ITcon
www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

BIM-BASED FRAMEWORK FOR OPTIMIZATION OF CCTV SURVEILLANCE IN BUILDINGS

SUBMITTED: April 2024
REVISED: August 2024
PUBLISHED: December 2024
EDITOR: Esther Obonyo, Robert Amor
DOI: 10.36680/j.itcon.2024.039

Taha Aziz,
National University of Sciences and Technology
ORCID: https://orcid.org/0009-0004-3706-2640
taziz.bece19nice@student.nust.edu.pk

Muhammad Umer Zubair*, Assistant Professor, (corresponding author)
King Faisal University
mzubair@kfu.edu.sa

Muhammad Usman Hassan, Assistant Professor
National University of Sciences and Technology
usman.hassan@nice.nust.edu.pk

Mehmood Ahmed,
National University of Sciences and Technology
mahmed.bece19nice@student.nust.edu.pk

Muhammad Arsalan Khan, Assistant Professor
International Islamic University
arsalan.khan@iiu.edu.pk

Waqas Arshad Tanoli, Assistant Professor
King Faisal University
wtanoli@kfu.edu.sa

SUMMARY: *Surveillance cameras are becoming an integral part of the buildings due to their ability to ensure security, as well as to promote safety and overall well-being. Finding the optimal camera configuration remains a challenge, as current practices depend heavily on professional experience and subjective judgment. These practices have several limitations which can adversely impact camera coverage. Building information modelling (BIM) usage is growing in the industry due to its ability to generate accurate spatial data. Therefore, this study proposes a BIM-based framework to optimize camera placement process using optimization algorithms (OAs). Firstly, the framework extracts spatial data of the target area from the BIM based on user defined requirements. Secondly, it adopts an optimization algorithm to find the optimal camera positions for the target area based on the user requirement. The selection of optimization algorithm was made following a comparative evaluation between Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). Lastly, it helps visualize the optimized results within the building using BIM. The framework was validated on a hospital building, revealing 27% increase in coverage, a significant reduction in overlap, and a lower camera requirement compared to experience-based camera configuration.*

KEYWORDS: *Building information modeling (BIM), CCTV Network, Automation, Optimization of camera placement, Genetic Algorithm (GA), Particle Swarm Optimization (PSO).*

REFERENCE: *Taha Aziz, Muhammad Umer Zubair, Muhammad Usman Hassan, Mehmood Ahmed, Muhammad Arsalan Khan, Waqas Arshad Tanoli (2024). BIM-based framework for optimization of CCTV surveillance in buildings. Journal of Information Technology in Construction (ITcon), Vol. 29, pg. 894-913, DOI: 10.36680/j.itcon.2024.039*

1. INTRODUCTION

Cameras play a crucial role in capturing a diverse range of information in activities such as surveillance, photography, and video recording. Moreover, their affordability and widespread availability make them valuable to the industry (Bourikas et al., 2016), aiding in overseeing equipment and resources without interfering with the ongoing operations (Chen et al., 2021). Specifically, surveillance cameras emerge as the most employed sensory devices, delivering real-time monitoring of the environment. As a result they hold great potential across various applications (Wang et al., 2021). Closed-circuit television (CCTV) cameras are pivotal for ensuring safety in diverse environments. Their wide-spread utilization is due to escalating criminal incidents and not only homes, but businesses, organizations, and public spaces rely on CCTV cameras for continuous surveillance. CCTV cameras help in effectively capturing illicit activities and facilitating ongoing monitoring (Soh and Ahmad, 2019). The continuous monitoring facilitated by cameras aids managers in making informed decisions by capturing crucial data on input resources, including workers' activities This enables real-time assessment of operational efficiency and identification of unsafe behaviors onsite (Kim et al., 2019). Despite their importance, surveillance systems or camera networks are designed manually by experts based on 2D plans. Sole reliance on the design experience of CCTV professionals has many practical issues, such as suboptimal monitoring layouts, extensive blind spots, and low monitoring efficiency (Xie et al., 2023). However, proper planning of a camera system can reduce both installation and operational costs as relocating an existing setup is quite expensive (Altahir et al., 2022). More cameras are provided to compensate for the lack of coverage, thereby resulting in greater overlap region and subsequently, higher cost. Challenges arise in establishing suitable camera configurations due to various conditions inherent in complex job sites. The experience based process of camera placement becomes both time-consuming and labor-intensive with discrepancies emerging between the actual coverage of installed cameras and the expectations of project managers (Kim et al., 2019). Therefore, there is a need to adopt an efficient camera placement method to enhance overall efficiency and reduce costs.

Accurate data extraction is crucial for obtaining reliable results. Due to accuracy and other numerous elements, Building Information Modeling (BIM) stands out as the most important development in the Architecture, Engineering, and Construction (AEC) industries. Using BIM, a digital representation of projects can be created within a virtual environment, including real-time data, information, and accurate geometrical details (Dahmane et al., 2020). Through the incorporation of Application Programming Interfaces (APIs) into BIM, specific information related to construction projects can be extracted, including project attributes, geometry, and spatial data (Guo and Zhang, 2021). BIM tools like Autodesk Revit have revolutionized the industry, with Revit generating substantial amounts of data. Revit also offers a wide range of add-ins and extensions that enhances its functionality (Beorkrem et al.). The usage of BIM in development of camera networks can improve the overall system performance and help identify various elements that surround the coverage area (Albahri and Hammad, 2016). So, an effective camera placement method can be developed using BIM as a valuable tool.

This study aims to develop a BIM-based framework to optimize camera placement process for buildings using optimization algorithms. Optimization of camera coverage is a complex problem requiring complex calculations and extensive iterations. However, this can be done effectively using optimization algorithms. When choosing an optimization algorithm, it is important to take specific objectives and constraints of the problem into account. Both Genetic Algorithm (GA) and Particle Swarm optimization (PSO) are part of evolutionary computation optimization methods and are gaining popularity in research for solving real life optimization problems (Kennedy and Eberhart, 1995). Evolutionary methods aim to regulate the random search process using the rules governing nature. Known for their versatility, these algorithms excel in optimizing complex scenarios with non-differentiable cost functions, making them valuable tools in various applications. GA is a global search technique that is based upon the principle of natural selection and genetics (Hassanat et al., 2019). PSO is another powerful global search technique that uses the principle of swarm intelligence to find optimal solutions or near optimal solutions (Kennedy and Eberhart, 1995). Both algorithms, GA and PSO, are well-suited for diverse range of applications due to their ease of implementation and versatility (Abraham and Jain, 2005, Assareh et al., 2010). These algorithms are similar in the sense that these two techniques are population-based search methods, and search for the optimal solution through successive updates of generations. Both algorithms use different approaches and computational effort to reach an optimized solution, therefore it is appropriate to compare their performance for a given problem (Panda and Padhy, 2008). PSO and GAs are particularly well-suited for single-objective optimization problems (Wang et al., 2022, Zhang et al., 2019), especially when applied to optimizing camera placement for a building, where the objective is to maximize coverage or minimize cost. Therefore, these algorithms are appropriate for the optimization problem in this research. These optimization algorithms have been used in various studies related to building services. Chan (2024) used GA with neighbouring search to optimize district heating system (DHS) piping networks, significantly improving material cost efficiency and flow patterns. Afroz et al. (2022) utilized PSO in HVAC systems to reduce energy consumption while maintaining indoor environmental quality. Yixuan (2022) employed GA for indoor design, optimizing spatial layouts for energy efficiency and structural performance. Cheng et al. (2022) used GA combined with Computational Fluid Dynamics (CFD) to optimize sensor placement for thermal comfort and air quality in multi-zone buildings. Hsiao and Sung (2021) applied PSO to an intelligent LED lighting system, optimizing electricity consumption through real-time data analysis. Allam et al. (2023) compared GA and PSO for optimizing daylighting and energy performance in office buildings, finding that both algorithms significantly reduced the computational effort required for effective optimization. These studies highlight that a significant number of studies in building management have relied on GA and PSO for decision-making.

