Performance Optimisation of Wireless Visual Sensors for Wild Life Monitoring

Shivali Dhaka[1] , Balabuksh[2]

[1] Research Scholar, Career Point University, Kota Rajasthan;
[2] Research Guide, Career Point University, Kota Rajasthan
Correspondence email – shivi359@gmail.com

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 10 May 2021

Abstract

Wireless Visual Sensor Network (VSN) has revolutionized the emerging world of Internet of Things, providing visual data as images and videos for emergency detection, localization, tracking, pattern recognition etc. For wild life surveillance, very limited research is explored using VSNs due to visual occlusion and dynamics of coverage failures. In this context, algorithms and optimization approaches have been investigated to perform different types of quality assessment and performance enhancement. Proposed work presents a faster method of optimum selection of Visual Sensors for maximum coverage of the predefined surveillance space. The Wild life habitat is modeled as surveillance space where occlusion (obstacle) is impairing the performance of VS. The other sets of VSs in the VSN provides feasible locations for wider coverage using an optimized search algorithm. The problem of optimum VS selection for maximum coverage considering both static and randomly moving obstacles is mapped as a Grey Wolf Optimization (GWO) problem. The proposed algorithm is computationally lighter and converges very fast as compared to Contemporary Genetic Algorithms (GA).

Keywords Visual sensors, GWO, optimisation, iterations, coverage, occlusion, wild life

1 Introduction

A Wireless Visual Sensor Network (VSN) or camera network is essential for surveillance of indoor or outdoor spaces, as a single camera cannot cover discrete events in a large surveillance space. The major challenge for such a VSN is to cover hotspot in a surveillance space, like multiple entrances (wild animals) and locations of various important activities, with optimum number of cameras and minimum predefined resolution, considering both static and dynamic occlusion. To cover such multiple distributed events with minimum required resolution, an algorithm is needed that can determine optimum locations of the Visual sensors cameras along with their pan, tilt and zoom levels [1, 2].

Identifying optimum configuration of VSN for such coverage is a combinatorial optimization problem. It will be difficult for simple search techniques to determine optimum placement configurations [3]. Genetic Algorithm (GA) [2, 4] and Particle swarm optimization (PSO) [5] have been used in the past for camera placement problem. Apart from GA and PSO, meta heuristics approach Grey wolf optimization (GWO) [6] has been used in a variety of fields viz. channel estimation in wireless communication [7], photovoltaic systems [8], wireless body area networks [9] and image processing [10]. GWO is inspired by the behavior of grey wolves to attack the prey for hunting and is preferred by many researchers for optimization purpose because of its fast rate of convergence and robust behavior [6, 9]. As compared to other optimization algorithms, GWO requires fewer operators and parameters to adjust. GWO has performed better than algorithms like ACO (Ant Colony Optimization), GA and PSO for general optimization problems. Hence, GWO is used as an optimization algorithm for developing an optimum VS placement strategy. Author’s contributions to the paper are as follows.

Fig.1 Group of Grey Wolf for optimal search (best VS selection)

In this paper, a VS placement algorithm is proposed for covering each wild –life-habitat, by different sets of VSNs so as to enhance the information content of images and to reduce the effect of occlusion due to randomly moving obstacles.

The paper proposes a meta-heuristics approach that optimizes the coverage of discrete habitat in a surveillance space by exploiting the merits of different sets of VSNs in vicinity.

The analytical model proposes a preprocessing mechanism that takes a number of VS and mobile obstacles as input, computing the resulting set of active VSs. Such computed set can be processed as an input to any existing optimization or quality assessment approach. The methodology presented includes different type of coverage failure, the impaired image/data set collected by VS is considered as visual occlusion. To the best of our knowledge, the coverage modelling problem and fault- less visual sensors in wild life monitoring is not addressed before.

The paper has 5 sections. Section 2 explores the literature survey. Section 3 provides analytical framework for state of the art problem. Section 4 constitutes the proposed optimisation architecture followed by mathematical model. Section 5 investigates the performance evaluation of the proposed algorithm-Prop-1. Paper ends with conclusion and future scope of work following List of references.

2 Literature Survey

The primary objective of visual sensors (VSs) is to view part of the monitored field and sometimes the quality of the network will be a function of this characteristic. But such ‘‘viewing’’ can be performed in different ways and with different objectives. In fact, coverage assessment as an indication of ‘‘quality’’ has been investigated in recent past by many researchers, achieving different promising results. The work in [11] proposed a distributed mechanism to optimize coverage, availability rate and achieve the load balancing to improve performance of the network. In [12] an approach is suggested to reduce the latency for faster communication viz. handover in emerging wireless networks. In [13] a game theoretic approach is investigated for sensor networks. The work in [14] proposed a new methodology to assess quality when performing barrier coverage, also defining an optimization mechanism to increase coverage in such scenario. In [15], a metric referred as Quality of Viewing (QoV) was proposed to extend the traditional perception of Quality of Service (QoS) when assessing the performance of applications based on visual sensors. The QoV is expected to be employed to measure the quality of retrieved data for the monitoring requirements of the applications and thus it depends on the particularities of each considered scenario. Coverage assessment and improvement were also performed in [16], which optimized the orientation of cameras for indoor monitoring. Metrics for coverage of targets and areas have also been proposed in the last years, being discussed and compared in some works [16], [17]. The selection of faultless visual sensor nodes is centered on the identification and processing of a set of mobile obstacles and their impact on the coverage of the considered sensors as shown in figure. [18]

In wild life monitoring, obstacle may be any moving object (e.g. wild animals, humans, vehicles, birds, dense vegetation etc), an imaginary rectangle is considered ''circumscribing'' the obstacle. This modelled rectangle, with width w(o) and height h(o), represents a virtual and mathematically defined instance of the real obstacle. Fig. 1 presents the generic idea of modelling obstacles as rectangles, which are considered as being perceived from a top-view perspective.

