



A Grey Wolf Optimization Based Algorithm for Optimum Camera Placement

Ajay Kaushik¹ · S. Indu² · Daya Gupta¹

Published online: 7 February 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

Camera placement is very important for surveillance applications. Proposed work presents a new method of optimum placement of visual sensors for maximum coverage of the predefined surveillance space. The surveillance space is modeled as priority areas (PAs), obstacles and feasible locations for placement of cameras. We are using PTZ (pan, tilt, zoom) cameras, which not only reduces occlusion due to randomly moving objects in the PA but also increases the covered area compared to pin hole cameras. The proposed approach will be useful for crowd monitoring in a big surveillance space holding multiple events and having multiple entrances. The problem of optimum camera placement 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 faster as compared to Genetic Algorithm (GA) based camera placement and Particle Swarm Optimization (PSO) based camera placement algorithm. The concept is validated using simulation as well as the experimental results.

Keywords Camera placement · GWO · Obstacles · Pan · Tilt · Zoom · Priority area · Visual sensors · FoV · Surveillance

1 Introduction

The use of video surveillance has become very popular for investigation purpose as they act as a deterrent of crime. Therefore, surveillance cameras are installed in both indoor and outdoor environments [1, 2]. A camera network is essential for surveillance of indoor or outdoor spaces, as a single camera cannot cover discrete events in a large surveillance

✉ Ajay Kaushik
ajaykaushik777@gmail.com

S. Indu
s.indu@dce.ac.in

Daya Gupta
dgupta@dce.ac.in

¹ Computer Science and Engineering, Delhi Technological University, Delhi, India

² Electronics and Communication Engineering, Delhi Technological University, Delhi, India

space. The major challenge for such a camera network is to cover PAs in a surveillance space, like multiple entrances 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, we need an algorithm that determines optimum locations of the cameras along with their pan, tilt and zoom levels [3, 4].

Identifying optimum configuration of camera network for such coverage is a combinatorial optimization problem. It will be difficult for simple search techniques to determine optimum placement configurations [5]. GA [4, 6] and PSO [7] have been used in the past for camera placement problem. Apart from GA and PSO, meta heuristics approach GWO [8] has shown good results in a variety of fields such as relay node (RN) placement problem [9–13], channel estimation in wireless communication [14], photovoltaic systems [15], wireless body area networks [16] and image processing [17]. GWO is inspired by the behaviour 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 behaviour [8, 15]. 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. This motivated us to use GWO as an optimization algorithm for developing an optimum camera placement strategy. Author's contributions to the paper are as follows.

- In this paper, we propose a camera placement algorithm for covering each PA at least by 3 cameras so as to enhance the information content of images and to reduce the effect of occlusion due to randomly moving obstacles. Even if randomly moving obstacle blocks the PA covered by one camera, the other two cameras will surface our requirement.
- The paper proposes a meta-heuristics approach GWO that optimizes the coverage of discrete PAs in a surveillance space by calculating optimum coordinates of the cameras along with their pan, tilt and zoom levels. GWO is computationally lighter as compared to GA and PSO and is faster in execution.

The article is divided into 7 sections. A literature review is presented in Sect. 2. The problem statement is explained in Sect. 3. Mathematical formulation of the floor plan and coverage matrix are discussed in Sect. 4. GWO mapping with the current problem and GWO implementation are discussed in Sect. 5. Results are explained in Sect. 6. We conclude our work in Sect. 7.

2 Literature Review

Several approaches have been proposed in the past to solve optimum guard location problem for a polygon area, for example, art gallery problem (AGP) [5] [18–20] that aims to visualize maximum area with minimum number of guards. The solutions presented to AGP were based on assumptions like unlimited field of view and infinite servo precision that makes them unsuitable for real world applications. The effects of random occlusion in offline camera placement were initially addressed by [21] which was later continued by [22] using a pin hole camera. Authors in [23] used binary integer programming (BIP) method for optimizing the orientations of a PTZ camera. The BIP method is computationally

complex and is not useful for large number of cameras. Authors in [24] used cheap motion sensors for real time control of PTZ cameras, but they did not optimize pan and tilt angles of a PTZ camera and also compromised with the image quality. Authors in [25] addressed the problem of detecting an unexpected activity from a large surveillance space. Fore-ground extraction method was adopted to mark people and objects in a surveillance space. Outliers from the daily surveillance patterns of a city were identified and considered as unexpected activity or area of interest. Authors in [26] used GA for optimizing pan and tilt angles of a PTZ camera for photogrammetric network. However, the algorithm is too complex to be used for 3D spaces. Authors in [4] used GA to optimally cover discrete PAs in a surveillance space with minimum required zoom incorporating the static obstacles. Authors in [7] used PSO to find the optimum camera orientations for 2D space by keeping the camera location fixed. Authors in [27] presented a camera placement approach that aims at minimizing the occlusion for object detection and tracking applications. Authors in [28] presented a framework that finds optimum number of sensors with their locations and orientations in a 3D terrain having multiple objectives. The framework finds a trade-off between conflicting objectives of covering maximum area at a low acquisition cost of sensor deployment. Authors in [29] proposed a video surveillance system based on face detection and recognition method that collects videos from multiple surveillance cameras to track and detect people over space and time. Authors in [30] used simulated annealing to find optimum camera configurations in a multi-camera system. The algorithm performs similar to BIP for small scale problems but performs better than BIP for large scale problems. Authors in [31] proposed a two phase camera placement algorithm based on BIP to find optimum camera coordinates in a large surveillance space. Phase 1 finds minimum number of cameras to cover a surveillance space while phase 2 achieves maximum coverage with the cameras established from phase 1. A low cost camera placement algorithm for bridge surveillance was proposed in [32]. They used uniqueness score and local search algorithm (ULA) that gives greater importance to the areas often left uncovered by other cameras. ULA yields better results than greedy algorithm and GA in terms of coverage and average surveillance cost.

Camera placement problem in a surveillance space is analogous to RN placement problem in WSN. Authors in [9–13] addressed RN placement problem using meta heuristic algorithms. In [9], if a part of network is damaged, the number of packets reaching the sink is decreased. So for correcting such cases, RNs are optimally placed using GWO to reactivate the network. GWO was used again in [10] to optimize the placement of RNs in a WSN. They used convex hull approach to restrict the deployment area of RNs and used alpha shapes to identify the structure and boundary of obstacles. Authors in [11] used BAT algorithm, interior search algorithm and moth flame optimizer to find the optimum location of RNs to achieve full coverage. Among the three bio-inspired algorithms used in [11], MFO required the least number of RNs to cover the entire deployment region. Authors in [12] used PSO to optimally place RNs in a WSN. They focused on minimizing the average distance between a RN and a sensor node to achieve an energy efficient network. An enhanced Ant Bee Colony (ABC) based RN deployment algorithm was proposed in [13] to extend the lifetime of a network. The first phase of the algorithm includes construction of network backbone structure by deploying RNs in the network, while in the second phase; the RNs are optimally placed using ABC. Although similar, relay node placement does not use directional sensors like a camera. We consider camera sensor that involves optimization of a variety of parameters like its location, pan angle, tilt angle and zoom level.

The proposed camera placement algorithm enhances the information content of images as well as reduces the effects of randomly moving obstacles. Also, the proposed method

is using a better optimization technique (GWO) which converges faster than the existing methods.

3 Problem Statement

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, pose (pan and tilt angles) and zoom level of each camera to achieve maximum coverage of all predefined PAs 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 PAs by at least 3 cameras 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 Coverage of Priority Points

Past research [4] had considered only static obstacles and the effects of randomly moving obstacles were overlooked. In order to achieve optimum coverage of the PAs in a large surveillance space with N cameras and randomly moving obstacles, we propose each PA to be covered by a minimum of 3 cameras. Let us consider the PAs in a large surveillance space covered by 3 cameras (Fig. 1), the following cases may occur.

3.1.1 Case I—Pillar as an Obstacle

In Fig. 1a we have 4 PAs, 3 cameras and one obstacle. Even with the presence of big obstacle like a pillar, we could place the cameras in such a way that all PAs are covered by 3 cameras from different angles.