In this research, the proposed framework is designed to generate optimized outcomes using BIM. Initially, data is extracted from the BIM of the building. Subsequently, the optimization algorithm utilizes this extracted data to produce the desired results. Finally, these results are visualized within the BIM environment. So, this study makes the following contributions: 1) Development of a practical framework applicable in real-world scenarios. 2) Comparative analysis between the performance of GA and PSO within the context of camera placement 3) The framework, designed to enhance coverage in the target region by increasing overall coverage percentage and minimizing camera overlap.

2. RELATED WORKS

Several studies have emerged in literature, focusing on optimizing coverage and implementing automatic camera placement strategies. The oldest coverage optimization is the Art Gallery problem (APG) (Chvátal, 1975). It involves measuring the position and minimum number of guards required to fully supervise important areas in a polygon. However, guards do not have any view restrictions as compared to the restricted field of view (FOV) of cameras so solutions of AGP cannot be directly applied to camera networks. To address this problem, Horster and Lienhart (2006) proposed a liner programming model. They assumed the polygon to be a simple rectangle and did not consider the presence of any obstructions or constraints inside the polygon. Moreover, Kenichi and Hitoshi (2008) proposed a segmented approach for the automatic placement of cameras. They suggested that for effective monitoring only selective areas need to be observed through cameras, they named those selective areas as "essential regions". Their method involved dividing the polygon into segments of rectangle allowing for coverage in irregularly shaped polygons, which can result in inaccurate calculations as small shapes behind obstacles might be missed during the camera coverage process. Xu and Lei (2009) optimized camera layout field of view using Particle Swarm Optimization. Fixed cameras were initially distributed, with positions held constantly. As a result, the optimization process was considered solely on enhancing the orientation of these cameras. Their work was extended in (Xu et al., 2010) by incorporating the camera positions in the optimization process. The researchers assumed mobile cameras with movable capabilities. They applied three operators (absorbing, penalty, and reflection) to optimize particle movement, considering defined constraints. They considered rectangular surveillance areas and assumed cameras of same specifications i.e., fixed FOV and range. Albahri and Hammad (2016) presented a simulation-based optimization method for the placement of a single fixed camera inside a BIM model. However, this approach may not be suitable for installing a large number of cameras simultaneously due to higher computational time. Chen et al. (2021) adopted a 2D circle FOV and grid mapping for coverage analysis. However, their study was limited to construction site and can't be directly applied to building environments. Murray et al. (2007) and Kim et al. (2008) optimized surveillance coverage in specific urban locations, considering cost constraints. Though effective, their approach had limitations. Firstly, it was tailored for external urban spaces as it utilized a 2D GIS grid impacting visibility accuracy, and secondly, it didn't consider the variations in importance levels of different areas. Nam and Hong (2014) optimized camera placement indoors, considering coverage and camera cost. However, the study lacked a clear segmentation method for the monitored area, impacting the identification of areas of interest. Their approach utilized a path-finding algorithm and failed to account for restrictions, such as privacy concerns, in certain areas that prevent the use of surveillance cameras. Han et al. (2019) optimized and analyzed camera coverage by Maximal Coverage Location Problem with Complementary Coverage (MCLP-CC) model. They accounted for the presence of Regions of Interest and obstructions. However, their approach relied on cameras with similar specifications. Range of camera plays an important part in monitoring operations as distant objects can't be identified clearly therefore, in (Albahri Ameen and Hammad, 2017, Zhang et al., 2019) the concept of visibility quality was adopted by setting a maximum camera range in their studies.

Most of the studies fail to consider both interested regions and camera customization. Among those that incorporate both aspects, they often omit the visibility range of cameras from their framework. Furthermore, the majority of these studies fail to account for building constraints, making their implementation on any type of building nearly impossible. Hence, there is a need for a single BIM-based framework that optimizes and automates the placement of camera network layout while considering relevant constraints along with providing visualization of cameras view before installation. Another limitation in most of the research is that many studies use either GA or PSO. None have compared their performance together in the context of camera placement. A comparative analysis between genetic algorithms and PSO would contribute valuable insights into the strengths and weaknesses of each approach, enabling a more informed choice of optimization algorithms in future studies. This proposed framework uses a 2D representation of the camera model due to its simplicity (Altahir et al., 2017). This representation is particularly relevant for cases such as unfurnished houses and large public areas of the buildings (Syu and Peng, 2022). In 2D, the data is presented in the form of grid points, whereas in 3D implementation the cubes are present, which increases complexity and computational demands (Altahir et al., 2017). The higher computational cost associated with 3D modelling makes installing a large number of cameras in larger spaces impractical. Therefore, the 2D approach offers a more feasible solution for optimizing camera placement in various scenarios.

3. METHODOLOGY

This research proposes a novel approach to optimize placement of camera network in buildings. The proposed framework as illustrated in Figure 1, has a systematic approach consisting of three key components i.e. Data extraction, Optimization framework and Visualization module. In data extraction module, information is extracted from the BIM model after which it is filtered to prepare the relevant data for optimization. Next, optimization framework is implemented on the relevant data. Implementation uses an OA to obtain optimized placements along with their respective coverage and overlap. Lastly, optimal placements are visualized back in the BIM Model of the building. Each module of the proposed framework is described in detail in the following sub-sections.

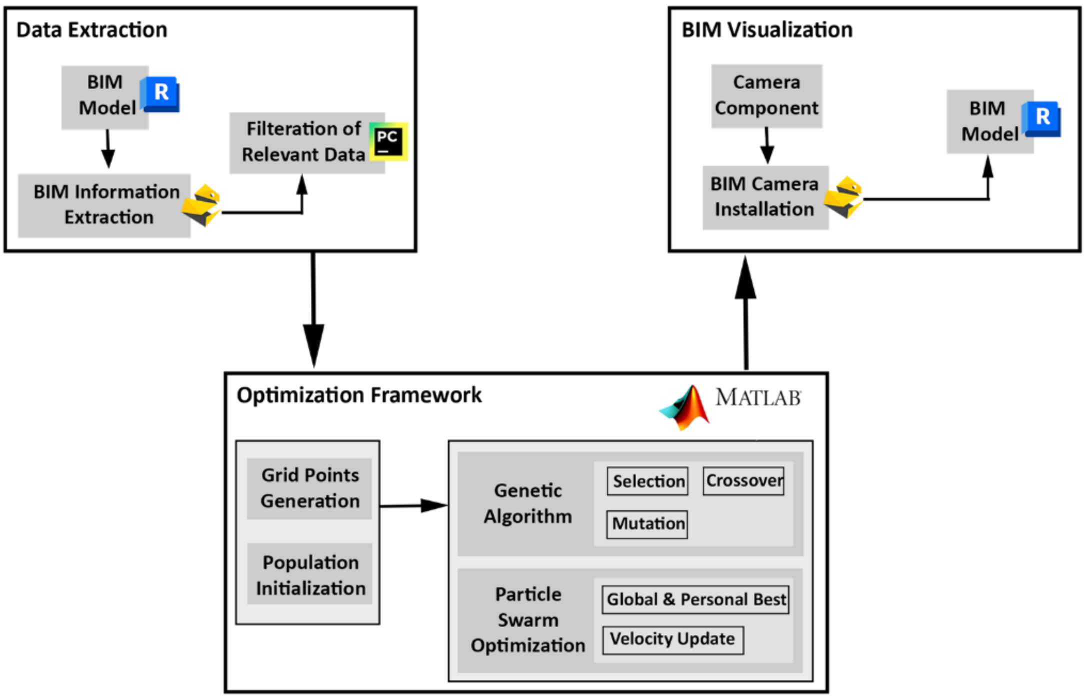

Figure 1: Flowchart of Research Methodology.