Every VS node s, for an initial set of S visual sensors, will be positioned at the (Ax(s), Ay(s)) coordinates, assuming that two different sensors can't be deployed at the exact same location. All visual sensors are defined as being static (the initial position is not altered during the considered operation time of the WVSN) and the employed cameras don't have PTZ (Pan-Tilt-Zoom) features.

SELECTING VISUAL SENSOR NODES

The proposed mathematical model is intended to allow the dynamic identification or even estimation (through simulation) of coverage failures, which are an important source of quality impairments in wireless visual sensor networks. Exploiting this model, it is possible to select only the visual sensors that should be considered when performing any kind of optimization or quality assessment in wild life environment, being a preprocessing step for many applications.
The selection of visual sensors will be performed based on the defined mathematical model. The visual sensors selection mechanism will take as input the original set of visual sensors *S*, the current configuration of mobile obstacles *M* (that will change along the time) and the defined condition for coverage failure.

Proper Tracking and localization: Many problems related to localization and tracking of targets have been proposed in the last years, exploiting the power of visual data processing. The use of mobile and rotatable cameras has also enhanced the applicability of WVSN. A proper selection of visual sensors is needed that should eliminate those VS which experiences some kind of coverage failure. The localization of group of visual sensors may not be accurate due to visual coverage failures, as discussed in this paper. In fact, there may be different causes for coverage failures and additional research is still required to find and to model each of such causes. In this paper, visual occlusion is caused by mobile obstacles and may constitute following causes for coverage failures:

Effect of Low ambient light: When regular cameras are employed, the ambient light may determine if a certain visual sensor is under a coverage failure, since the retrieved visual data may become useless under low ambient light. For wildlife monitoring, visual sensors may also become faulty during the night;

Effect of Heavy rain or fog: Visual data processing algorithms may be used to identify if a camera's lens is dirty or with too many water drops. Alternatively, additional sensors may identify adverse weather conditions that can be mathematically computed when processing coverage fails;

Effect of Redundant orientation: When visual sensors are viewing areas that are not intended by a particular application, they may be assumed as faulty nodes, even though their FoV are not occluded.

Coverage failures are processed as a total or partial situation, depending on the application monitoring requirement(s). This perception may result in visual sensors being processed with some level of priority, weighting the operation of the network.

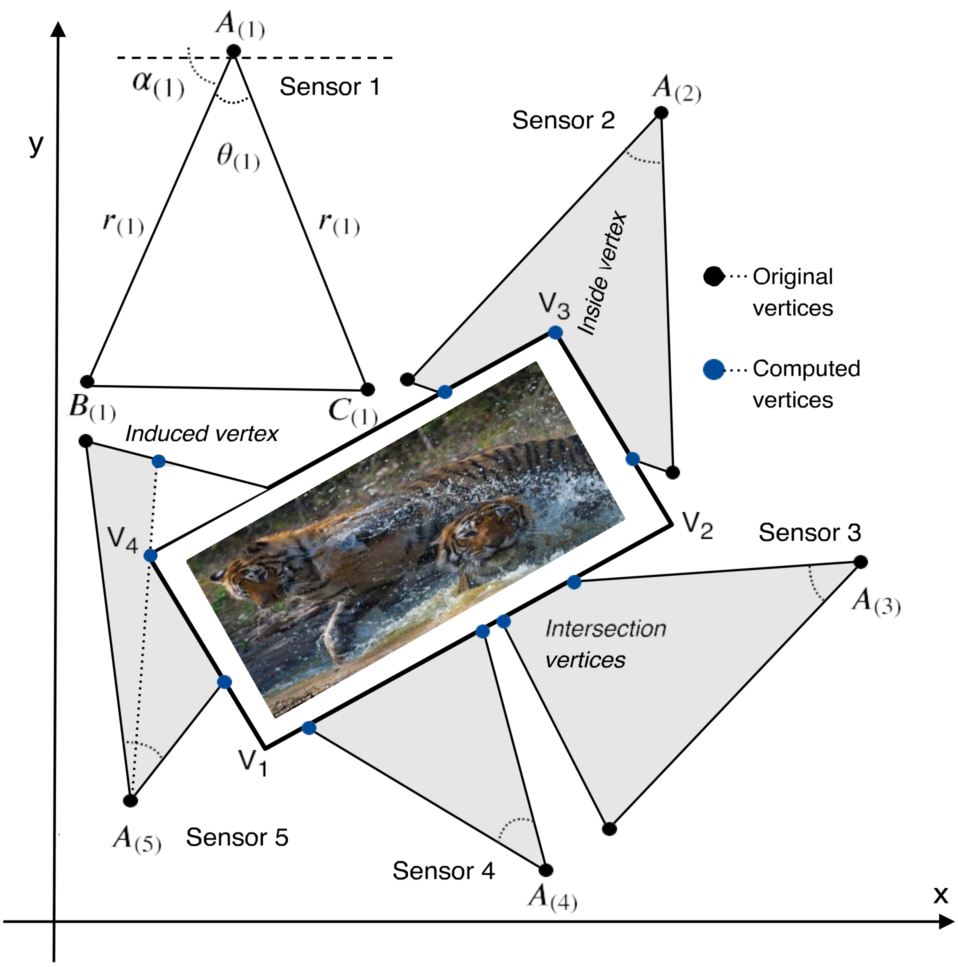

Fig.1 Occluded Image covered by 5 Visual Sensors

Finally, it is expected that the adoption of faultless visual sensors selection as a preprocessing mechanism can bring significant results for optimization and quality assessment in (wireless) visual sensor networks.

3 Analytical frame work

In wild life environment, monitoring large surveillance space having distributed multiple activities with a predefined resolution is a challenging task. The aim of the proposed work is to determine the optimum location of each VS to achieve maximum coverage of all predefined View Point (VP) of a large surveillance space satisfying the task-based constraints which may be static or dynamically varying according to the requirements. The major constraint is covering all distributed habitats by a set of VSs i.e. VSN for capturing maximum information content in the images. This problem is mapped as an optimization problem. The proposed algorithm is computationally simple as the methodology is based neither on calibration of cameras nor on learning environment.