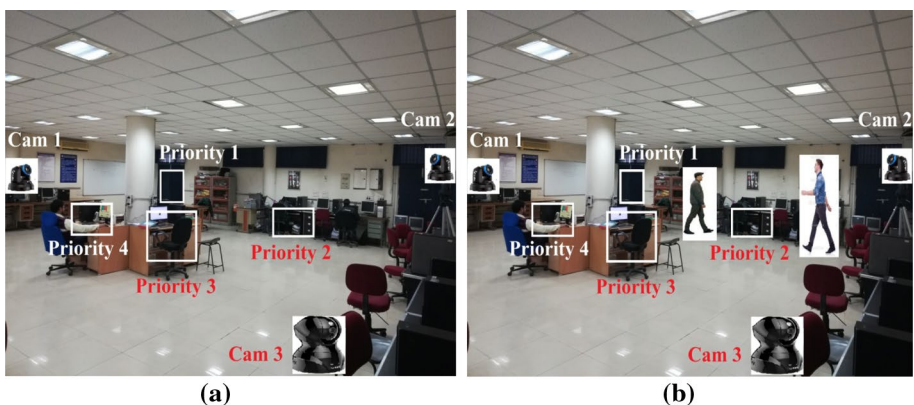


Fig. 1 **a** All PAs covered by 3 cameras. **b** FoV of cam 1 and cam 2 blocked by randomly moving obstacles (men)

3.1.2 Case II—View of 2 Cameras Blocked by Random Obstacles at Different Times

Figure 1b shows that the randomly moving obstacles (men in this case) block the view of cam 1 and cam 2 at time t_1 and t_2 respectively ($t_1 \neq t_2$). In this case, cam 3 and either of cam 1 or cam 2 cover PA 2 from different angles.

3.1.3 Case III—View of 2 Cameras Blocked by Random Obstacles at the Same Time

Figure 1b shows the worst case scenario where randomly moving obstacles (men) block the view of cam 1 and cam 2 at the same time t . Even in the worst-case scenario, PA 2 is covered by one camera (cam 3).

We propose a camera placement algorithm that covers all predefined PAs with minimum 3 cameras to reduce the effects of occlusion due to randomly moving obstacles. This also improves information content in images as the surveillance area is covered from 3 different angles. The angle between field of view (FoV) of adjacent cameras is equal to or slightly greater than 60 degrees to allow cameras to cover a larger space.

3.2 Camera Model

PTZ camera is used to capture images as it can cover bigger space in comparison to a pin hole camera as it is panning continuously. Some of the important parameters of PTZ camera are following.

FoV of a simple camera is presented in Fig. 2a. PTZ camera can rotate at an angle of $\pm\theta$ along its tilt and pan axis to have an extended FoV as shown in Fig. 2b.

Figure 3a shows the camera model used in [23]. A rotation along x-axis corresponds to tilt (θ_1) while a rotation around y-axis corresponds to pan (θ). Figure 3b shows the camera model used in this work incorporating zoom limits as a function of depth of view. The depth of view is the distance between the nearest and the farthest object appearing in the FoV of a camera. The nearest and the farthest distance are denoted by near focus plane and far focus plane respectively as shown in Fig. 3b.

In the proposed algorithm, we aim to cover each PA by 3 cameras. Figure 4a shows that the PA c is covered by FoV of 3 cameras Cam 1, Cam 2, and Cam 3. We introduce the concept of line of sight of a camera that helps us to figure out the effect of an obstacle on the

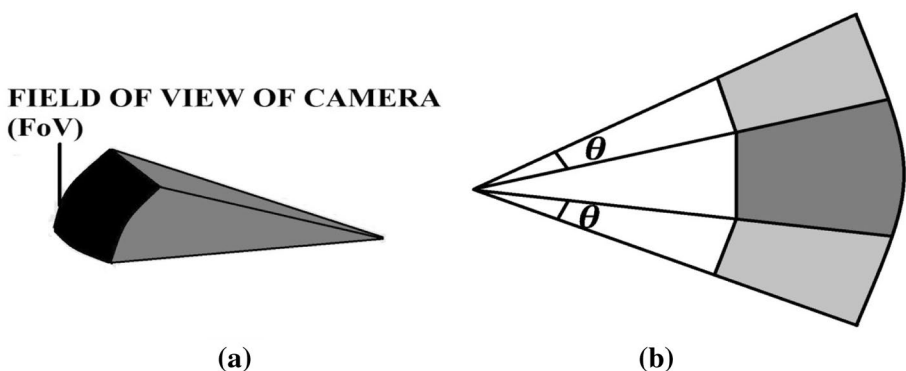


Fig. 2 a FoV of a simple camera. b Extended FoV of a PTZ camera

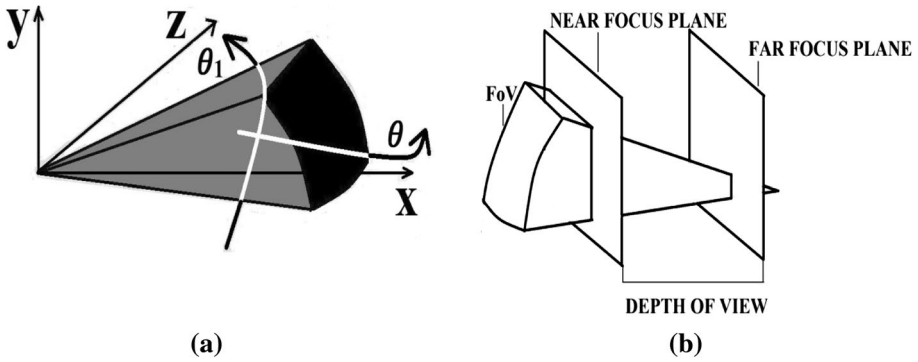


Fig. 3 a Camera model used in [23]. b The depth of view of a PTZ camera

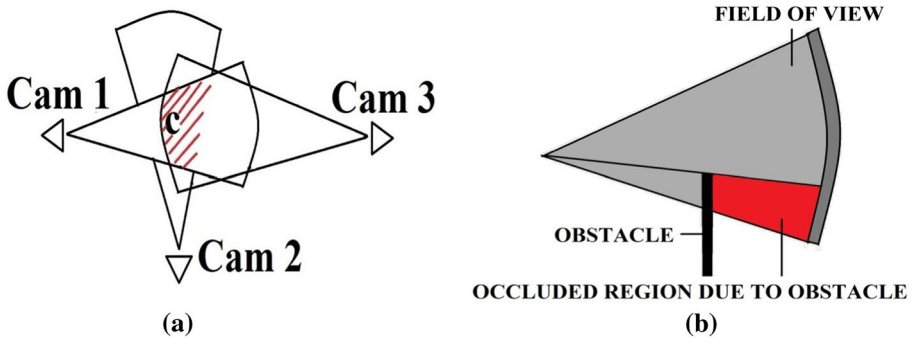


Fig. 4 a PA c covered by 3 cameras. b FoV occluded by an obstacle

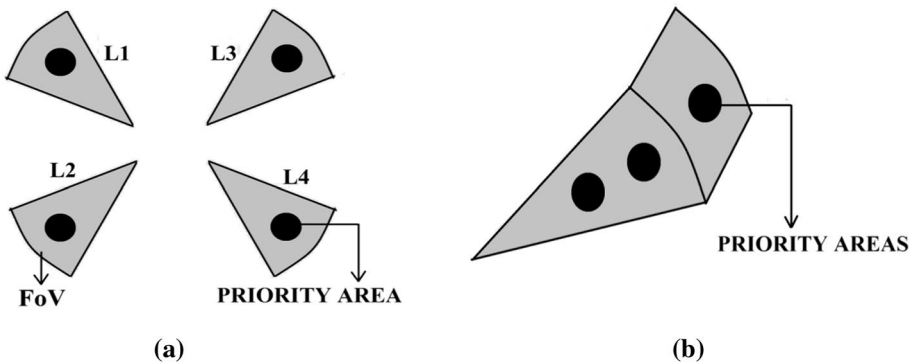


Fig. 5 a Multiple cameras required to cover distributed PAs. b Single camera covering all PAs

FoV of a camera. If an obstacle occludes the FoV of a camera from the line of sight, the occluded coverage area is removed from the FoV as shown in Fig. 4b.

Let us imagine there are 4 discrete PAs located at L1, L2, L3 and L4 in the surveillance space as shown in Fig. 5a. Coverage of these discrete PAs depends on their distribution in the

surveillance space and location of the cameras. For instance, we need 4 cameras to cover the PAs shown in Fig. 5a while the PAs in Fig. 5b can be covered using a single camera. The proposed work aims to achieve optimum location of the camera for covering maximum such PAs.