3.1 Data Extraction

The initial step in the process involves creating a BIM model using Autodesk Revit. Once the BIM model is created, Revit add-ins like pyRevit can be used to filter all the information embedded in the BIM model and extract the useful information (Guo and Zhang, 2021). Using pyRevit, data including geometrical coordinates of the rooms and doors with their respective tag and level will be extracted from BIM. The locations of the doors are important for determining the Region of Interest (ROI) and Forbidden Region (FR) in each room. The ROI includes specific points and areas that require constant surveillance, whereas FR are those points whose surveillance needs to be avoided at all costs to prevent invasion of privacy. The pyRevit interacts with Revit API to extract geometrical data i.e. coordinates, as shown in Figure 2. The data is exported in an organized format which allows for easy manipulation of the collected information. The extracted data, including room and door coordinates, is exported to a spreadsheet for further analysis.

The exported spreadsheet contains coordinates of all the rooms and doors; Therefore, the exported spreadsheet is filtered to only have the necessary data. A python script used in PyCharm IDE asks to define the level (e.g., Ground Level, First Floor etc.) at which cameras are to be installed. All room names belonging to that level are displayed and the script asks to enter room where surveillance is required. However, it is important to note that certain case-specific constraints are implemented to ensure only appropriate rooms are selected. If a room name that is within these constraints (e.g., restrooms, private chambers) is entered, a message appears indicating room is prohibited. Once a valid room is selected, the script allots ROIs and FRs by examining which door names contain "FR" and "ROI" in the spreadsheet. All rooms, except for the one entered by the user, are considered obstacles as they have the potential to obstruct the camera's coverage. On completion, the relevant information is exported and stored into a spreadsheet, ready to be used for optimization. The algorithm ensures that all data regarding the rooms and doors is extracted before the optimization process begins. However, only one room can be selected for optimizing camera positions at a time, with doors automatically chosen based on their ROI and FR tags.

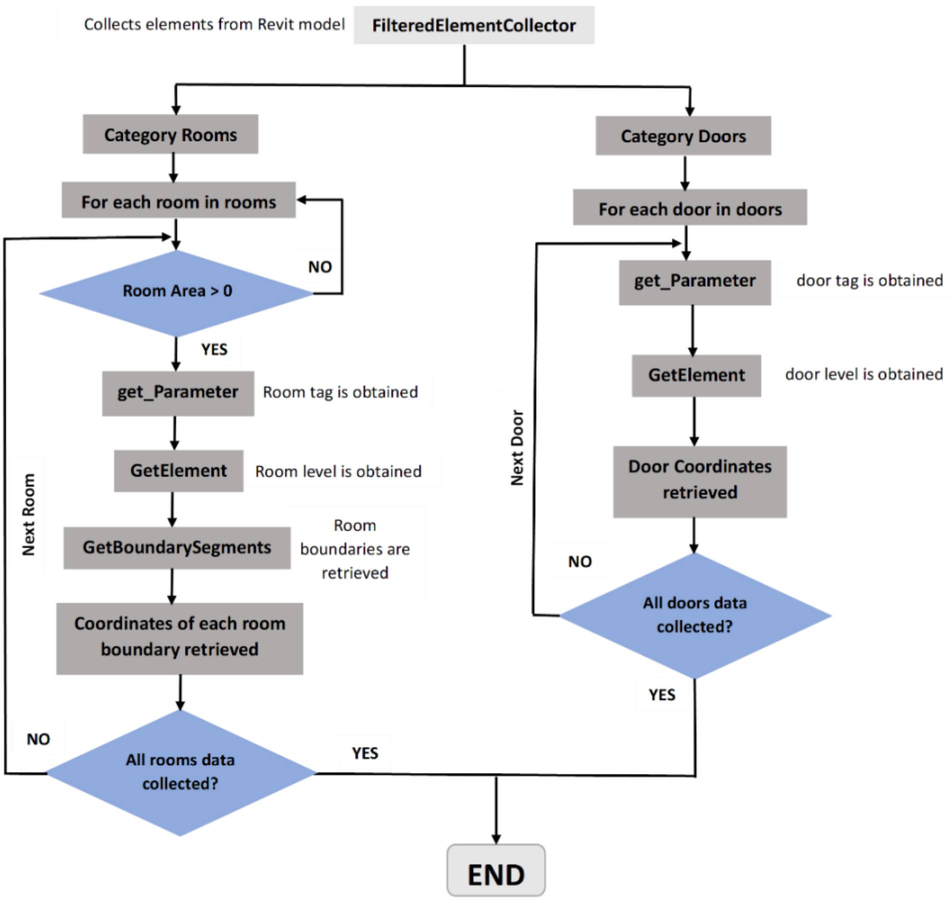

Figure 2: Flowchart of Data Extraction from BIM.

3.2 Optimization Framework

It consists of three major components: optimization parameters, fitness function, optimization algorithms i.e. genetic algorithm and particle swarm optimization. OAs are implemented in MATLAB environment. There are generally two objectives of the camera optimization process i.e. maximize the coverage or minimize the camera installation cost. Maximizing the coverage involves getting the maximum possible surveillance with fixed number of cameras as done by (Albahri and Hammad, 2016). The OA will proceed to find optimized positions for the cameras to maximize their coverage. It will aim to cover as much of the room as possible while considering the given FOV of each camera. In minimizing cost (Si et al., 2017), OA minimizes the number of cameras using desired camera coverage. Desired coverage percentage is the percentage of room required to be covered. The OA starts with a single camera and continues to iterate through cameras until the desired coverage percentage is achieved. The purpose of this scenario is to minimize the number of cameras needed while still meeting the coverage requirement. This optimization ensures that the coverage percentage is fixed, and minimum number of cameras are used to achieve the coverage percentage, thereby minimizing the cost. In both cases, the algorithm tries to minimize the overlap between cameras to achieve better coverage of the target area (Suresh et al., 2020, Wang et al., 2019). Reducing camera overlap improves overall area coverage, but it doesn't imply complete elimination of overlap. The algorithm converges when the highest fitness value remains constant and shows no improvement for over 50 iterations.

3.2.1 Optimization Parameters

The variables used for optimization include the position and orientation of camera (x, y represents the position of camera) and theta represents the orientation of camera. The height of camera (Z) is assumed to be at maximum ceiling height (Albahri and Hammad, 2016) and will not be used in the optimization of camera placement. Limiting the optimization process to 2D allows for faster computation and easier implementation. In addition, most of the coverage models use 2D rather than 3D space due to the model simplicity (Horster and Lienhart, 2006). This optimization process uses the 'floor-to-ceiling walls' model (Chen et al., 2021), where the obstacle wall completely obstructs the camera's field of view. Camera orientation is relative to the x-axis inside a cartesian plane.

Before optimizing camera placement, a search space must be defined. In our proposed optimization, the search space is defined by a set of grid points that are uniformly generated within the boundaries of the room. The number of grid points along x and y axes affects the accuracy of the solution. A higher number corresponds to better optimization results and longer computational time. Grid points are generated based on the minimum and maximum x, y values of the room and it is unavoidable that some points are generated outside the defined boundaries of the room. To address this issue and ensure precise camera placement, a robust algorithm known as 'Point in Polygon' is utilized. Point in Polygon algorithm (Taylor, 1994) analyzes the geometric relationship between the point and the polygon's edges, counting the number of intersections between them. This algorithm efficiently identifies and removes cameras that lie outside the designated room boundary, thereby optimizing the accuracy and integrity of the grid point generation process (Alciatore and Miranda, 1995). Grid points inside the obstacles are again removed using the point in polygon algorithm. This algorithm will again be utilized multiple times.

Optimization process starts with generating the initial population/swarm, at random positions ranging between minimum and maximum values of x and y coordinates of the room and orientations ranging -180 to 180 degrees, with respect to x-axis. Point in polygon algorithm is utilized to avoid generation of cameras inside the obstacle and outside the room boundaries. Figure 3 shows a typical population/swarm sample. where X and Y are the coordinates of camera position, θ is the orientation of the camera and n is total number of cameras (cameras to be installed inside an area).

