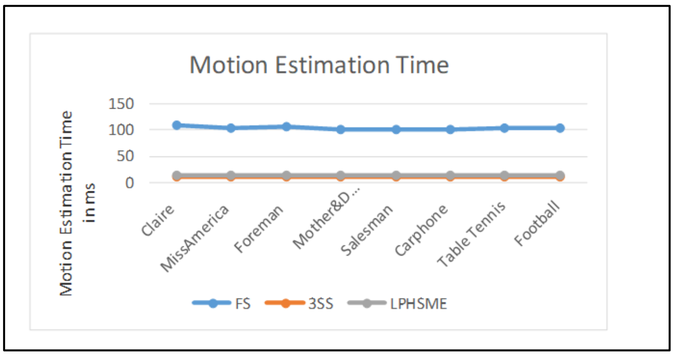

Figure 8. The motion estimation time of the proposed LPHSME algorithm versus the ME component in BBME (Fs and 3SS) for different video sequences

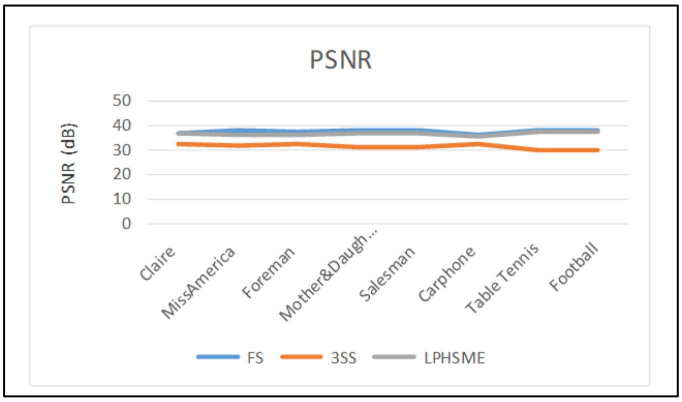

Figure 9. PSNR of the proposed LPHSME algorithm versus ME component in BBME (Fs and 3SS) for different video sequences

5. Conclusions

This paper proposes an in-depth study of the research problems and to devise innovative mechanisms, algorithms, software to meet the challenges. In high-security systems, bitrate reduction is desirable because of the limited bandwidth. , one of the main feature of this paper is introducing a Low-Power Low-Bit rate video compression algorithm to accommodate the low power requirements at sensor nodes, and the low bit rate requirement for the communication protocol. A proposed Data-Integration System that is in charge of integration of data collected from each sensor.
The objectives of the proposed system are the following:

- **Integrate additional sensor automatically**. The wireless sensor network is in constant change some of the sensors inside the network may die before others. Also new sensor may be added. After the network has been deployed. The DIS must adapt to this situation.
- **Integrate multi-data based on value of information**. The information provided for some of the sensors may be more important because they are closer to the source of the event. Therefore DIS must provide a mechanism to give a greater weight to the most important data.
- **Data must be integrated only if they provide new information**. If data from some sensors has not been change in the last scan, then no new information is

provided. Therefore, these signals must not be included in the integration process.

- **Data from different rates must be combined**. DIS must deal with signals that are not available at the same time and have different sampling rates.
- **Treatment of uncertainties**. The system must be prepared to deal with signal that may not be trusty. For example, in a battlefield some countermeasure may be employed, by introduce or change some signals. DIS must be capable to decide whether to combine a signal that may be corrupted.
- **Real-time operation**. Data integration must be done on real-time speed.

Table 1. Performance of the proposed ALBR algorithm against the DCT component of BBME.

Video Sequence (30 Frames) DCT with quantization	DCT Component of *BBME* 180 bits/block		DCT Component *ALBR* 180 bits/block							
	x=0 , no index		x=8 , 1 bits/index				x=64 , 1 bits/index			
	Num DCT blocks	Average PSNR (dB)	Num DCT blocks	Num indices	Aver PSNR (dB)	CR ~86%	Num DCT blocks	Num indices	Average PSNR (dB)	CR ~95%
Table Tennis (4CIF)	6336	39.9	752	5584	35.5	88.6%	99	6237	18.0	97.89%
Football (4CIF)	6336	40.5	752	5584	35.1	88.6%	99	6237	17.9	97.89%
Claire (CIF)	1584	41.5	198	1386	37.55	87.0%	25	1559	18.2	97.87%
MissAmerica (CIF)	1584	40.2	198	1386	37.12	87.0%	25	1559	17.9	97.87%
Foreman (CIF)	1584	40.9	198	1386	36.9	87.0%	25	1559	18.2	97.87%
Mother&Daughter (QCIF)	396	40.4	50	346	36.4	86.8%	7	389	18.4	97.68%
Salesman (QCIF)	396	40.4	50	346	35.7	86.8%	7	389	17.4	97.68%
Carphone (QCIF)	396	40.2	50	346	36.2	86.8%	7	389	17.2	97.68%