3.3 Surveillance Space Model

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 FoV of three cameras with minimum required resolution. In this paper, user defines a set of high activity PAs, location of the obstacles with its shape and size and feasible locations for camera placement using a user friendly graphical user interface (GUI). A GUI shown in Fig. 6 defines PAs, obstacles and feasible locations for camera placement with cubical dimensions $m * m * m$ where m is the largest side of the PA/obstacle.

4 Mathematical Formulation of the Floor Model

We consider a floor plan of $m * m * m$ grid points. PAs are denoted by matrix P as shown in Eqs. 1 and 2.

$$P = \left| p_{xyz} \right|_{m*m*m} \tag{1}$$

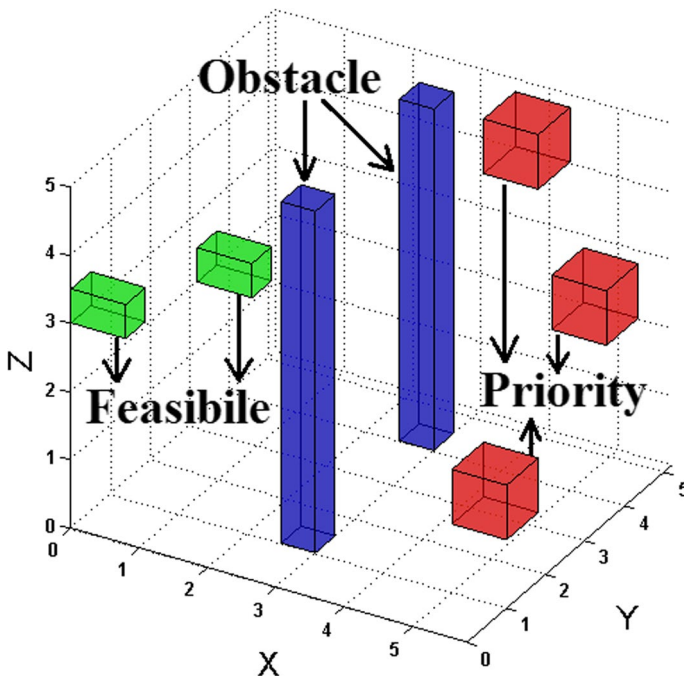


Fig. 6 Feasible locations for camera placement, obstacles (pillars) and PAs defined with help of GUI

$$p_{xyz} = \begin{cases} 1, & \text{if } x, y, z \text{ is the priority area} \\ 0, & \text{else} \end{cases} \tag{2}$$

Feasible location for camera placement is denoted by matrix F as shown in the Eqs. 3 and 4.

$$F = \left| f_{xyz} \right|_{m*m*m} \tag{3}$$

$$f_{xyz} = \begin{cases} 1, & \text{if } x, y, z \text{ is a feasibility area} \\ 0, & \text{else} \end{cases} \tag{4}$$

Obstacles are denoted by matrix O as shown in Eqs. 5 and 6.

$$O = \left| o_{xyz} \right|_{m*m*m} \tag{5}$$

$$o_{xyz} = \begin{cases} 1, & \text{if } x, y, z \text{ is a obstacle} \\ 0, & \text{else} \end{cases} \tag{6}$$

For each camera location, visibility measurement of all points covered by the camera is determined and defined as visibility matrix. Let a camera located at (x_c, y_c, z_c) with a pan angle p_c , tilt angle t_c and a zoom level Z_m covers a 3D point (x_p, y_p, z_p) . Visibility matrix of the 3D point viewed by a camera with different pose (p_c, t_c) and zoom levels can be defined by 9 dimensional matrix (Eq. 7).

$$A = [x_p, y_p, z_p, x_c, y_c, z_c, p_c, t_c, Z_m] \tag{7}$$

The 9D matrix is reduced to 4D matrix using Eqs. 8–10 as 9D matrix is very inconvenient to work with. We use array indexing to convert 3D coordinates (x_c, y_c, z_c) of a camera into 1D point (i) as shown in Eq. 8.

$$i = (x_c - 1) * N * N + (y_c - 1) * N + z_c \tag{8}$$

N in Eq. 8 represents the total number of cameras. Similarly, each pose can be mapped into 1D point, represented by j in Eq. 9. These functions are effectively creating a one-dimensional index for multi-dimensional points.

For each pose (p_c, t_c)

$$j = M * (p_c - 1) + t_c \tag{9}$$

M in Eq. 9 represents various pan or tilt angles of a camera.

$$Zoom(Z_m) = (high Z_m L - low Z_m L) * \frac{Z_m}{Z_m L} \tag{10}$$

$Z_m L$ in Eq. 10 represents the number of various zoom levels of a camera and ‘ Z_m ’ range from 1 to $Z_m L$. Now the visibility matrix shown in Eq. 7 is reduced to 4 dimensional

matrix. The performance measure of all visible points due to the camera pose are calculated and represented as 4D visibility matrix A in Eq. 11. Each term in the matrix A denotes the visibility measure of a point k viewed by a camera located at point i with pose j and zoom level Z_m .

$$A = [a_{ijkZ_m}]_{m^3 * M^2 * m^3 * Z_m L} \tag{11}$$

where

$$a_{ijkZ_m} = \begin{cases} 1, & \text{if point } k \text{ is covered} \\ 0, & \text{else} \end{cases} \tag{12}$$

From the visibility matrix of all cameras, we should be able to calculate the number of cameras covering a point k . The covered points are now classified into priority and non priority areas. The PAs are further classified into PAs covered by 3 cameras, 2 cameras, 1 camera and placed in A_3, A_2 , and A_1 respectively. A point k is assumed to be covered by 3 cameras if it lies in the FoV of 3 cameras placed at different locations with different pose and zoom levels.

Figure 7 represents the FoV of 3 cameras cam 1, cam 2 and cam 3 to cover the PAs P1, P2, P3, P4, P5 and P6. The visibility matrix of the PAs P1 and P2 covered by 3 cameras (cam1, cam 2 and cam 3) will be denoted by $A_3(i, j, k, Z_m)$, visibility matrix of the PAs P3 and P4 covered by 2 cameras (cam 1 and cam 3) will be denoted by $A_2(i, j, k, Z_m)$, visibility matrix of the PAs P5 and P6 covered by 1 camera (cam 3) will be denoted by $A_1(i, j, k, Z_m)$ and visibility matrix of non PAs NP1 and NP2 will be denoted by $A_n(i, j, k, Z_m)$.

4.1 Coverage Matrix

We define the coverage matrix based on following inferences

1. A single lens is used as an optical sensor. Focal distance from the object is measured from the center of the lens.

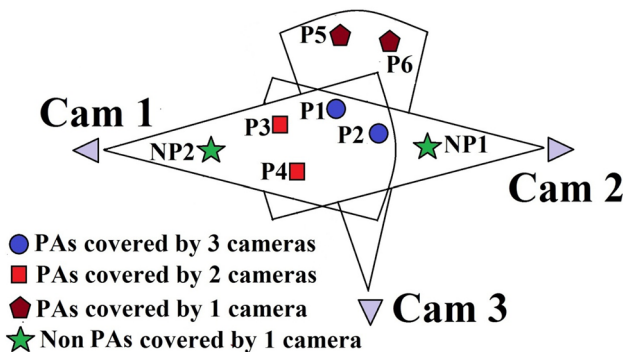


Fig. 7 Priority areas (PAs) covered by 3 cameras, 2 cameras, 1 camera and Non PAs covered by 1 camera

2. Lens aperture is assumed to be constant throughout the process.
3. Geometric distortion effect and blurring of objects has been ignored.

Coverage matrix is defined in Eqs. 13–18.

$$C = n \sum_{\text{priority}(3\text{-cam})} P_x + u \sum_{\text{priority}(2\text{-cam})} Q_y + v \sum_{\text{priority}(1\text{-cam})} R_z + w \sum_{\text{nonpriority}} H_s, 0 < n, u, v, w, C < 1 \tag{13}$$

where

$$P_x = A_3(i, j, k, Z_m) \text{ if priority area is covered by 3 cameras} \tag{14}$$

$$Q_y = A_2(i, j, k, Z_m) \text{ if priority area is covered by 2 cameras} \tag{15}$$

$$R_z = A_1(i, j, k, Z_m) \text{ if priority area is covered by 1 cameras} \tag{16}$$

$$H_s = A_n(i, j, k, Z_m) \text{ for non priority areas} \tag{17}$$

$A(i, j, k, Z_m)$ denotes the visibility matrix at a point in the surveillance space calculated in Eq. 11. n, u and v denotes the weightage given to the PAs covered by 3 cameras, 2 cameras and 1 camera respectively and w denotes the weightage given to the non PAs that are covered.

$$n + u + v + w = 1 \tag{18}$$

The coverage matrix (or fitness function) in Eqs. 13–18 should maximize the PAs covered by 3 cameras and minimize the PA covered by 2 cameras and 1 camera respectively. To achieve this objective, relative measure of the weighing functions is kept as $n > u > v > w$ which give higher weight to the PAs covered by 3 cameras thus reducing the probability of occlusion. We maximize the value of C (Eq. 13) using GWO.