X_1	Y_1	θ_1	X_2	Y_2	θ_2	X_3	Y_3	θ_3	....	X_n	Y_n	θ_n

Figure 3: Single unit of the population.

3.2.2 Fitness Function

Fitness function has multiple variables to measure, serving as primary means of evaluating how well a potential solution performs (Hassanat et al., 2019). In this research, the fitness function is designed to measure how well a solution covers grid points. If a solution covers more grid points, it gets a higher fitness score. However, the fitness score also considers some other factors in its calculation like penalty (David et al., 2007). Following fitness function is used to assess the performance of each individual in the OAs:

$$fitness\ (individial) = \frac{coverage + roi_{coverage} - fr_{coverage} - overlap\ penalty * overlap}{total\ grid\ points\ within\ a\ room} \quad (1)$$

where Individual = list of tuples, each tuple representing a camera position and orientation i.e. x, y and theta, Coverage = total number of grid points within the room covered by the cameras , $\mathrm{roi_{coverage}}$ = total number of grid points within the regions of interest covered by the cameras, $\mathrm{fr_{coverage}}$= total number of grid points within the forbidden region covered by the cameras, overlap= total number of grid points within the room covered by more than one camera, overlap penalty= constant value that penalizes overlapping grid points between cameras, Total grid points within room=total number of grid points within the room, excluding any points inside the obstacles.

Grid points are included in camera coverage if they pass the following criteria:

$$\text{Distance} < \text{Range}$$
$$Distance = \sqrt{dx^2 - dy^2} \quad (2)$$

Distance between camera (C) and a grid point (G) is calculated using Eq. (2). As shown in Figure 4, a camera can only cover a grid point if it is within its specified range (R).

$$\frac{-\text{CAMERA FOV}(\boldsymbol{\alpha})}{2} < \text{Angle}(\Phi) < \frac{\text{CAMERA FOV}(\boldsymbol{\alpha})}{2} \quad (3)$$

$$Angle(\Phi) = \tan^{-1}\frac{dy}{dx} - \theta \quad (4)$$

Calculation of the difference in the angle of line segment (camera, grid point) and orientation of camera using Eq. (3). As shown in Figure 4, a grid point only falls within a camera's field of view if it satisfies the above condition.

Grid point not blocked by an obstacle

Intersection between Line Segment: L = (Camera, Grid point) and obstacles is checked. No intersection between these two means there are no obstacles blocking the grid point.

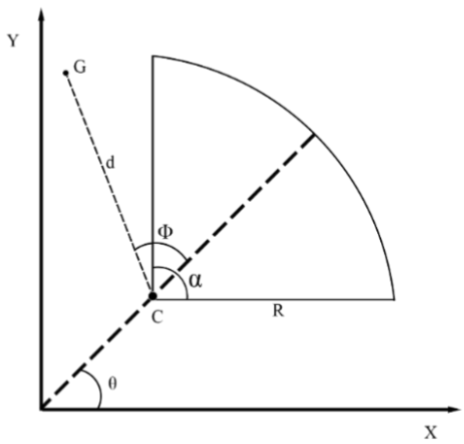

Figure 4: Visual representation of camera field of view in the search space.

This process is repeated for each individual grid point and coverage of the camera is calculated. Similarly, the region of interest and overlap coverage are calculated. The same criteria are used to check grid points within the ROIs and when it comes to overlap, same criteria are used to check if a grid point is covered by multiple cameras. Grid points covered by multiple cameras are included in the overlap coverage.

3.2.3 Genetic Algorithm

GA starts with generating an initial set of solutions called population. Individuals inside the population are called chromosomes and a single unit inside the chromosome is labeled as a gene. A fitness function is used to calculate chromosome's fitness values and evaluation is performed based on this fitness value (Li et al., 1998). Once population has been initialized, genetic operations like selection, elitism, crossover, and mutation are iterated on this population to converge the solution to an optimized placement. Certain parameters like number of generations, population size, mutation rate and elitism rate are selected after experimentation. This study makes use of the integer representation of GA as integers have the ability to represent values with higher precision than binary representations (Deep et al., 2009). Elitism is a technique used to make sure that the top-performing individuals (chromosomes) are directly carried over to the next generation without any changes, ensuring that their superior quality is maintained in the population. This helps the genetic algorithm hold onto its best solutions and build upon them in subsequent generations. (Vasconcelos et al., 2001)

Several selection strategies have been developed and utilized in genetic algorithm optimization. They are categorized into four different groups. (Shukla et al., 2015) Proportionate Roulette Wheel Selection operates by favoring individuals with higher fitness values. It's like spinning a roulette wheel where the size of each individual slice of the wheel is proportional to their fitness. Individuals with higher fitness have larger slices and are more likely to be selected for reproduction. Linear Ranking Selection involves ranking individuals by their fitness, and selection probabilities are assigned based on their rank rather than their actual fitness values. This helps maintain diversity and reduce the influence of outliers. Exponential Ranking Selection differs from linear ranking selection due to the use of exponentially weighted probabilities. The algorithm for both Linear Ranking and Exponential Ranking is identical, with the sole difference lying in how selection probabilities are computed. Tournament Selection involves choosing "n" number of individuals from the population. These individuals compete, and the one with the highest fitness within the tournament is selected for the pool. The number of individuals taking part in a tournament is referred to as tournament size. This process is repeated to fill the mating pool for the next generation. In this research, tournament selection is used due to its efficiency and ease of implementation (Shukla et al., 2015). Selection calculates each individual fitness score using the fitness function and selects parents for the crossover function. For the camera optimization problem addressed in this study, tournament selection operates as illustrated in Figure 5(a).

In crossover, two chromosomes are selected using the selection function and they are used as parents for breeding two new children to populate the next generation. This is repeated until the new population reaches the desired size, typically matching the old population. A single-point crossover technique is applied to the selected population. This process involves choosing two parent chromosomes and exchanging them at a randomly selected single point along their sequences (Hassanat et al., 2019). By doing so, new offspring, known as children, are generated (as shown in Figure 5(b). The crossover point for parents is calculated as following:

$$crossover\ point = random(1, \{duration\ of\ parent\ 1\} - 1) \tag{5}$$

Once the crossover point has been calculated, both parents are exchanged along this point as shown:

$$child\ 1 = parent1[:crossover\ point] + parent2[crossover\ point:] \tag{6}$$

$$child2 = parent2[:crossover\ point] + parent1[crossover\ point:] \tag{7}$$

Children generated in crossover as per Eq. (6) and Eq. (7) are then passed on for mutation. Mutation is used to introduce small random changes into chromosomes of the population. It helps maintain genetic diversity and introduces unique genetic material into the population by bringing random changes into camera's position and orientation, potentially leading to better solutions (Hassanat et al., 2019). (Figure 5(c) illustrates the mutation process). A mutation rate was set, which is a parameter that determines how often mutation occurs and it is in the range of 0 to 1 (Hassanat et al., 2019). Point in polygon utilized here to ensure changes in position don't place the camera outside the room or inside an obstacle.

$$\begin{gathered} If\ random(0,1) < mutation\ rate \\ x' = x + \mathrm{random}(-\mathrm{k}, \mathrm{k}) \\ y' = y + \mathrm{random}(-\mathrm{k}, \mathrm{k}) \\ \theta' = \theta + \mathrm{random}(-\mathrm{j}, \mathrm{j}) \end{gathered} \tag{8}$$

where x and y are the coordinates of camera, θ is the orientation of camera and k,j are the constant values governing how much change is done to the camera position.