3.1 Methodology of Grey Wolf Optimisation

A novel method for optimizing the coverage of visual sensors is proposed by adjusting their locations, using a meta-heuristic algorithm GWO. The proposed GWO based approach is computationally lighter and faster in execution. Dwelling area and natural habitat of animal kingdom is mapped to surveillance space, obstacles and feasibility areas, considering a wildlife monitoring application. Optimally selected VSN ensures good quality and resolution of the images. The proposed algorithm covers each VP by a set of VSN (say 3) cameras which not only enhances the information content in the images but also reduces the possibility of occlusion due to static and randomly moving obstacles.

In nature, animals search for food in a random or quasi-random manner. In general, the foraging path of an animal is effectively a random walk because the next move is based on the current location/state and the transition probability to the next location. Which direction it chooses depends implicitly on a probability which can be modeled mathematically. For example, various studies have shown that the flight behavior of many animals and insects has demonstrated the typical characteristics of Lévy flights.

3.2 Modeling of Surveillance space

Surveillance space is modeled in terms of cubical grids. A point in the surveillance space is considered to be covered if it lies in the VP of the cameras with minimum required resolution. In this paper, user defines a set of high activity View Points (VPs), location of the obstacles with its shape and size and feasible locations for camera placement. A GUI shown in Fig. 2 defines VPs, obstacles and feasible locations for camera placement with cubical dimensions where is the largest side of the VP/ obstacle.

3.3 Mathematical framework for - GWO Optimiser

Each wolf is assigned a weightage according to the fitness function (coverage matrix). The values of the vector will be the optimum solution which gives optimum position of a VS to achieve maximum coverage of all VPs.

In this way, a population of wolves in GWO represents a set of VSNs belonging to the solution space. Our problem is now redefined to search for the fittest individual from solution space. The fitness function for each wolf is obviously the coverage matrix calculated. The optimization criterion is set for minimization. A GWO population comprising of N wolves (each wolf representing a VSN).

In each group there is a cluster head - CH who is liable for dealing with the entire bunch and its part. CH additionally looks towards the new hubs and out-going hubs from the groups. CH likewise deals with the event of node not in excess of a bunch like cluster. Presently go to the fundamental exploration question that how Grey Wolf functions for making the upgraded number of bunches.

At first the information is haphazardly produced by the boundaries (Number of hubs, Transmission range and network size) of the VSN. Later, the system is worked by the organization of nodes in the grid. The grouping (clustering) is made based on their resemblance or same highlights of nodes VSN performance could be function of hub's speed, heading, area, position and channel condition. For making the productive grouping, it is vital that a hub ought to be in one bunch/subnet in particular.

The pyramid of grey wolf initiate from alpha (α). In the mapping, alpha is a known as leader in the VSN pack. They give the instruction to others. Other wolves keep it tail down to follow the alpha for obeying the instructions. The main decisions of whole pack are usually taken by the alpha wolves. These decisions contains sleeping, wakeup time and many mores. Alpha wolves got the natural skills for organizing the pack. Alpha also keeps the pack well disciplined. After the alpha wolf, there is position of beta (β) grey wolves. These can also be male and female, betas are considered as second in the hierarchy of grey wolves. These wolves support the alphas for making the decision and betas help the alphas for implementation of their instruction to the lower level of grey wolves in the packs. Betas wolves are used by the alphas for the feedback purpose as well.

If any of the alpha wolves dies then one of the betas wolf is promoted to alpha wolf. Third order of grey wolves is Delta (δ). These wolves are categorized into spies, guards, predators and caretakers. Delta wolves help to protect the complete pack, also they keep eyes on the boundaries so that in case of danger counter measures can be taken for the pack. Hunters (Delta) provide the food for the others, caretakers look after the aged, weak and sick wolves in the pack. If case of death of beta wolf the senior delta wolf is promoted to beta wolf.

GWO Optimiser contains mainly four steps i.e. from exploration (searching) to exploitation (attacking). Omega exist in the last position of grey wolves. Due to the last in the position of wolves they always have to pay more than others in return of very small reward. Omega wolves also seem as babysitters, with no importance individually in the pack but lost of omega wolves also creates the problem. They are allowed to eat lastly after hunting. Death of delta promotes the any one of the omega to delta. There are some important phases of the grey wolf for the hunting as explained below.

3.3.1 Social hierarchy for Prey-Hunting and decision making

The best wolves for decision making and prey- hunting, alpha (α) wolves are considered and are the fittest solution in the ordering of grey wolf optimization. Beta (β) is considered as the second most and consequently delta (δ) and omega (ω). α, β, δ are used for the guidance in hunting process. Omega (ω) wolves just follow all three of upper hierarchy. Mathematically, the know-how of the grey wolf optimisation algorithm and the associated important stages are described in the following section.

3.3.2 How prey is encircled

Grey wolf encircle the prey during the process of hunting as;

$$\vec{D} = |\vec{C}.\overrightarrow{X_p}(t) - X(t)| \quad (1)$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A}.\vec{D} \quad (2)$$

where *A* and *C* are co-efficient vectors, *X* p is the position vector of prey, *X* is the position vector of grey wolves. The $\vec{A}$ and $\vec{C}$ is; $\vec{D}$ shows the 2-Dimensional position of the possible neighbors.

$$\vec{A} = 2\vec{a}.\overrightarrow{r_1} - \vec{a} \quad (3)$$

$$\vec{C} = 2.\overrightarrow{r_2} \quad (4)$$

In Eqs. (3) and (4) , $\overrightarrow{r_1}$and $\overrightarrow{r_2}$ are random vector range from 0 to 1. Whereas $\vec{a}$ is the factor which linearly decrease from 2 to 0. Eqs. (1) and (2) is used to update the position of wolves from current location to new-location. The two dimensional position of grey wolf, with respect to the prey. If wolf is at position (X, Y) and prey at (X^*, Y^*). The grey wolf will update its position according to the movement of prey which is mathematically modelled as in Eqs. (3) and (4) . The positions are adjusted with the help of vectors $\vec{A}$ and $\vec{C}$. If the wolf is at any position (*X, Y, Z*) and prey at (X^*, Y^*, Z^*) any of the position in 3-D so wolf will update their new position of random vectors , $\overrightarrow{r_1}$and $\overrightarrow{r_2}$.