The coverage matrix is subject to the constraint

$$\theta_{\text{adjacent}} \geq 60^\circ \tag{19}$$

where θ_{adjacent} is the angle between 2 adjacent cameras in the surveillance space. Equation 19 ensures that all cameras in a surveillance space are separated by a minimum angle and allows the cameras to cover a larger space.

5 GWO Mapping

The proposed work uses meta heuristic algorithm GWO [8] to find optimum locations, pose and zoom levels for visual sensors to cover each PA in a surveillance space by at least 3 cameras. A wolf ($W_i, 1 < i < N, N = \text{total number of cameras}$) 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, z coordinates), pose (pan and tilt angles) and zoom level in form of a vector shown in Eq. 20.

$$W_i = |X_{\text{coord}}(i)| |Y_{\text{coord}}(i)| |Z_{\text{coord}}(i)| |\mathbf{Pan}(i)| |\mathbf{Tilt}(i)| |\mathbf{Zoom}(i)| \tag{20}$$

where $1 < i < N$.

Each wolf will be assigned a weightage according to the fitness function (coverage matrix in Eq. 13). The values of the vector (Eq. 20) will be the optimum solution which gives optimum position, pose and zoom of a camera to achieve maximum coverage of all PAs with

each PA being covered by at least 3 cameras. The algorithm converges when the value of C in Eq. 20 becomes maximum.

From Eq. 20, we have

$$X_{coord}(i) = [x_1, x_2, \dots \dots x_N] \tag{21}$$

$$Y_{coord}(i) = [y_1, y_2, \dots \dots y_N] \tag{22}$$

$$Z_{coord}(i) = [z_1, z_2, \dots \dots z_N] \tag{23}$$

Variable values $x_1, x_2, \dots \dots x_N$ in Eq. 21 represent the set of x coordinates for N cameras. Similarly variable values $y_1, y_2, \dots \dots y_N$ and $z_1, z_2, \dots \dots z_N$ in Eqs. 22 and 23 represent the set of y and z coordinates for N cameras.

$$Pan(i) = [p_1, p_2, \dots \dots p_N] \tag{24}$$

$$Tilt(i) = [t_1, t_2, \dots \dots t_N] \tag{25}$$

$$Zoom(i) = [Z_{m1}, Z_{m2}, \dots \dots Z_{mN}] \tag{26}$$

Variable values $p_1, p_2, \dots \dots p_N$ and $t_1, t_2, \dots \dots t_N$ in Eqs. 24 and 25 represent the set of pan angles and tilt angles for N cameras. Variable values $Z_{m1}, Z_{m2}, \dots \dots Z_{mN}$ in Eq. 26 represent the set of zoom level for N cameras.

In this way, a population of wolves in GWO represents a set of cameras 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 in Eq. 13. The optimization criterion is set for maximization. A GWO population comprising of N wolves (each wolf representing a camera) is shown in Fig. 8.

$$\begin{aligned}
 W_1 &= | X_{coord}(1) | Y_{coord}(1) | Z_{coord}(1) | Pan(1) | Tilt(1) | Zoom(1) \\
 W_2 &= | X_{coord}(2) | Y_{coord}(2) | Z_{coord}(2) | Pan(2) | Tilt(2) | Zoom(2) \\
 W_3 &= | X_{coord}(3) | Y_{coord}(3) | Z_{coord}(3) | Pan(3) | Tilt(3) | Zoom(3) \\
 W_4 &= | X_{coord}(4) | Y_{coord}(4) | Z_{coord}(4) | Pan(4) | Tilt(4) | Zoom(4) \\
 &\quad \cdot \\
 &\quad \cdot \\
 &\quad \cdot \\
 &\quad \cdot \\
 W_N &= X_{coord}(N) Y_{coord}(N) Z_{coord}(N) Pan(N) Tilt(N) Zoom(N)
 \end{aligned}$$

Fig. 8 GWO population of N wolves

5.1 GWO Implementation

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. 27 where N denotes the number of wolves in the population

$$\vec{X} = \{x[1], x[2], x[3] \dots \dots x[N]\} \tag{27}$$

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.

5.1.1 Encircling the Prey (Exploration)

Wolves encircle the prey in order to stop its movement. The encircling process can be mathematically formulated in Eqs. 28–29.

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_p(t) - \vec{X}(t) \right| \tag{28}$$

$$\vec{X}(t + 1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \tag{29}$$

Here t shows the current iteration, \vec{X}_p denotes the position vector of the prey and \vec{X} denotes the position vector of the wolf, $\vec{X}(t + 1)$ is the updated position of the wolf towards the prey. The coefficients \vec{A} and \vec{C} are calculated as per Eqs. 30 and 31.

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \tag{30}$$

$$\vec{C} = 2\vec{r}_2 \tag{31}$$

The value of \vec{a} is reduced from 2 to 0 over the iterations. Vectors \vec{r}_1 and \vec{r}_2 varies randomly in the interval [0,1].

5.1.2 Hunting the Prey (Exploitation)

We assume 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 \vec{X}_α , \vec{X}_β and \vec{X}_δ for the current iteration t and oblige all other wolves in the population (ω) to update their positions according to \vec{X}_α , \vec{X}_β and \vec{X}_δ as shown in Eqs. 32–34.

$$\vec{D}_\alpha = \left| \vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}_{W_i} \right|, \vec{D}_\beta = \left| \vec{C}_2 \cdot \vec{X}_\beta - \vec{X}_{W_i} \right|, \vec{D}_\delta = \left| \vec{C}_3 \cdot \vec{X}_\delta - \vec{X}_{W_i} \right| \tag{32}$$

In Eq. 32, X_{W_i} denotes the position of wolf W_i , \vec{X}_α is α wolf's position and \vec{D}_α is updated α position. Similarly, \vec{X}_β is β wolf's position and \vec{D}_β is the updated β position. \vec{X}_δ is δ wolf's position and \vec{D}_δ denotes the updated δ position. C_1, C_2, C_3 are calculated as per

Eq. 31. The positions $(\vec{X}_1, \vec{X}_2, \vec{X}_3)$ of a wolf for the current iteration are evaluated as per Eq. 33 [8].

$$\vec{X}_1 = \left| \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha) \right|, \quad \vec{X}_2 = \left| \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta) \right|, \quad \vec{X}_3 = \left| \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta) \right| \quad (33)$$

Coefficients A_1, A_2, A_3 are calculated as per Eq. 30. The wolf (W_i) finally updates its position towards the prey (optimum solution) as shown in Eq. 34.

$$\vec{X}(t + 1) = \left(\vec{X}_1 + \vec{X}_2 + \vec{X}_3 \right) / 3 \quad (34)$$

GWO hunt the prey on basis of $\alpha, \beta,$ and $\delta,$ therefore it is prone to be stuck at local optima. Although some exploration was done during encircling the prey, GWO needs more operators to carry out exploration and to search for better solutions.

5.1.3 Searching for the Prey (Exploration)

Coefficient vector \vec{A} plays an important role in movement of a wolf towards the prey (Eqs. 29–30). Figure 9 shows that \vec{A} take random values greater than 1 to enable the wolves diverge from the prey and search for better solutions [8].

Vector \vec{C} is another important factor used for exploration in GWO. As evident from Eq. 28, the value of distance (\vec{D}) depends on \vec{C} while encircling the prey. Instead of decreasing \vec{C} linearly, we deliberately keep \vec{C} as random as possible in the interval $[0, 2]$ to emphasize exploration [8]. This factor is very important for avoiding local optima in both initial as well as final iterations.