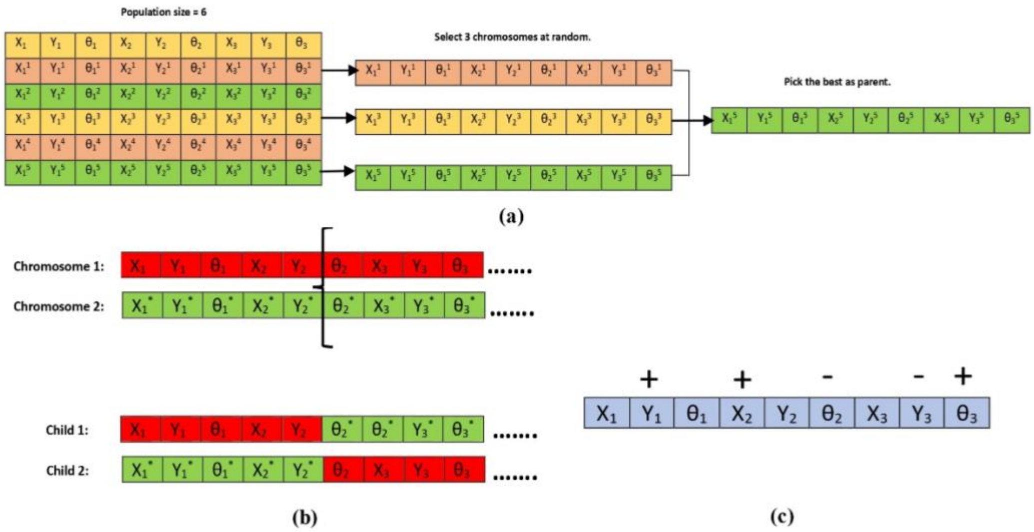

Figure 5: Genetic Algorithm operations in camera placement: (a) Tournament selection (b) Crossover and (c) Mutation.

So, genetic algorithm executes in following steps. Firstly, the population of chromosomes is initialized to represent potential solutions, and a fitness function is employed to evaluate the fitness of each chromosome. Following this, the top-performing chromosomes, also known as elites, are preserved for the subsequent generation. Afterward, chromosomes selected through tournament selection undergo crossover and mutation processes. The next generation's population is formed by merging the elite individuals with the outcomes from the previous step. This entire iterative process is repeated for a predetermined number of cycles or until the solutions converge to an optimal outcome.

3.2.4 Particle Swarm Optimization

PSO is a population-based search algorithm which is inspired by the natural flocking and swarming of the birds and insects (Kennedy and Eberhart, 1995). It is initialized with a set of random solutions and through successive iterations, these individuals refine their fitness value by improving their positions within the search space. Just like with genetic algorithm, solutions in PSO are evaluated based on their fitness value calculated by a fitness function. Each particle in PSO stores two key information: the best position it has attained so far, which corresponds to the highest fitness value achieved by the particle, called its personal best position and the best position with highest fitness encountered by the entire swarm/population, also called the global best position. These two factors help guide the swarm particles to the most optimal solution within the search space (Wang et al., 2018).

Each particle of swarm has an individual velocity, according to which it adjusts its position. This velocity is influenced by both the best-known position (p_{best}) and the global best-known position among all particles (g_{best}). In each iteration, the velocity of the particle is updated according to the following equation:

$$\mathrm{v}_i = \omega \mathrm{v}_i + \mathrm{c}_1 \phi^{\mathrm{t}}{}_1 \left(\mathrm{p}^{\mathrm{t}}{}_{best} - \mathrm{x}^{\mathrm{t}}{}_i\right) + \mathrm{c}_2 \phi^{\mathrm{t}}{}_2 \left(\mathrm{g}^{\mathrm{t}}_{best} - \mathrm{x}^{\mathrm{t}}{}_i\right) \tag{9}$$

Where t is the iteration number, x_i is the particle position, ω is the inertia weight allowing the particle to maintain its current velocity to some extent (Shi and Eberhart, 1998), ϕ_1 and ϕ_2 are randomly generated weights. c_1 and c_2 are constant parameters known as acceleration coefficients, which are positive values determining the maximum step size a particle can undertake. At each iteration, the particle position is updated by adding the velocity factor to the particle position.

$$\mathrm{x^t}_i = \mathrm{x^t}_i + \mathrm{v}_i \tag{10}$$

A higher inertia weight tends to favor global exploration in the search space, whereas a lower inertia weight leans towards refining existing solutions, emphasizing exploitation. Optimal choices for the inertia weight and acceleration coefficients (c1 and c2) are crucial for maintaining a balance between exploration and exploitation.

So, the PSO algorithm executes through the following steps. Firstly, initialize the population of particles, including their velocities, and assess the fitness of the particles using a fitness function. Afterwards, update the personal and global best positions of the particles based on their fitness values, comparing the particle's best position with its current position as indicated by Eq. (11). If the current generation exhibits a superior fitness value than the current global best position, the global best is updated. Following this, velocities are updated by considering both personal and global best-known positions and adjust particle positions accordingly using the updated velocities. Lastly, repeat the entire iterative process for a predetermined number of cycles or until the solutions converge to an optimal outcome.

$$\mathrm{p^t}_{best} = \begin{cases} \mathrm{x^t}_i, if\ fitness(\mathrm{x^t}_i) > fitness(\mathrm{p^t}_{best}) \\ \quad \mathrm{p^t}_{best}, \quad otherwise \end{cases} \tag{11}$$

3.3 BIM Visualization

One of the strengths of BIM is its capability to store, combine, and display both meaningful and relational data alongside geometric information (McArthur et al., 2018). Autodesk Revit add-in, pyRevit is utilized again here to import the results back into the BIM. Before running the script, a camera family needs to be loaded into the project and such family is loaded from a free open-source BIM website. Once the family component is loaded into the project, script adds the cameras at position and orientation from the output spreadsheet of GA. Camera component is added on the exact coordinates and then rotated using the orientation from the spreadsheet file (output of GA). The height of the camera can be changed if needed to make sure its position is at the desired ceiling level.

4. CASE STUDY

To validate the proposed framework of optimized camera placement and evaluate performance of genetic algorithm and particle swarm optimization, a case study was conducted on a real hospital. Surveillance system in healthcare facilities has grown increasingly necessary in recent years. Crimes against people and property occur a lot in hospitals. For example, patients are abused, infants are abducted from nurseries, patient's valuables are stolen, drugs and medical supplies are stolen, lost or misused, equipment and facilities are vandalized; and violent behavior against hospital staff occurs in emergency rooms (Goldstein, 2021). This requires a well-established surveillance system covering all essential regions for the safety and wellbeing of patients, staff, and visitors.

Validation experiments were carried out on Ground floor lobby of the hospital. This area was selected based on its complex and irregular shape, presenting a unique set of challenges. It has several corridors and comprises of both narrow and wide spaces. Figure 6 shows the BIM model of the hospital. Figure 7 shows the ground floor with red indicating the restricted areas. There are some general constraints lying in the coverage area like restrooms and changing rooms. Also, in healthcare facilities there are certain areas like examination rooms and doctor chambers where camera coverage may breach privacy. These specific areas were termed as case-specific constraints and cameras will not be installed in these rooms. Figure 8 shows the target coverage area of the ground floor. A Hikvision (5MP) Camera has been selected for use in this case study, with its specifications detailed in Table 1.

Table 1: Camera Specifications.