Table 2. Performance of the proposed ALBR algorithm against the DCT component of BBME

Num of DCT blocks in BBME	Num of DCT blocks in ALBR	Num of indices in ALBR	Compression Ratio CR	Aver PSNR (dB)
x=2				
4CIF	3168	3168	50%	39.5
CIF	792	792	50%	39.43
QCIF	198	198	50%	39.7
x=4				
4CIF	1584	4752	75%	38.1
CIF	396	1188	75%	38.12
QCIF	99	297	75%	38.2
x=8				
4CIF	792	5544	87%	35.3
CIF	198	1386	87%	35.4
QCIF	50	346	87%	35.6
x=16				
4CIF	396	5940	93%	30.5
CIF	99	1485	93%	30.12
QCIF	24.75	371.25	93%	31.1
x=32				
4CIF	198	6138	96%	24.5
CIF	49.5	1534.5	96%	24.12
QCIF	12.375	383.625	96%	35.2
x=64				
4CIF	99	6237	98%	18.1
CIF	24.75	1559.25	98%	17.9
QCIF	6.1875	389.8125	98%	18.25

Table 3. Performance of the proposed LPHSME algorithm versus the ME component in BBME (Fs and 3SS)

Video Sequence	*Conventional Full Search (FS)*		*Three Step Search: 33S*			*LPHSME*		
	PSNR (dB)	ME-Time (ms)	PSNR (dB)	Number of Matches Vs. FS	ME-Time (ms)	PSNR (dB)	Number of Matches Vs. FS	ME-Time (ms)
Claire	37.1	109.3	32.6	82.02%	11.23	35.1	89%	13.8
MissAmerica	37.9	102.3	31.9	83.5%	11.1	36.2	91%	14.1
Foreman	37.7	105.33	32.5	81.11%	11.7	36.1	92.4%	13.98
Mother&Daughter	38.1	101.2	31.3	82.44%	10.4	37.1	91.4%	13.2
Salesman	37.82	99.5	31.23	81,5%	9.8	36.77	91.4%	13.7
Carphone	36.4	101.5	32.7	80.5%	10.1	35.7	92%	12.8
Table Tennis	38.18	103.7	30.1	77.5%	11.1	37.3	94.1%	13.4
Football	38.01	104.1	30.2	78.5%	11.4	37.21	93.2%	13.28

6. Acknowledgement

This research was funded by the Deanship of Scientific Research at Princess Nourah bint Abdulrahman University through the Fast-track Research Funding Program

References

[1] Amiruddin Amiruddin, Anak Agung Putri Ratna, Riri Fitri Sari, "Systematic Review of Internet of Things Security," International Journal of Communication Networks and Information Security (IJCNIS), Vol 11, No 2 (2019), pp. 248-255.

[2] Miguel, M.; Jamhour, E.; Pellenz, M.; Penna, M. "A Power Planning Algorithm Based on RPL for AMI Wireless Sensor Networks," Sensors 2017, 17, 679

[3] Bederson, B.B., Clamage, A., Czerwinski, M.P., Robertson, G.G.: Datelens "A fisheye calendar interface for PDAs," ACM Trans. Comput.-Hum. Interact. 11(1), pp. 90–119 (2004).

[4] Hai-Peng Ren, Kun Tian & Ren Zhou (2018)" Local time-varying topology identification of network with unknown parameters based on adaptive synchronization," Automatika, 59:3-4, 391-399.

[5] Panda B.S., Shetty D.P., Pandey A. (2015) "k-Distinct Strong Minimum Energy Topology Problem in Wireless Sensor Networks," In Distributed Computing and Internet Technology. ICDCIT 2015, pp. 250-254.

[6] Salim EL KHEDIRI, Rehan Ullah Khan, Waleed Albattah, "An optimal clustering algorithm based distance-aware routing protocol for wireless sensor networks," International Journal of Communication Networks and Information Security (IJCNIS), Vol 11, No 3 (2019), pp. 391-396.

[7] Michael M. Zavlanos, Alireza Tahbaz-Salehi, Ali Jadbabaie, George J. Pappas, "Distributed topology control of dynamic networks," in Proc. of American Control Conference 2008, pp. 2660-2665, 2008.

[8] P. Mudali, T.C. Nyandeni, N. Ntlatlapa, M.O. Adigun, "Design and implementation of a topology control scheme for Wireless Mesh Networks," AFRICON 2009, pp. 1-6, 2009.

[9] Nedal Ababneh, Anastasios Viglas, S. Selvakennedy, Nadia Boukhatem, "A Topology Control Algorithm with Good Spanner Properties for Wireless Sensor Networks," Communication Networks and Services Research Conference (CNSR) 2010 Eighth Annual, pp. 179-186, 2010.

[10] Xiaoli Xiao, Zhiqian Zhou, "Study on WSN's Fault-Tolerant Topology Control Based on GA," International Conference on Computer Sciences and Applications (CSA) 2013, pp. 146-149, 2013.