5.1.4 Convergence of GWO

During optimization of GWO, ω wolves iteratively improve their fitness according to $\alpha, \beta,$ and $\delta.$ When the improvement in the fitness of ω wolves reaches a threshold and

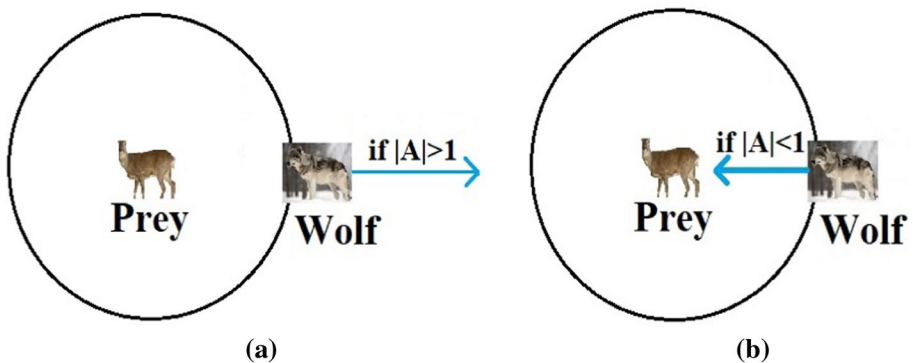


Fig. 9 a wolf diverging away from the prey to search for better solutions when $|A| > 1.$ b wolf moving towards the prey when $|A| < 1$

fitness of ω wolves cannot be further improved on the basis of the fittest solutions α , β and δ , GWO is said to converge [8]. The pseudo code for the proposed work is shown by Algorithm 1.

Algorithm 1 – GWO implementation for optimum camera placement

1. Initialize the population size P_t , Initialize the max numbers of iteration I_{max} .
 2. **while** $I < I_{max}$, **for each** iteration **do**
 3. **for each** wolf W_i **do**
 4. Clone the camera position, pose and zoom to W_i as per equation 20.
 5. Compute the fitness function of each wolf using equation 13.
 6. Update leader wolves $X_\alpha, X_\beta, X_\delta$. - three best wolves.
 7. Use Eq 30 and Eq. 31 to calculate the coefficient vectors.
 8. Update the wolves' (W_i) position using Eq. 34.
 9. **end**
 10. **end**
 11. Optimum camera configurations are obtained after final iteration/convergence.
-

6 Experimental Results

The proposed algorithm is implemented using Python and the floor plan is of size $100*100*5 \text{ m}^3$ which is divided into cubical grids of size $1*1*1 \text{ ft}^3$. For simulation, we place 50 cameras throughout the surveillance space to cover 60, 80 and 100 distributed PAs with static and dynamic obstacles. The locations of the static obstacles are shown in Table 1.

Figure 10a shows the random camera placement of 50 cameras in the surveillance space. The FoV of the cameras is represented by cones. The locations and pose of all cameras are calculated for maximum coverage of PAs at least by 3 cameras using the proposed algorithm. The optimum camera locations after implementing the algorithm are shown in Fig. 10b–d respectively for 60, 80 and 100 PAs. It is seen that all PAs are covered by at least one camera, some are covered by 2 cameras, but maximum PAs are covered by 3 cameras. The result is shown in Fig. 11.

Table 1 Static obstacles used for simulation

Obstacle no.	x	y	Z
1	62	45	5
2	34	22	5
3	24	76	5
4	91	49	5
5	56	68	5

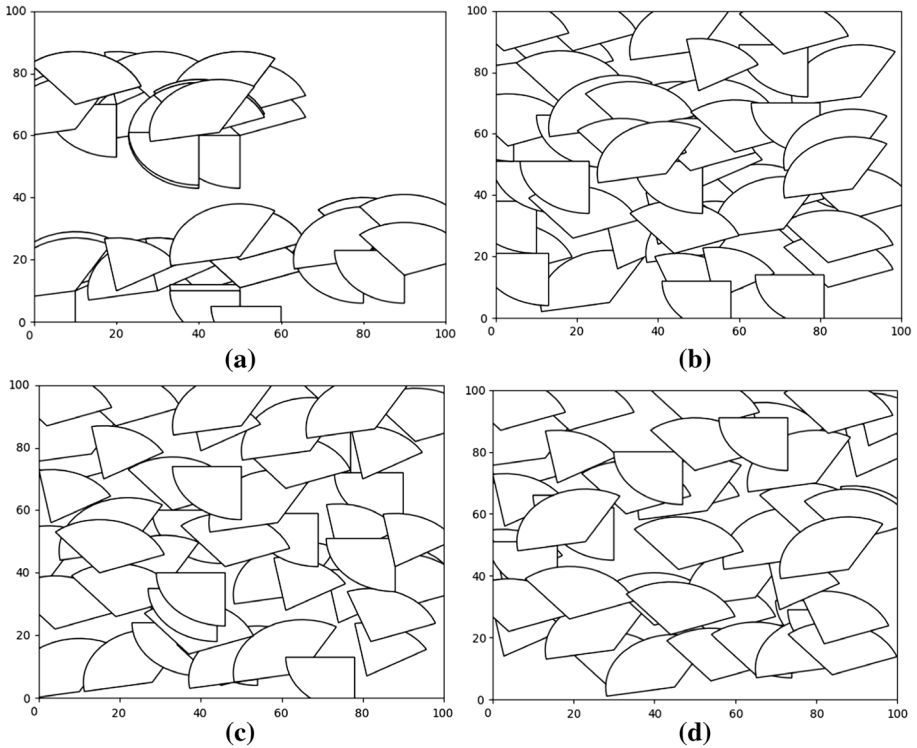


Fig. 10 **a** Random camera placement. **b** GWO based camera placement for coverage of 60 PAs. **c** GWO based camera placement for coverage of 80 PAs. **d** GWO based camera placement for coverage of 100 PAs

Fig. 11 Maximum PAs are covered by 3 cameras in GWO camera placement

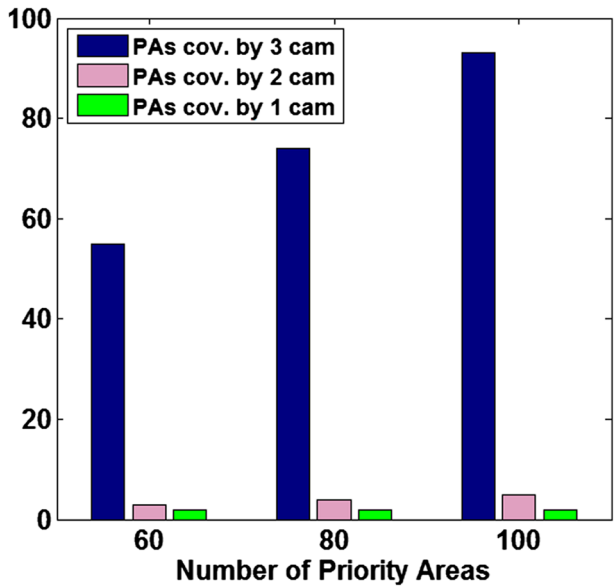


Table 2 shows the optimum camera coordinates and poses of 25 (out of 50) PTZ cameras calculated using the proposed algorithm.

GWO based camera placement is compared with random camera placement, GA camera placement [4] and PSO camera placement [7] for surveillance space coverage. Figure 12 shows that 50 cameras provide a coverage of 29, 66 and 78% for random camera placement, GA camera placement and PSO camera placement respectively; while in the proposed GWO camera placement algorithm, only 45 cameras provide 100 percent coverage which significantly reduces the hardware cost of any surveillance application.