Camera Model	Resolution	FOV (°)	Focal length (mm)	Detection Range (ft)	Recognition Range (ft)
DS-2CD2655FWD-IZS	2944 × 1656	88	2.8	164	40

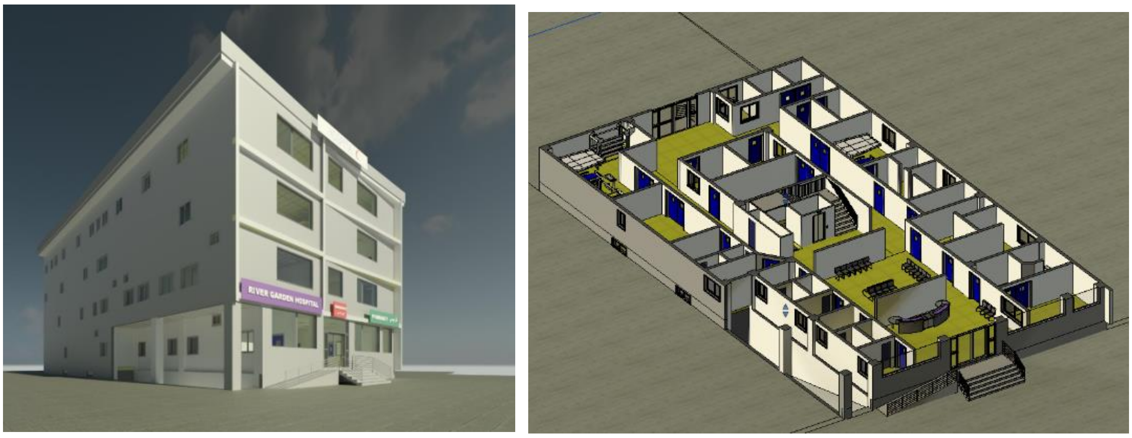

Figure 6: BIM Model of River Garden Hospital.

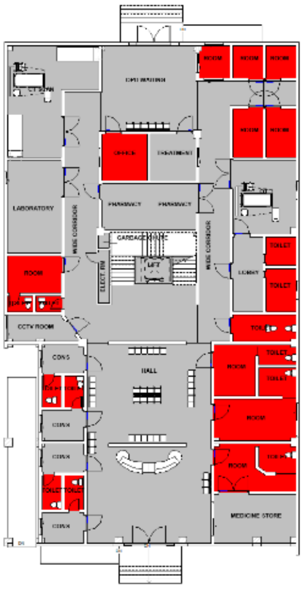

Figure 7: Hospital ground floor with grey showing the allowed regions and red showing the restricted regions.

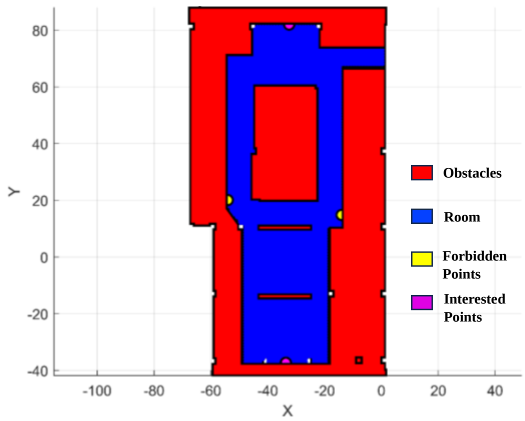

Figure 8: Target region of ground floor with obstacles, interested and forbidden regions.

4.1 Parameters for GA and PSO

Parameters are the settings that influence the behavior of the algorithm and the quality of the solution it produces. These parameters can significantly impact the performance and efficiency of an optimization process. Parameters which lead to reduced computational time and more optimized results are the most desirable. For GAs, computational speed is a major concern (Li et al., 1998). Meanwhile, PSO is prone to converging to local maxima instead of global maxima (Schmitt and Wanka, 2015). Exploration and exploitation are important aspects of optimization algorithms and finding the balance between them is essential in generating optimal results. Exploration means finding new and maybe better solutions, while exploitation focuses on refining and improving known solutions (Črepinšek et al., 2013).

To fine-tune the parameters of GA and PSO, a series of experiments were conducted on the Ground floor of the hospital by installing six cameras. The number of cameras was recommended by a CCTV professional for the ground floor and was considered sufficient to test various aspects of the algorithms' functions, such as coverage overlap, ROIs and blind spots, ensuring a realistic and effective evaluation of their performance. Three different settings as shown in Table 2 and Table 3 were tested for each algorithm. Population and swarm sizes too low may lead to a reduced search space, whereas sizes that are too large increase computational demands (Hassanat et al., 2019). The range of 40 to 80 for population and swarm size was selected because sizes under 40 led to reduced search space, while over 80 increased computational demands. Similarly, a mutation rate that is too low preserves existing solutions but may limit exploration, while a rate that is too high promotes exploration but can cause instability (Hassanat et al., 2019). To balance stability and exploration, a mutation rate range of 0.3 to 0.5 was chosen. An elitism rate of 0.025 was selected to avoid reducing the search space with the smaller population of 40. In contrast, a larger population of 80 allowed for more elites to be retained. The inertia range of 0.4 to 0.8 was found to provide a balance between exploration and exploitation by adjusting the influence of previous velocities (Li et al., 2019). The cognition range of 1.4 to 1.7 determines local exploitation, with smaller values reducing it and larger values increasing it. The social behavior range of 1.4 to 1.7 was selected to balance global exploration, with smaller values providing less social learning and larger values increasing it (Li et al., 2019). Figure 9 presents the fitness graph for each algorithm with their different parameters. GA #3 and PSO #2 achieved higher fitness values during testing therefore, these parameters will be used for the validation phase.

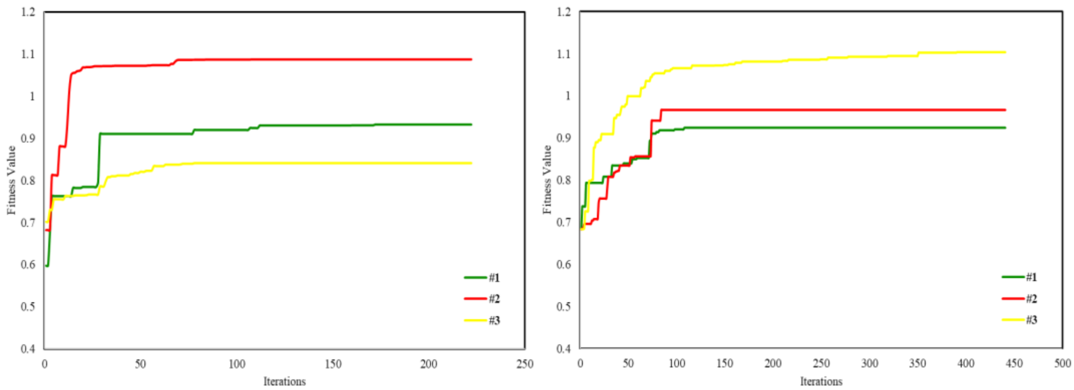

Figure 9: Fitness graphs for PSO parameters (left) and GA parameters (right).

Table 2: Testing parameters for Genetic Algorithm (GA).

GA settings	Population size	Mutation rate	Elitism rate
#1	40	0.4	0.025
#2	60	0.3	0.033
#3	80	0.5	0.05

Table 3: Testing parameters for Particle Swarm Optimization (PSO).

PSO settings	Swarm size	Inertia (ω)	Cognition (c_1)	Social behavior (c_2)
#1	40	0.8	1.7	1.7
#2	60	0.5	1.5	1.5
#3	80	0.4	1.4	1.4

4.1.1 Comparative analysis

To evaluate the performance of OAs with respect to optimal camera placement problem, a comparison is conducted between GA and PSO. Figure 10 and Table 4 shows the comparison between GA and PSO, when installing six cameras on the Ground floor of the hospital. According to the results, PSO converges at 108^{th} iteration, significantly faster than GA which converges at the 391^{st} iteration. GA achieves a better fitness value than the PSO, indicating it can provide a camera layout with a broader coverage area and lesser overlap between the cameras. Our primary goal in camera optimization is to get better coverage than manual placement. For this purpose, GA is more suited to this camera placement problem.

After selecting the better performing algorithm for validating our proposed framework, a CCTV professional with more than 11 years of industrial experience was contacted to formulate a camera layout design. This expert crafted a manual camera layout using both 2D floor plans and a 3D model of the building, ensuring a comprehensive and reliable benchmark for our comparative analysis. The professional installed six cameras on the floor, camera layout was created using the proposed framework for the same number of cameras. The manual camera layout and layout plan provided by genetic algorithm will be compared based on coverage percentage, overlap percentage and coverage of interested and forbidden points. Figure 11(a) shows the manual layout of camera. The manual layout failed to include one of the hospital entrances and missed covering some regions of the corridor. Figure 11(b) shows the layout generated by genetic algorithm. This layout covers all crucial points and provides better overall coverage by covering as much of the area as possible. Table 5 shows the comparison between both layouts. The proposed layout has 40% more coverage and 56.9% lesser overlap, proving that the manual method is much more likely to give suboptimal coverage and miss important regions.