3.3.3 How hunting is decided by leaders (alpha wolves)

These wolves try to find the location of optimum (prey) and encircle it for the hunting. Alpha are the senior most or most strengthen wolves in the whole pack designates for the hunting. Sometime betas and deltas also perform this (hunting) task. In mathematical stimulation we store the best three solutions and convey it to remaining wolves (Omega) for updating their position accordingly. These tasks are performed with the help of following equations.

$$\overrightarrow{D_\alpha} = |\overrightarrow{C_1}.\overrightarrow{X_\alpha} - \vec{X}| \quad (5)$$

$$\overrightarrow{D_\beta} = |\overrightarrow{C_1}.\overrightarrow{X_\beta} - \vec{X}| \quad (6)$$

$$\overrightarrow{D_\delta} = |\overrightarrow{C_1}.\overrightarrow{X_\delta} - \vec{X}| \quad (7)$$

$$\overrightarrow{D_\alpha} = |\overrightarrow{C_1}.\overrightarrow{X_\alpha} - \vec{X}| \quad (8)$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1}.(\overrightarrow{D_\alpha}) \quad (9)$$

$$\overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2}.(\overrightarrow{D_\beta}) \quad (10)$$

$$\overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3}.(\overrightarrow{D_\delta}) \quad (11)$$

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1}+\overrightarrow{X_2}+\overrightarrow{X_3}}{3} \quad (12)$$

3.3.4 How attacking is executed for prey (Exploitation)

Subsequent to circling and hassling the prey, grey wolves assaults the prey when it quits moving. We demonstrated it in mathematical conditions by taking the estimation of α˙ . As, change in $\vec{A}$ likewise diminish the estimation of $\vec{a}$. The estimation of a˙ is between 2 to 0. At the end of the day if the estimation of |A| < 1, it upholds the wolf pack to assault the ideal (prey). In addition, if estimation of |A| > 1, this authorizes grey wolves to investigate more territory rather than exploitation.

$$a = 2 - 1 * \left[\frac{2}{TH_{ITER}}\right] \quad (13)$$

The alpha wolf contains the base an incentive as it is viewed as the best arrangement, trailed by beta and delta separately. At long last the alpha gives us the optimized number of bunches. The $\vec{a}$ value is significant as it is the directly diminishing variable. At the point when the estimation of a ranges to zero it give us the upgrade arrangement. The estimation of $\vec{A}$ and $\vec{C}$ are additionally talked about in system area in subtleties. The given Fig. 3, is utilized to show the various stages or exercises during the execution. The hubs are instated in the system arbitrarily, we make the bunch grid by finding the neighbors and remembering that just a single hub ought to be chosen uniquely in one group. In addition, the two target variable w1 and w2 is utilized to assess the group network. The state of greatest emphasis, which is likewise the halting measures is utilized. In the following stage the fitness estimations of search operators are determined. The directly diminishing element is additionally used to take the execution toward the outcome. After single cycle the places of hubs are refreshed and process proceeds. Toward the end, when straightly diminishing element a pushes toward zero, the alpha wolf give us the upgraded arrangement. The quantity of clusters for this situation.

3.3.5 How search is done for new prey (Exploration)

The principle task in the entire optimisation process is to look/search for the prey or investigate by the grey wolf which is reliant on the situation of alpha, beta and delta. These wolves spread in the quest space for the investigation and afterward join to assault the chase.

Vector $\vec{A}$, values more than 1 or under -1 help the grey wolves to move away from the chase. Because of which it implements the inquiry operators to investigate universally. As referenced before that |A| > 1 methods scan for the better prey. Vector $\vec{C}$ additionally has go from [0, 2]. On the off chance that the estimation of C < 1 it de-emphasizes and if C > 1 it underscores prey in defining the separation. The vector $\vec{C}$ causes the streamlining agent to evade the neighborhood optima and uphold the procedure of investigation. It is significant to state here that $\vec{C}$ isn't directly diminished by $\vec{A}$. The estimation of $\vec{C}$ is allocated deliberately with the goal that it favors the looking of search space in all the emphases (from initial to final) to follow the fitter prey. Since there is a likelihood that might be fitter arrangement can be found in final cycle. $\vec{C}$ is otherwise called the impact of hindrances in the way for finding the prey.

Essentially these impediments in the way of moving toward powers to look altogether and prevent from quickly and helpfully finding prey. $\vec{C}$ really allots the some irregular load to the prey.

4 Proposed architecture

The proposed work uses meta heuristic algorithm Grey wolf Optimization (GWO) [8] to find optimum locations of the best visual sensors to cover each VP in a surveillance space by at least 3 VSN. A wolf ($W_{i,}$,1 < i < N, N = total number of VS) in GWO denotes a candidate solution and the entire population comprises of N number of wolves searching for the prey or the optimum solution. Each wolf (W_i) in the proposed algorithm represents a camera location (x, y) coordinates). The mapping of sensor nodes entities deployed in wid-life-habitat with the different parameters of grey-wolf optimisation process is described in the following section.

4.1 Decision model using GW Optimisation

The search process of GWO is initialized by a set of wolves (candidate solutions) searching for the prey (optimum solution). The position of each wolf is represented by a vector shown in Eq. 1 where N denotes the number of wolves in the population

Wolves encircle the prey in order to stop its movement. The encircling process can be mathematically formulated.