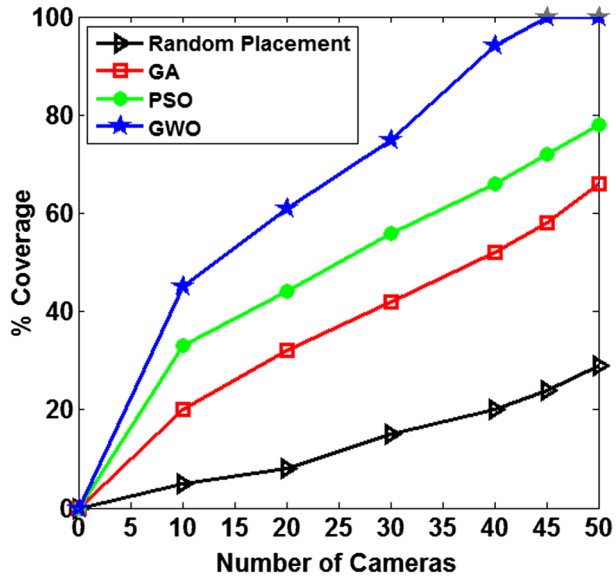
6.1 Experimental Evaluation

We conducted 3 experiments to validate the proposed algorithm. The aim of all experiments was to cover distributed PAs in a surveillance space of size $10*10*5 \text{ m}^3$, at least with 3 cameras per PA. We use 4 PTZ cameras for the experiments with static as well as random obstacles in the surveillance space. The use of 5 discrete clusters of PAs and 4 PTZ cameras covers all important activities in a big surveillance hall from different angles. The experimental evaluations are performed in Senior Machine Lab, DTU, DBMS Lab, DTU and Machine Shop, DTU which are cubical in shape. Senior Machine Lab, DTU has 2

Table 2 Optimum parameters of 25 cameras obtained from GWO implementation

Cameras	x	y	Z	Pan	Tilt	Zoom
1	38	42	4	15	45	2
2	53	86	3	50	15	9
3	73	41	2	47	33	8
4	8	7	3	-54	-48	1
5	5	74	3	-30	84	2
6	25	87	2	65	-64	2
7	47	80	5	-17	15	6
8	67	79	2	38	-39	9
9	23	51	4	10	-41	6
10	2	87	5	-42	85	4
11	45	6	4	36	45	5
12	30	16	5	58	-62	7
13	39	52	2	-48	-80	8
14	47	32	3	36	39	2
15	54	8	4	-34	60	5
16	52	18	4	-50	65	7
17	71	29	5	60	-80	8
18	13	78	3	75	44	6
19	27	66	3	15	35	5
20	19	87	5	-18	65	8
21	46	42	3	60	-77	7
22	53	61	2	-39	-38	2
23	33	60	3	45	-76	4
24	2	87	4	-44	66	6
25	47	80	4	37	45	4

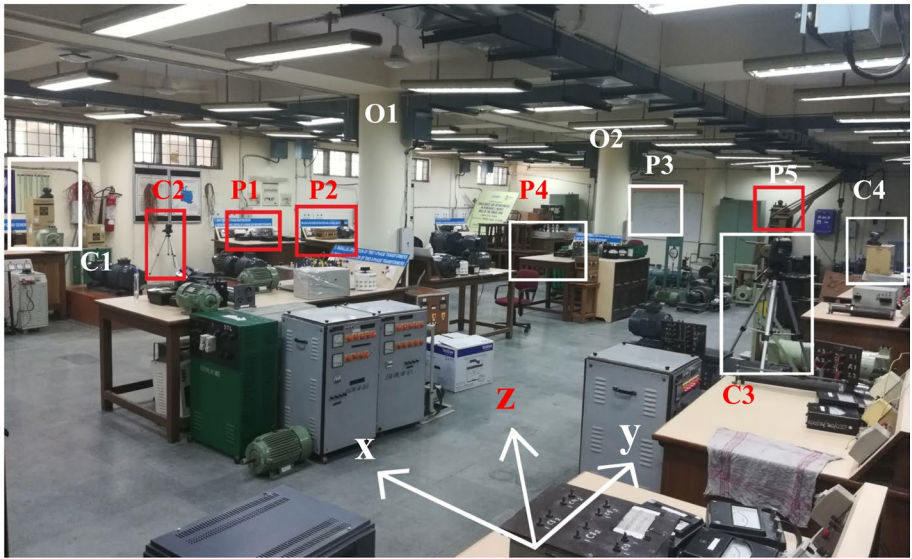
Fig. 12 Performance of GWO based camera placement compared with random camera placement, GA based camera placement and PSO based camera placement in terms of % coverage



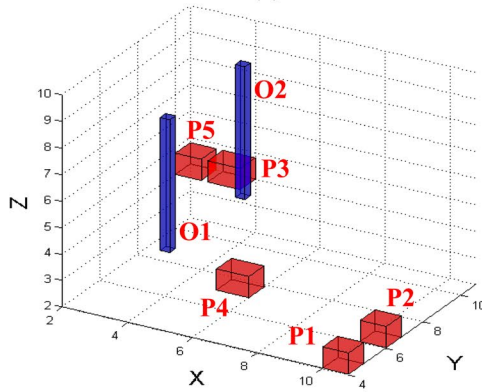
pillars at the center of the hall which we consider as obstacles. DBMS Lab, DTU has one obstacle at the center of the hall and has 2 high activity main doors. Machine shop DTU has 2 obstacles at the center of the shop and has 3 high activity main doors. Experimental evaluation of the proposed algorithm in 3 diversely structured surveillance halls with distributed PAs and obstacles allows us to validate our algorithm. We considered all locations on the wall as feasible location for camera placement. Main doors of a hall need to be covered with adequate number of cameras for any surveillance purpose as multiple doors are high activity areas. The PTZ camera used is shown in Fig. 13.

Fig. 13 PTZ camera used for experimental evaluation of the proposed algorithm





(a)



(b)



(c)

(d)

Fig. 14 Camera coverage performance for experiment 1. **a** Experimental Setup **b** GUI mapping of the experimental setup viewed from 30° angle, **c** Image taken from camera 1. **d** Image taken from camera 2. **e** Image taken from camera 3. **f** Image taken from camera 4

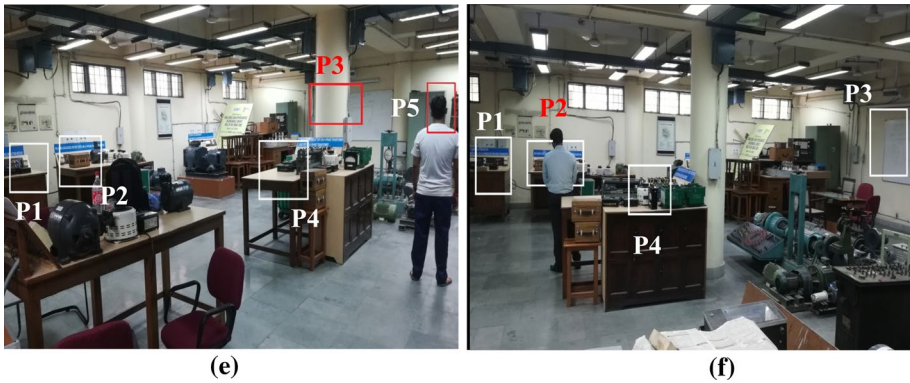


Fig. 14 (continued)

Table 3 Obstacle and PA coordinates for experiment 1

Areas	x	y	Z
Obstacle 1 (Pillar O1)	5	4	5
Obstacle 2 (Pillar O2)	5	8	5
Priority 1 (P1)	10	4	2
Priority 2 (P2)	10	6	2
Priority 3 (P3)	5	10	4
Priority 4 (P4)	5	7	2
Priority 5 (P5)	2	10	4

6.1.1 Experiment 1

We consider 5 distributed PAs (P1, P2, P3, P4 and P5) to be covered by 4 PTZ cameras (C1, C2, C3 and C4) in Senior Machine Lab, DTU of size 10*10*5 m³, having 2 static obstacles (O1 and O2) and 2 randomly moving obstacles as shown in Fig. 14a. GUI mapping of PAs and obstacles in the surveillance space viewed from an angle of 30° is shown in Fig. 14b. The obstacle and priority area coordinates are shown in Table 3.

We are considering P3 occluded by a pillar (Fig. 14e) and P5 and P2 occluded by randomly moving obstacles (Fig. 14e, f). We aim to cover all PAs with at least 3 cameras by optimally placing 4 PTZ cameras. The camera coordinates obtained from the proposed algorithm are shown in Table 4.

The images captured by 4 cameras are shown in Fig. 14c–f. From Fig. 14c–f, it is seen that all PAs are covered by at least 3 different cameras with static occlusion.