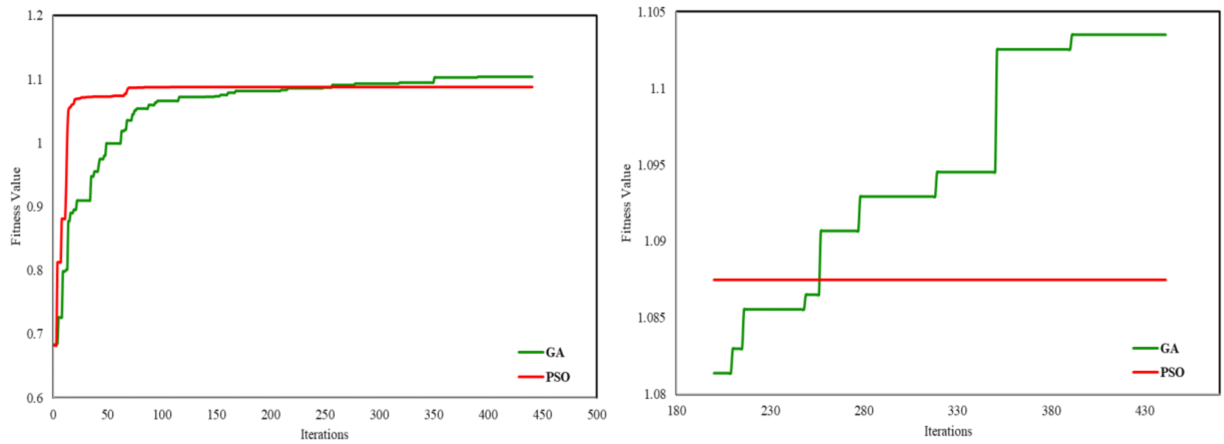

Figure 10: Fitness curve for comparison between GA and PSO.

Table 4: Genetic Algorithm (GA) vs Particle Swarm Optimization (PSO), Search space= 3121 grid points.

	Computation Speed (s/iteration)	Total iterations	Total Computation Time (s)	Coverage Value (%)
Genetic Algorithm	19.4	441	8555.4	88.50
Particle Swarm Optimization	10.3	158	1627.4	85.65

Next, the number of cameras was taken as the subject of optimization keeping the rest of the parameters same, to get the desired coverage with minimum number of cameras. Our target coverage was set at 63.18%, aligning with the coverage from manual placement. Results produced through this optimization process are shown in Figure 12. Five number of cameras were used to provide 80.15% coverage with 0.15% overlap region. The percentage coverage was increased by 26.88% and overlap was decreased by 98.77% while using one fewer camera when compared to manually designed layout. All regions of interest were also included in the coverage. Table 6 shows comparison between both layouts. This demonstrates that an optimized algorithm-based layout can achieve equivalent results as a manually designed layout but with reduced number of cameras. The result of this optimization process is stored in a spreadsheet. By maximizing camera coverage using the framework, blind spots were concurrently minimized, ensuring a more thorough surveillance approach. The 2D representation of uniform grid points in coverage is considered fairly accurate (Altahir et al., 2017). While the 3D models provide more accurate representation, the computational power required for large spaces, such as the ground floor in our case with six cameras, can make the whole framework quite inefficient and time-consuming.

After the completion of the optimization process, the resulting optimized cameras are visualized in Autodesk Revit through the utilization of pyRevit plugin. The entity used to visualize camera placement is shown in Figure 13(left). Upon clicking the “import cameras” pushbutton in pyRevit Tab, camera positions are imported from the excel file and the camera entity is added at those imported coordinates. Figure 13(right) shows the ground floor with cameras installed and Figure 14 displays the installation of one of the cameras within the BIM model. The proposed framework was validated on a desktop with the specifications of AMD Ryzen 5 2600X 3.6GHz CPU and 16GB RAM.

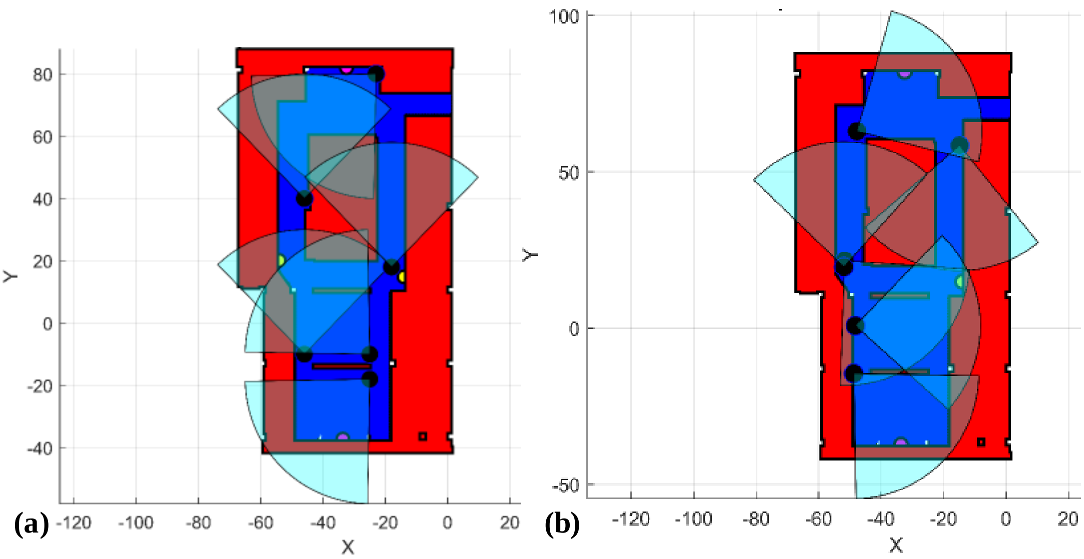

Figure 11: (a) Manual placement by the professional and (b) Optimized positions by Genetic Algorithm.

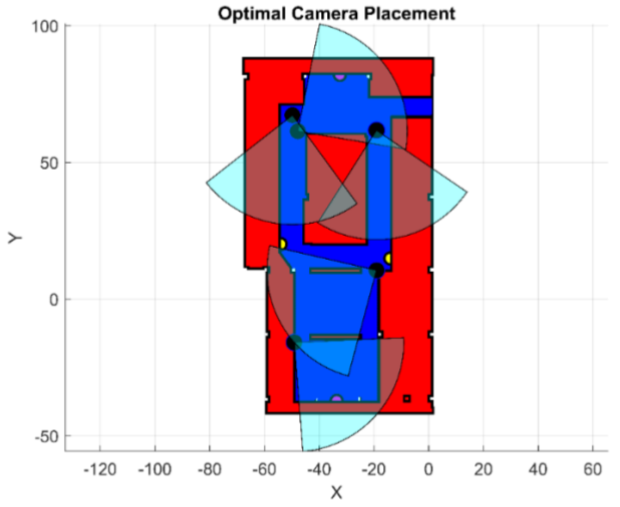

Figure 12: Desired coverage optimized camera positions.

Table 5: Comparison of manual placement and genetic algorithm optimization results.

PSO settings	**Camera locations**	**IR (%)**	**FR (%)**	**Total Coverage (%)**	**Overlap (%)**
Manual experience based	(-23, 80, -135), (-46, 40, 90), (-46, -10, 90), (-25, -18, -135), (-25, -10, 135), (-18, 18, 90)	50	0	63.18	12.43
Genetic Algorithm	(-14.9064, 58.4541, -95), (-51.6631, 21.6266, -48), (-51.9411, 19.5874, 92), (-48.2509, 0.783875, 2), (-48.7693, -14.5353, -45), (-47.7202, 62.9747, 30)	100	0	88.50	5.35

Table 6: Comparison between experience and genetic based layout.