The fitness of a candidate solution/wolf in GWO is calculated using Eq. 13. The top 3 fittest wolves in the population are denoted by α, β and δ and the rest of the population is denoted by ω. In most cases, α decides the hunting, sleeping and walk decisions for a pack of wolves and entire pack follows the decision made by α. In some democratic cases, α is seen to follow the decisions made by β and δ. The next level in the hierarchy is β. They help α to make decisions. The lower level in the hierarchy is ω. They simply follow the rules. The hunting behavior of the wolves consists of 3 major steps namely chasing the prey, encircling the prey, and finally attacking the prey.

It is assumed that the position of the prey is best known to first three best solutions α, β and δ. Therefore, we save the position of the first three best solutions as α, β and δ for the current iteration *t* and oblige all other wolves in the population (ω) to update their positions. Coefficients A_1, A_2, A_3 are calculated. The wolf (W_i) finally updates its position towards the prey (optimum solution) as depicted in pseudo code-1, in section 4.2.

During optimization of GWO, ω wolves iteratively improve their fitness according to α, β, and δ. When the improvement in the fitness of ω wolves reaches a threshold and Wolves encircle the prey in order to stop its movement. The encircling process can be mathematically formulated.

GWO hunt the prey on basis of α, β, and δ, therefore it is prone to be stuck at local optima. Although some exploration was done during encircling the prey, here GWO needs more operators to carry out exploration and to search for better solutions. With the proposed algorithm - *Prop-1,* the merits of GWO are hybridized with new proposed module to converge fast and come close to the global best solution. In most of the simulation cases the GWO optimizer is stuck in local optima hence characteristics of Lévy flights are incorporated to get out of local optima that too in less iterations.

4.2 Modeling of GWO Optimiser (Pseudo code-1)
Initially, GWO optimizer is modeled as per the concept/literature discussed in section 3. A pseudo code is presented to get an insight.

Start

Initialize alpha, beta, and delta positions

Configure dimension, iterations and bounds

dim=5; ub=x; lb=1;Th_iter=100;(say)

Configure no. of search agents

Initialize the positions of search agents

Sort the positions

Update the new positions

While 1< Th_iter

Return back the search agents (Encircling)

Calculate objective function for each search agent

Update Alpha, Beta, and Delta (Exploitation)

Decrease the surveillance space <2

Include other solutions/ omega's (Exploration)

Compute fitness/cost function - sort

Update the new positions of alpha, beta and gamma

Get the mean from alpha, beta and gamma

Get the final position

End

Stop

4.3 Modeling of proposed algorithm – I (Prop-1)
GWO optimizer is used in the proposed model with the merits of levy's flight as shown in the following pseudo code – 'Prop-1' .

Start

Initialize alpha, beta, and delta positions

Configure dimension, iterations and bounds

dim=2; ub=x; lb=1;Th_iter=100; (say)

Configure no. of search agents

Initialize the positions of search agents

Sort the positions

Update the new positions

While 1< Th_iter

Return back the search agents (Encircle)

Calculate objective function for each search agent

Update Alpha, Beta, and Delta (Exploitation)

Decrease the surveillance space <2

Include other solutions/ omega's (Exploration)

Compute fitness/cost function - sort

Update the new positions of alpha, beta and gamma

Get the mean from alpha, beta and gamma

Get the final position

End

Obtain the solution set from GWO / above surveillance space

Calculate the levy exponents – alpha, beta, gamma and sigma

Set -1 for random walk

Levy exponent and coefficient

Use step=1, for standard random walks,;

Obtain the difference factor (s-best)

when the solution is the best solution, keep it unchanged.

Limit length scale

Counter over efficiency/aggressiveness

by Updating actual random walk/flight

Apply bounds/limits/constraints-lb/ub

end

Stop

5 Performance Evaluation

Modeling and simulation is done in MATLAB-2020 A - tool. Different objective function are tested viz. the benchmark objective function which are convex function are mostly provides a global best solution. For the performance comparison and analysis of proposed algorithm, F1, F3, F11, F15, F18 and F21 are used.

5.1 Simulation - configuration setting for proposed model
Following parameters are used to configure the modeling, on MATLAB-v-2020b

No. of Visual Sensors= 30

Dimension = 2/11

No. variables = 2/11

Iteration Threshold = 50/100/500

Upper and lower bounds = as per the objective function – F1 to F21

5.2 Performance evaluation of GWO optimizer

Simulations are carried on MATLAB (version-2020b) software for GWO optimizer as per the pseudo code shown in the section 4.2. With the above simulation configuration, performance of GWO optimizer is evaluated. For 50 iterations optimizer was ran. The cost of the objective function-F1, is almost reduced to zero after 23 iterations. The algorithm converges very fast in 23rd iteration and the subsequent optimizer coefficient ‘a’ is reducing after every iteration in Fig. 5.2. This implies the search helped in getting closer to the target/object/ or prey. In Fig. 5.3, new solution set is accommodated to search for new better solutions i.e. exploration. In fig. 5.4, GWO optimser suggest global best solutions/ VSN after 10th iteration. In fig. 5.5, Optimiser hits almost very close to the global best solution in between 30th to 45th iteration.