Table 4 Camera coordinates for experiment 1

Cameras	x	y	Z	Pan	Tilt	Zoom
C1	10	1	3	33	4	5
C2	10	3	2	46	-82	6
C3	0	3	3	64	30	5
C4	0	7	2	26	-12	4

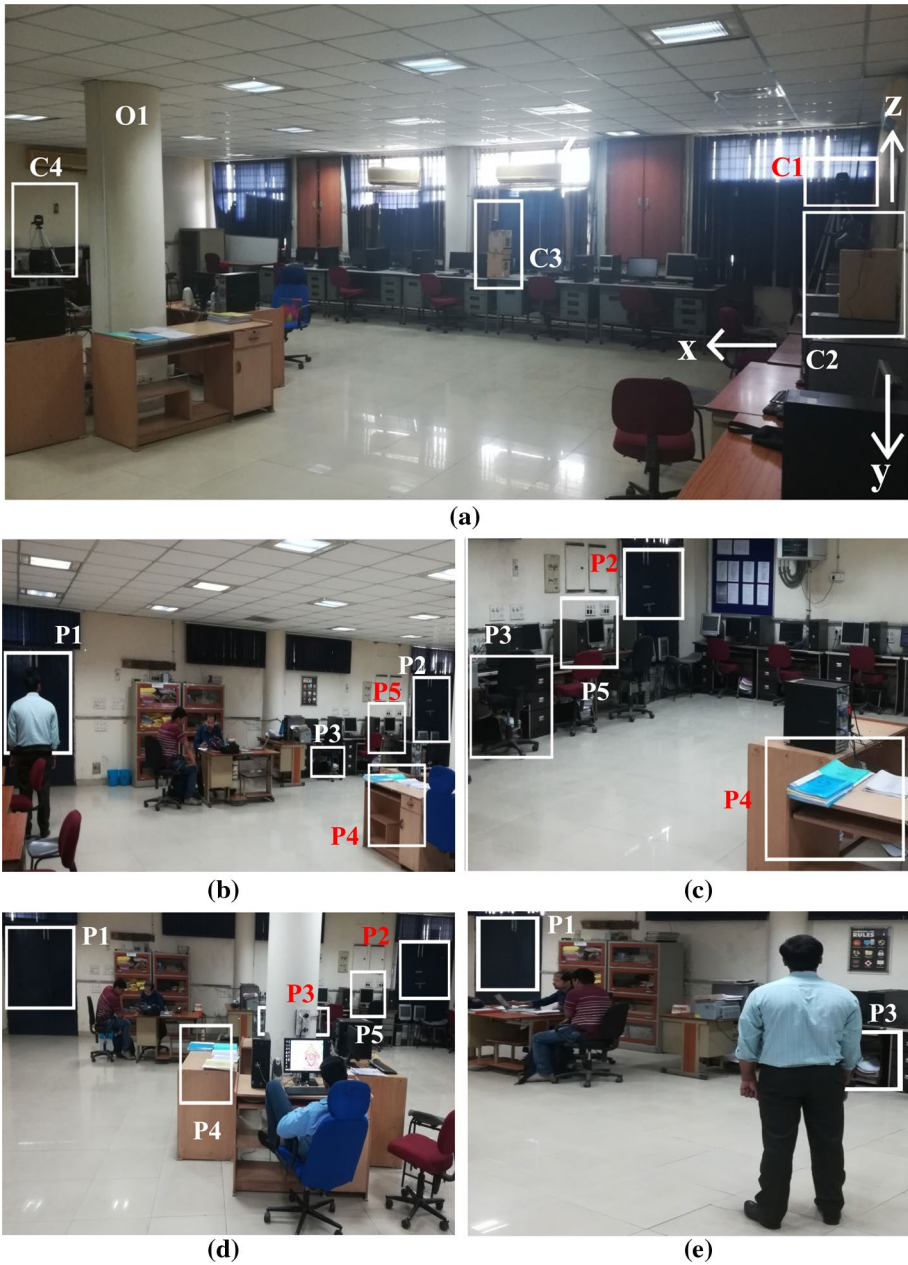


Fig. 15 Camera coverage performance for experiment 2. **a** Experimental setup. **b** Image taken from of camera 1. **c** Image taken from camera 2. **d** Image taken from camera 3. **e** Image taken from camera 4

However, after incorporating the dynamic occlusion, P2 and P5 are covered by 2 cameras. Dynamic occlusion will not last for long duration as our camera is continuously panning.

Table 5 Obstacle and PA coordinates used for experiment 2

Areas	x	y	z
Obstacle (Pillar O1)	5	5	5
P1	0	10	4
P2	10	10	4
P3	5	9	1
P4	4	5	2
P5	7	10	2

Table 6 Camera coordinates used for experiment 2

Cameras	x	Y	z	Pan	Tilt	Zoom
C1	0	2	4	55	12	4
C2	0	5	2	44	78	5
C3	5	0	2	33	-20	4
C4	10	3	3	54	-25	6

6.1.2 Experiment 2

In experiment 2, we consider 5 distributed PAs (P1, P2, P3, P4 and P5) to be covered by 4 PTZ cameras (C1, C2, C3 and C4) in DBMS Lab, DTU of size $10*10*5$ m³, having 1 static obstacle (O1) and 2 randomly moving obstacles as shown in Fig. 15a. The obstacle and PA coordinates are shown in Table 5.

We are considering P3 occluded by a pillar (Fig. 15d) and P1 and P3 occluded by randomly moving obstacles (Fig. 15b, e). We aim to cover all the PAs with at least 3 cameras by optimally placing 4 PTZ cameras. Camera coordinates obtained from the proposed algorithm are shown in Table 6.

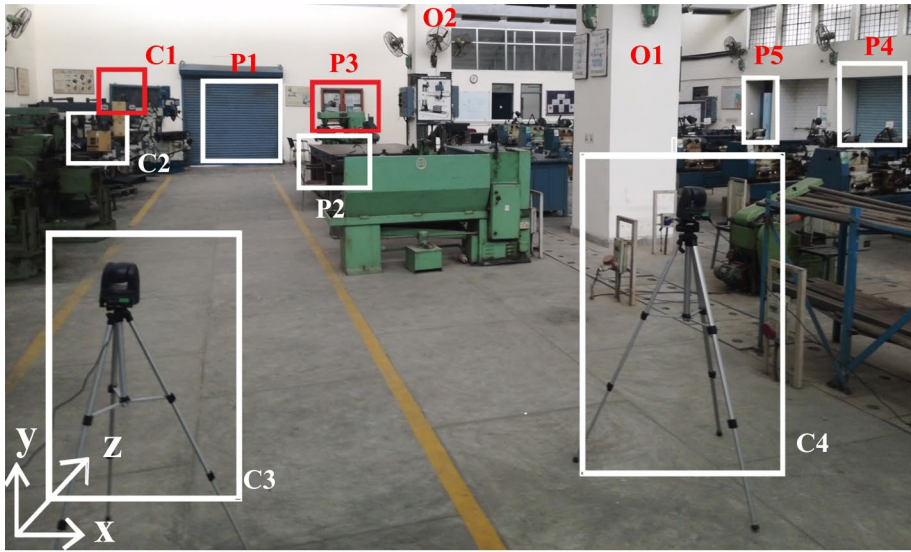
The images taken by 4 cameras are shown in Fig. 15b–e. It is visible from Fig. 15b–e that all the PAs are covered by at least 3 cameras considering static obstacles. However, after incorporating the randomly moving obstacles, P1 and P3 are covered by 2 cameras.

6.1.3 Experiment 3

We consider 5 distributed PAs (P1, P2, P3, P4 and P5) to be covered by 4 PTZ cameras (C1, C2, C3 and C4) in Machine shop, DTU of size $10*10*5$ m³, having 2 static and 2 randomly moving obstacles and 3 main doors as shown in Fig. 16a. The obstacle and priority area coordinates are shown in Table 7.

We are considering P5 occluded by a pillar (Fig. 16d) and P3 and P5 occluded by randomly moving obstacles (Fig. 16d, e). We aim to cover all PAs with at least 3 cameras. The camera coordinates obtained from the proposed algorithm to optimally cover all PAs are shown in Table 8.

The images captured by 4 cameras are shown in Fig. 16b–e. From the Fig. 16b–e, it is seen that all PAs are covered by at least 3 different cameras with static obstacles. However, after incorporating the randomly moving obstacles, P5 is covered by only 2 cameras. Since PA 5 is viewed from 2 different angles, most of the information in the PA 5 is covered.