[11] Pallavi Yarde, Sumit Srivastava, and Kumkum Garg, "A cross-layer routing protocol for wireless sensor networks," Advances in Intelligent Systems and Computing, vol. 847, pp. 83–91, 2019.

[12] Kalpana Murugan, Al-Sakib Khan Pathan, "A Routing Algorithm for Extending Mobile Sensor Network's Lifetime using Connectivity and Target Coverage," International Journal of Communication Networks and Information Security (IJCNIS), Vol 11, No 2 (2019), pp. 290- 296.

[13] Wonki Lee, DaeEun Kim, "Autonomous Shepherding Behaviors of Multiple Target Steering Robots Sensors," (Basel) 2017 Dec; 17(12).

[14] Ali M. A. Abuagoub, "IoT Security Evolution: Challenges and Countermeasures Review," International Journal of Communication Networks and Information Security (IJCNIS), Vol 11, No 3 (2019), pp. 342-350.

[15] Hicham Lakhlef, Youcef Imine, Abdelmadjid Bouabdallah, "A Distributed Collision-free Distance-2 Coloring Algorithm for Ring Networks," Software Telecommunications and Computer Networks (SoftCOM) 2019, pp. 239-243.

[16] Lee, D.; Choi, S. "Multisensor fusion-based object detection and tracking using active shape model," In Proceedings of the IEEE International Conference on Digital Information Management(ICDIM), Melbourne, Australia, 26–28 September 2011; pp. 108–114.

[17] M. Thiruveni, D. Shanthi, "Efficient VLSI Architecture for 16-Point Discrete Cosine Transform," Proceedings of the National Academy of Sciences, India Section A: Physical Sciences, 2018, pp. 254-258.

[18] Renato J. Cintra, Fábio M. Bayer, "Comments on 'Area and power efficient DCT architecture for image compression," EURASIP Journal on Advances in Signal Processing, vol. 2, 2017, pp. 342-346.

[19] Wen-Hsiung Chen, C. Harrison Smith, And S. C. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform," IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. 9, 1977, pp.125-133.

[20] Wei Liu, Wei Chen, "Recent Advancements in Empirical Wavelet Transform and Its Applications," Access IEEE, vol. 7, 2019, pp. 103770-103780.

[21] K J A S Sundar, V. Vaithiyanathan, "Design and Analysis of Fusion Algorithm for MultiFrame Super-Resolution Image Reconstruction using Framelet," Defence Science Journal, vol. 65, no. 4, pp. 292-299, 2015.

[22] K. J. A. Sundar, M. Jahnavi and K. Lakshmisaritha, "Multi-sensor image fusion based on empirical wavelet transform," 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), 2017, pp. 93-97.

[23] x. Lu and C. Xu, "Novel Gaussian mixture model background subtraction method for detecting moving objects," 2018 IEEE International Conference of Safety Produce Informatization (IICSPI), Chongqing, China, 2018, pp. 6-10.

[24] N Srivastav, S L Agrwal, S K Gupta, S R Srivastava, B Chacko, H. Shanna, "Hybrid object detection using improved three frame differencing and background subtraction," 7th International Conference on Cloud Computing Data Science & Engineering-Confluence, 2017, pp. 613-617.

[25] Panda, Deepak Kumar and Sukadev Meher. "A new Wronskian change detection model based codebook background subtraction for visual surveillance applications," J. Visual Communication and Image Representation 56 (2018): 52-72.

[26] V. Daiya, T. S. S. krishnan, J. Ebenezer, K. Madhusoodanan, S. A. V. SatyaMurty and B. Rao, "Dynamic architecture for Wireless Sensor Network-implementation & analysis," 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, 2016, pp. 1206-1211.

[27] O. M. Dumin, V. A. Katrich, Y. A. Koltunov and I. V. Neroda, "Numerical methods of structure-statistical object classification for remote sensing," 2013 IX Internatioal Conference on Antenna Theory and Techniques, Odessa, 2013, pp. 426-428.

[28] M. U. Yaseen, M. S. Zafar, A. Anjum and R. Hill, "High Performance Video Processing in Cloud Data Centres," 2016 IEEE Symposium on Service-Oriented System Engineering (SOSE), Oxford, 2016, pp. 152-161.

[29] K. Takaya and R. T. Reinhardt, "Low bit-rate facial motion picture coding using image warping," IEEE WESCANEX 97 Communications, Power and Computing. Conference Proceedings, Winnipeg, Manitoba, Canada, 1997, pp. 138-143.

[30] P. Smiti, S. Srivastava and N. Rakesh, "Video and Audio Streaming Issues in Multimedia Application," 2018 8th International Conference on Cloud Computing, Data Science & Engineering, Noida, 2018, pp. 360-365.

[31] Tarif Riyad Rahman, Miftahur Rahman, "Compression Algorithms for Audio-Video Streaming," 2010 International Conference on Intelligent Systems Modelling and Simulation, 2010, pp. 240-244.