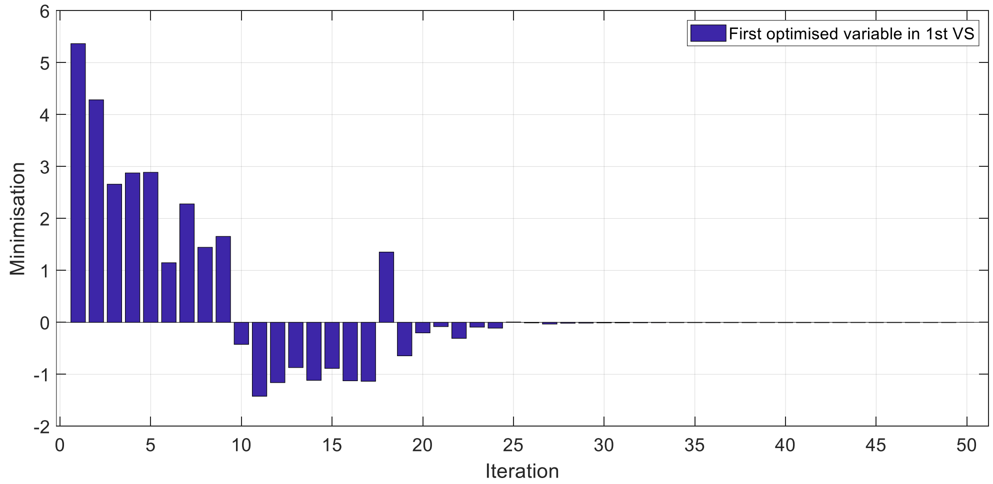

Fig.5.1 Iteration vs Minimisation of 1st variable in 1st VS

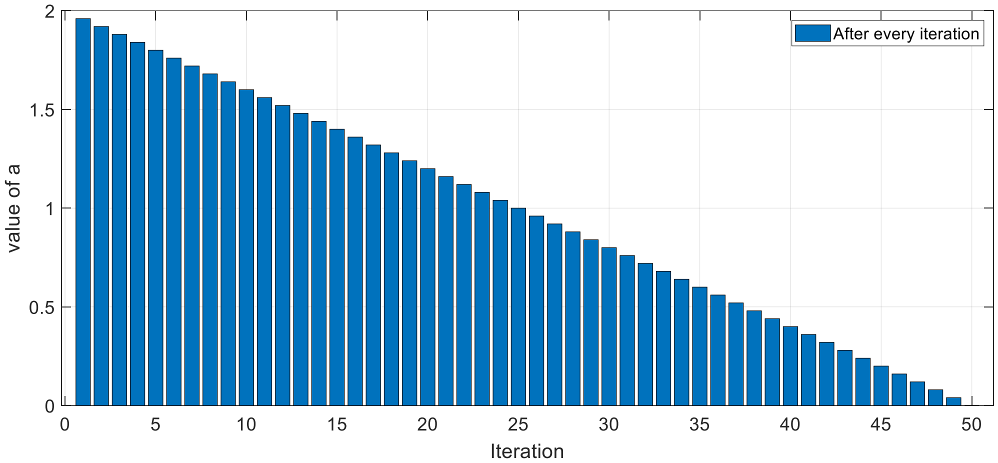

Fig.5.2 GWO - Optimiser converging to better solution after every iteration

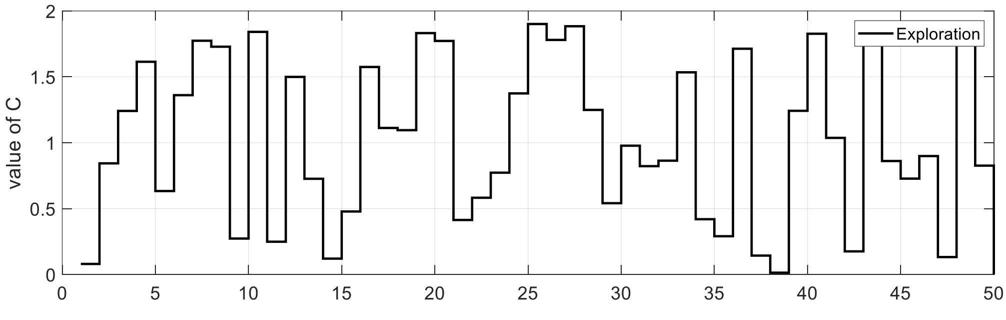

Fig.5.3 GWO Optimiser accommodating new solutions (better VSNs)

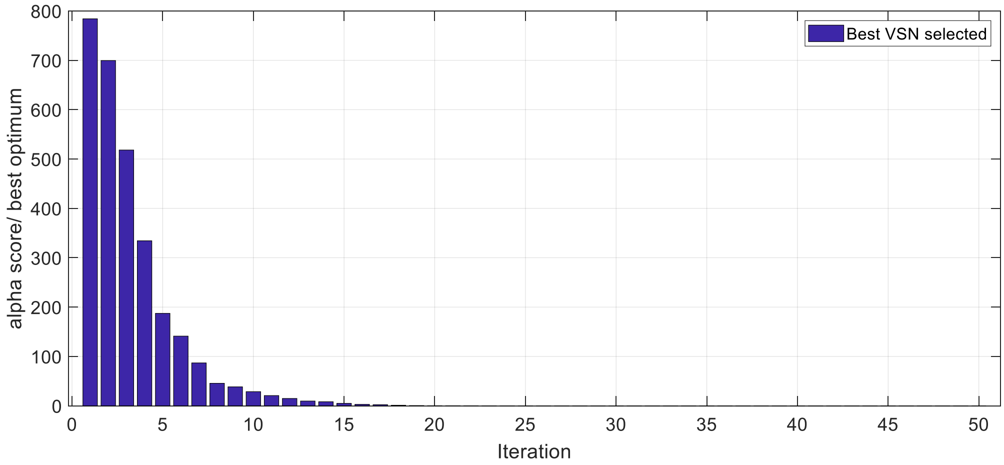

Fig.5.4 GWO optimiser's fitness value for alpha sets (best VSNs)