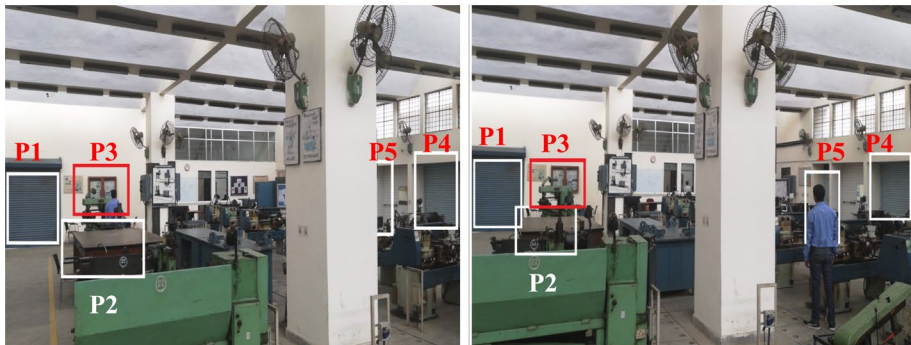


(a)



(b)

(c)



(d)

(e)

Fig. 16 Camera coverage performance for experiment 3. **a** Experimental Setup. **b** Image taken from camera 1. **c** Image taken from camera 2. **d** Image taken from camera 3. **e** Image taken from camera 4

Table 7 Obstacle and PA coordinates for experiment 3

Areas	X	y	Z
Obstacle 1 (Pillar O1)	5	4	5
Obstacle 2 (Pillar O1)	5	8	5
P1	1	10	4
P2	2	7	1
P3	2	10	3
P4	10	6	3
P5	10	9	3

Table 8 Camera coordinates for the experiment 3

Cameras	x	Y	Z	Pan	Tilt	Zoom
C1	0	8	2	22	54	3
C2	0	6	2	22	34	4
C3	1	0	3	-64	56	5
C4	3	0	3	12	-60	5

From the experimental evaluation, we made the following observations

1. Coverage of different sets of PAs by camera 1, 2, 3 and 4 is shown in Figs. 14, 15 and 16. All the PAs are covered by at least 3 cameras.
2. In case of randomly moving obstacle(s) blocking the view of one or more cameras, the PA is always covered by at least 1 camera.
3. For the PTZ cameras that are placed far away from the PA (Fig. 14d), a higher degree of zoom is used to get an accurate image. This explains the accuracy of the model used.
4. Experiment 1, 2 and 3 are a testimony to the fact that all the PAs are covered for a longer duration and with optimum image resolution.

7 Conclusion

In this work, we present a new method for optimizing the coverage of visual sensors by adjusting their locations, pose and zoom levels using a meta heuristic algorithm GWO. The proposed GWO based approach is computationally lighter and faster in execution. Since we are using priority areas, obstacles and feasibility areas, our approach will be useful in practical applications. Zoom level as a parameter ensures good quality and resolution of the images. The proposed algorithm covers each PA by 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. We have shown the experimental evaluation of our work using PAs, obstacles, and feasible camera locations. The proposed work gives an economical and cost-effective solution for any surveillance application.

Acknowledgements We acknowledge the suggestions and help of Prof. Santanu Chaudhury (IIT Delhi) to this work.

References

1. Astaras, S., et al. (2017). Visual detection of events of interest from urban activity. *Wireless Personal Communications*, 97(2), 1877–1888.
2. Luboby, S. C., et al. (2018). Mesh IP video surveillance systems model design and performance evaluation. *Wireless Personal Communications*, 100(2), 227–240.
3. 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).
4. 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).
5. Rourke, J. O. (1987). *Art gallery theorems and algorithms*. Oxford: Oxford University Press.
6. 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).
7. 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.
8. Mirjalili, S., et al. (2014). A Grey wolf optimizer. *Advances in Engineering Software*, 69, 46–61.
9. Kumar, G., & Ranga, V. (2017). Healing Partitioned WSNs, Ubiquitous Computing and Ambient Intelligence. *Lecture Notes in Computer Science*, 10586, 545–557.
10. Kumar, G., & Ranga, V. (2017). Meta-heuristic solution for relay nodes placement in constrained environment. In *10th International Conference on Contemporary Computing*, IEEE.
11. Saunhita, S., et al. (2018). Optimized relay nodes positioning to achieve full connectivity in WSNs. *Wireless Personal Communications*, 99(4), 1521–1540.
12. Zhao, C., et al. (2007). PSO for optimal deployment of relay nodes in hybrid sensor networks. In *IEEE congress on evolutionary computation*, (pp. 3316–3320).
13. Hashim, H. A., et al. (2016). Optimal placement of relay nodes in WSN using ABC algorithm. *Journal of Network and Computer Applications*, 64, 239–248.
14. Sujitha, J., & Baskaran, K. (2018). Genetic GWO based channel estimation in wireless communication system. *Wireless Personal Communications*, 99(2), 965–984.
15. Mohanty, S., et al. (2016). A new MPPT design using GWO technique for photovoltaic system under partial shading conditions. *IEEE Transactions on Sustainable Energy*, 7(1), 181–188.
16. 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.
17. 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>.
18. Suzuki, I., et al. (2001). Searching a polygonal region from the boundary. *International Journal of Computation Geometry and Applications*, 11(5), 529–553.
19. Bose, P., et al. (1997). The flood light problem. *International Journal of Computation Geometry and Applications*, 7(1), 153–163.
20. Castro, V. E., & Rourke, J. O. (1995). Illumination of polygons with vertex lights. *Information Process Letter*, 56(1), 9–13.
21. Chen, X., & Davis, J. (1999). Camera placement considering occlusion for robust motion capture, *Technical Report CS-TR-2000-07*.
22. Mittal, A., & Davis, L. (2004). Visibility analysis and sensor planning in dynamic environment. In *European conference on computer vision*. Springer, 175–189.
23. Horster, E., & Lienhart, R. (2006). Calibrating and optimizing pose of visual sensors in distributed platforms. *ACM Multimedia Systems Journal*, 12(3), 195–210.
24. Kankanhalli, M. et al. (2006). A design methodology for selection and placement of sensors in multimedia systems. In *Proceedings of VSSN-06*, (pp. 121–130).
25. Astaras, S., et al. (2017). Visual detection of events of interest from urban activity. *Wireless Personal Communications*, 97(3), 1877–1888. <https://doi.org/10.1007/s11277-017-4651-z>.
26. Olague, G., & Dunn, E. (2007). Development of a practical photogrammetric network design using evolutionary computing. *Photogrammetric Record*, 22(117), 22–38.
27. Nam, Y., Hong, S. (2012). Optimal placement of multiple visual sensors using simulation of pedestrian movement. In *International Conference on Computing, Networking And Communications*, (pp. 67–71).
28. Topcuoglu, H., et al. (2011). Positioning and utilizing sensors on a 3d terrain part i-theory and modeling. *IEEE Transactions on Systems Man Cybernetics Part C*, 41(3), 376–382.
29. Mishra, R. et al. (2013). Monitoring a large surveillance space through distributed face matching. In *4th National Conference on Comp. Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, (pp. 1–5).

30. Liu, Junbin, et al. (2014). Optimal camera planning under versatile user constraints in multi-camera image processing systems. *IEEE Transactions on Image Processing*, 23(1), 177–184.
31. Jun-Woo, A., et al. (2016). Two-phase algorithm for optimal camera placement. *Scientific Programming*. <https://doi.org/10.1155/2016/4801784>.
32. Sungbum, J. (2018). Placing visual sensors using heuristic algorithms for bridge surveillance. *Applied Sciences*, 8(1), 70. <https://doi.org/10.3390/app8010070>.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Ajay Kaushik is a senior research fellow (Ph.D. research scholar) in the Department of Computer Science and Engineering at Delhi Technological University (DTU), Delhi, India. His areas of interest are Wireless sensor networks and Nature Inspired Intelligence. He did his M.Tech. (Master of Technology), Post B.Tech. in Computer Science & Engineering from Kurukshetra University, Kurukshetra. He qualified UGC NET-JRF with a percentage of 70.8% in year 2013. He has always been among the top rank-holders of the university in these courses.



Prof. S. Indu is a Professor in the Department of Electronics and communication Engineering, Delhi Technological University, Delhi, and is currently the head of the department there. She has done Ph.D. in Computer Engineering from University of Delhi and M.Tech, Post B.Tech. from University of Kerala, Kerala. She was awarded the Best Branch Counselor award 2013 IEEE USA. Her research interests are in the field of Computer Vision, Image Processing, and wireless sensor network. She has 3 completed sponsored research projects. She published several research papers in international journals and conferences.



Prof. Daya Gupta is a Professor in the Department of Computer Science and Engineering, Delhi Technological University, Delhi. She has done Ph.D. in Computer Engineering from University of Delhi and M.Sc. (Maths), Post M.Sc. Diploma (Computers Sc.) from IIT, Delhi. She is a senior member of IEEE and a life member of CSI. Her research interests are in the field of requirement engineering, network security, ad hoc networks and image classification in remote sensing. She has published several research papers in international journals and conferences and has chaired sessions and delivered invited talks at many national and international conferences.