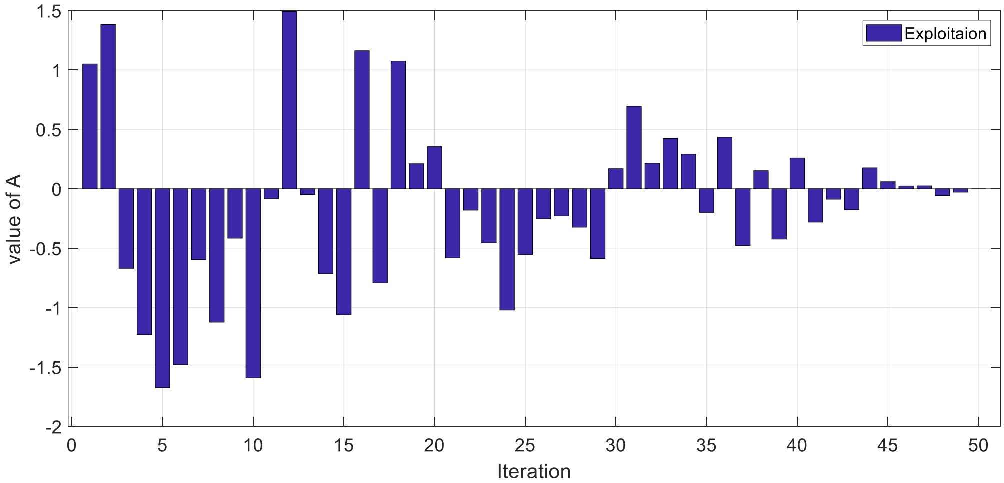

Fig.5.5 GWO convergence after every iteration (best VSNs)

5.3 Result and discussion

Modeling and Simulations is done for proposed optimisation algorithm as per the pseudo code shown in section 4.3. Based on the above simulation configuration, performance of proposed algorithm is compared with GWO optimizer. For 50 iterations optimizer was ran. In fig. 5.6 (b) When F3, objective function is used, the proposed algorithm outperforms the GWO algorithm. Although both algorithms converges with the increase in iterations but GWO is locked near the local optima where as Prop-1, further reduces the fitness value / score and suggest some more better solution sets. Slope and gap near 10^{th} iteration clearly shows that the best (global sub-optimal) solution is achieved by prop-1. In Fig. 5.6(b) and 5.7(b), it is observed that with the increase in number of iteration, the fitness score of the objective function is decreasing almost radically (from 20 to 50) for proposed scheme (Prop-1), performing better as compared to GWO optimser. Similarly the performance is evaluated for other benchmark functions viz. F3 and F21 as shown in Fig. 5.6 (a) and Fig. 5.7 (a) respectively.

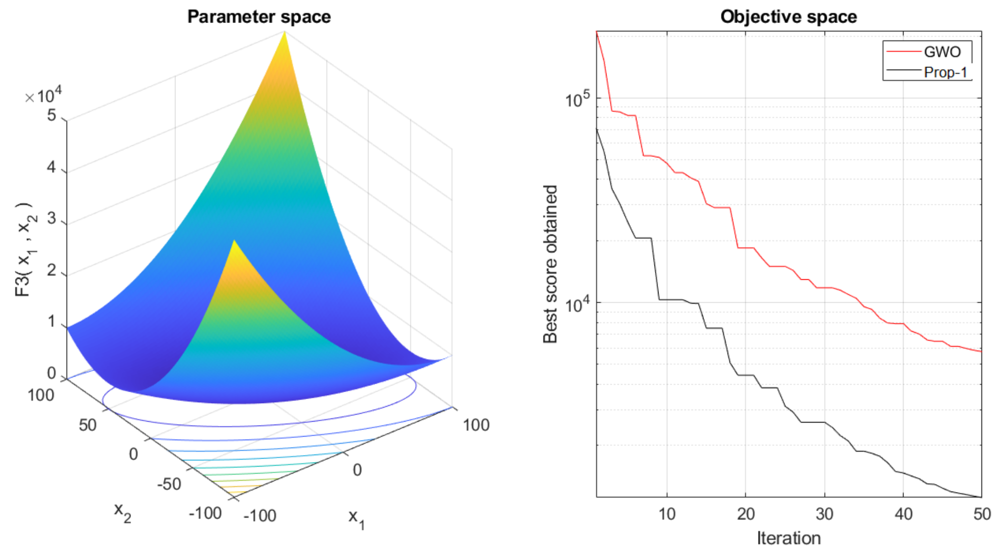

Fig.5.6 (a) F3 -Search space convergence (b) Optimised score of prop-1 vs GWO

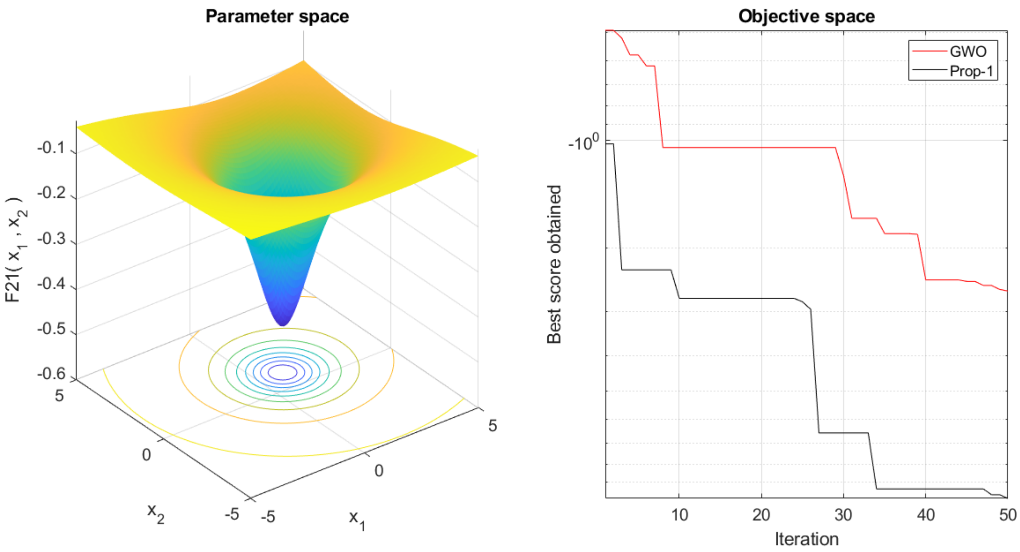

Fig.5.7 (a) F21- Search space convergence (b) Optimised score of prop-1 vs GWO

The performance of the proposed algorithm is compared with GWO optimiser. The proposed algorithm quickly converges to the solution i.e. reaches to the target solution very fast with less number of iterations. In fig 5.7 b, at 20^{th} iteration or at 35^{th} iteration, there is a big gap in performance (fitness score value). The proposed Optimisation algorithm (Prop-1) out performs the GWO algorithm almost by an order. When a network of Visual sensors is deployed then due to occlusion, images get impaired and are poorly captured by the nearby camera/sensors. There is an immediate need to get alternate VSNs without disrupting the service/application. Hence the selection of alternate visual sensors (alpha wolf) should be very fast to monitor the dynamic movement of the objects. The proposed optimisation algorithm is used for localization and tracking of objects, it avoids the local optimal solutions and responds with global best / global sub-optimal solutions with less number of iterations.

Conclusion

In this paper, a novel method for optimizing the coverage of visual sensors is proposed by adjusting their locations, using a meta heuristic algorithm GWO. The proposed GWO based approach is computationally lighter and faster in execution. Dwelling area and natural habitat of animal kingdom is mapped to surveillance space, obstacles and feasibility areas, considering a wildlife monitoring application. The experimental evaluation using View Points (habitat), obstacles, and feasible camera locations is modeled as optimization problem. The proposed work gives an economical and cost-effective solution for wild life surveillance application. From the modeling, simulation and investigation, the proposed algorithm performs better yielding less number of iterations. Lesser number of iterations implies significant reduction in computational time (delay) and therefore for non-delay tolerant networks mainly VSNs, the proposed algorithm is a better substitute. The proposed algorithm is converging at a much faster rate and hence is one of the best options for wild life monitoring application.

Future scope of work

Performance comparison with other GA algorithms like PSO, Cuckoo search etc. can be explored for future performance enhancement.

References

[1] Yabuta, K., & Kitazawa, H. (2008). Optimum camera placement considering camera specification for security monitoring, In *International symposium on circuits and systems,* IEEE, (pp. 2114–2117).

[2] Indu, S. et al. (2009). Optimal sensor placement for surveillance of large spaces. In 3rd *ACM/IEEE international conference on distributed smart cameras,* (pp. 1–8).

[3] Rourke, J. O. (1987). *Art gallery theorems and algorithms*. Oxford: Oxford University Press.

[4] Indu, S., Garg, R., & Chudhury, S. (2011). Camera and light source placement, a multi objective approach. In *3rd national conference on computer vision pattern recognition image processing and graphics*. IEEE, (pp. 187–191).

[5] Xu, Y. C. L., Lei, L., & Hendricks, E. A. (2011). *Camera network coverage Improving by PSO* (p. 458283). EURASIP Journal on Image and Video Processing: Springer.

[6] Mirjalili, S., et al. (2014). A Grey wolf optimizer. *Advances in Engineering Software, 69,* 46–61.

[7] Sujitha, J., & Baskaran, K. (2018). Genetic GWO based channel estimation in wireless communication system. *Wireless Personal Communications, 99*(2), 965–984.

[8] Mohanty, S., et al. (2016). A new MPPT design using GWO technique for photovoltaic system under par- tial shading conditions. *IEEE Transactions on Sustainable Energy, 7*(1), 181–188.

[9] Banu, S. S., & Baskaran, K. (2018). Hybrid FGWO based FLCs modeling for performance enhancement in wireless body area networks. *Wireless Personal Communications, 100*(3), 1163–1199.

[10] Li, L., et al. (2017). Modified discrete grey wolf optimizer algorithm for multilevel image thresholding. *Computational Intelligence and Neuroscience*. https://doi.org/10.1155/2017/3295769.

[11] R. Kumar and B. Singh (2013) , "Enhanced availability rate and load balance in emerging heterogeneous wireless network using proactive group VHO algorithm," *International Journal of Future Generation Communication and Network*, SERSC, Korea, vol. 6, no.1, pp.9-45.

[12] R. Kumar and B. Singh (2013), "Reduced VHO latency using EVM measurement and AdHoc cooperation for heterogeneous ubiquitous networks," *International Journal on Information and Communication Technology (IJICT)*, Inderscience Publishers, Switzerland, vol. 5, no.1, pp.45-63.

[13] B. Shahrokhzadeh and M. Dehghan, ''A distributed game-theoretic approach for target coverage in visual sensor networks,'' *IEEE Sensors J.*, vol. 17, no. 22, pp. 7542–7552, Nov. 2017.

[14] C.-F. Cheng and K.-T. Tsai, ''Distributed barrier coverage in wireless visual sensor networks with β-QoM,'' *IEEE Sensors J.*, vol. 12, no. 6, pp. 1726–1735, Jun. 2012.

[15] D. G. Costa, L. A. Guedes, F. Vasques, and P. Portugal, ''QoV: Assessing the monitoring quality in visual sensor networks,'' in *Proc. IEEE 8th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2012, pp. 667–674.

[16] K. R. Konda, N. Conci, and F. De Natale, ''Global coverage maximization in PTZ-camera networks based on visual quality assessment,'' *IEEE Sensors J.*, vol. 16, no. 16, pp. 6317–6332, Aug. 2016.

[17] F. Yap and H.-H. Yen, ''A survey on sensor coverage and visual data capturing /processing/transmission in wireless visual sensor networks,'' *Sensors*, vol. 14, no. 2, pp. 3506–3527, Feb. 2014.

[18] V. P. Munishwar and N. B. Abu-Ghazaleh, ''Coverage algorithms for visual sensor networks,'' *ACM Trans. Sensor Netw.*, vol. 9, no. 4, pp. 1–36, Jul. 2013.