	Number of cameras	**IR (%)**	**FR (%)**	**Total Coverage (%)**	**Overlap (%)**
Manual experience based	6	50	0	63.18	12.43
Genetic Algorithm (Option B)	5	100	0	80.15	0.15

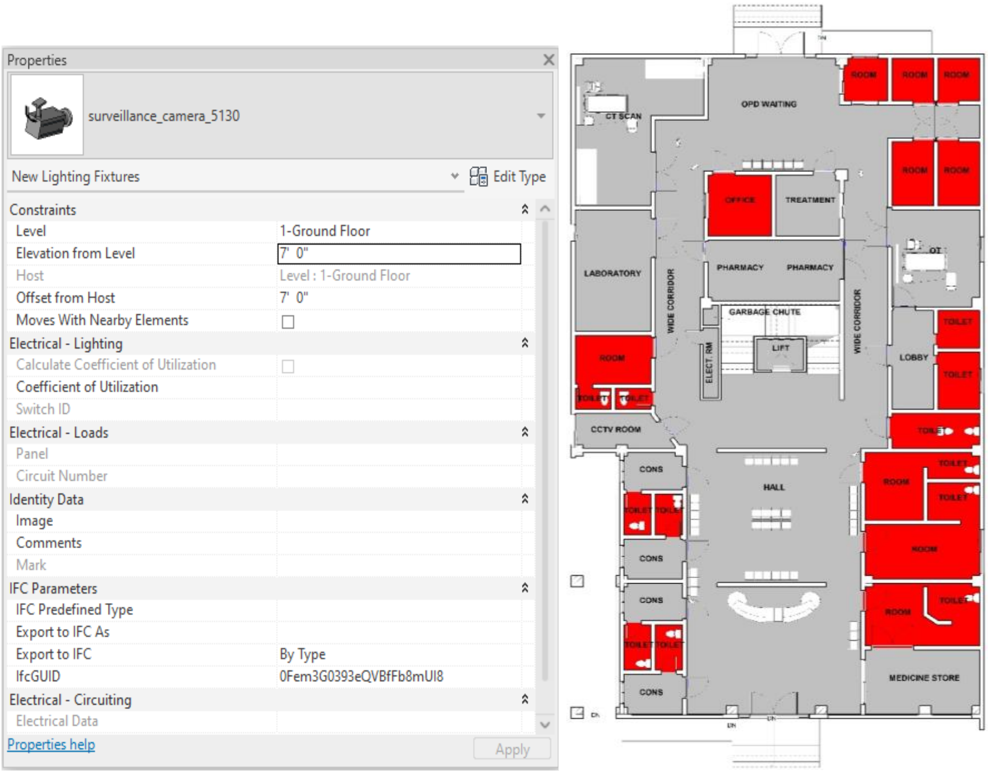

Figure 13: (left) Camera component loaded into Revit and (right) installed at ground floor using option B.

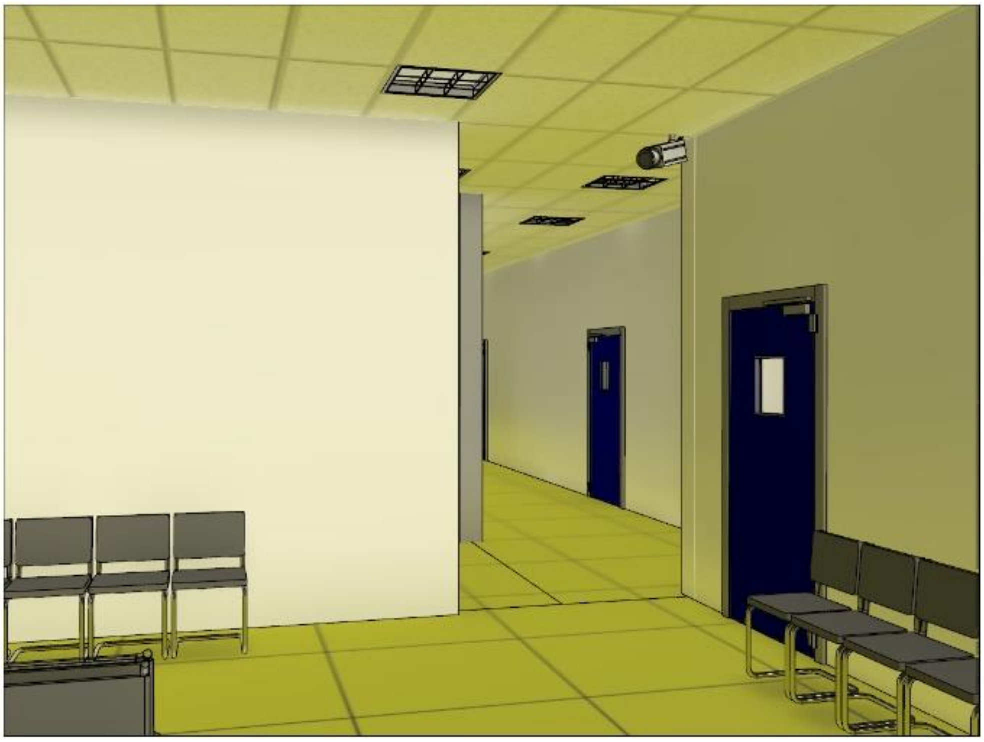

Figure 14: Camera installed in the BIM model of River Garden hospital at ground floor.

5. CONCLUSION

The primary objective of this study is to establish a framework for optimized and automated camera placement, addressing general and case-specific constraints within a target area. This framework aims to minimize the total number of cameras required while avoiding unnecessary overlap in their coverage areas. It involves extracting building coordinates from BIM, specifying coverage areas and constraints, and then employing a main program with an optimization algorithm to get camera positions. During the development of framework, PSO and GA were compared, and after conducting experiments GA was chosen as the better choice due to its better fitness values overall. Lastly, the optimized results are imported back in BIM, and cameras are automatically placed through a plugin. A validation experiment was conducted on the ground floor lobby of River Garden Hospital to validate the performance of the proposed optimization framework. The camera layout was designed for this area by both the proposed framework and manually by an industry professional and a comparative analysis was conducted to evaluate the results.

From the comparative analysis, the following significant contributions of the study were found: 1) The development of a framework with real-world applicability. This framework can be implemented in practical scenarios, offering a valuable tool for optimizing camera placement 2) Comparison of GA and PSO in context of camera placement, with GA being more effective in producing better camera positions. 3) The framework demonstrated improved coverage in the target region by increasing the overall coverage percentage and minimizing overlap between cameras. This results in a more comprehensive surveillance system with fewer blind spots. When compared with manual camera placement, the framework delivered 40% more coverage and a significant 56.9% reduction in overlap while using the same number of cameras. Achieving the same coverage as manual placement required one less camera with 27% increase in coverage and a 99% decrease in overlap, leading to a decrease in the overall cost of cameras installation. By accommodating cameras with various specifications, this framework ensures cost-efficient camera installation specific to the needs of each building. Using BIM models facilitates accurate and adaptable placement, minimizing blind spots and enhancing overall security. The framework's incorporation of constraints to prevent privacy violations or ethical concerns ensures compliance with regulations, making it an asset for both new constructions and existing buildings undergoing security upgrades. Its contribution lies in not only improving surveillance effectiveness but also in optimizing resource utilization within the building surveillance industry.

This study has demonstrated promising results; however, it does have some limitations. Firstly, the vertical rotation (tilt) of cameras and variable heights of obstacles were not considered. The study focused on a 2D implementation, omitting the Z factor associated with camera tilt and the variability in obstacle heights in a 3D environment. Future research will aim to enhance the existing framework to operate in 3D settings, accounting for the full range of camera movements and incorporating variable obstacle heights. Additionally, there will be a focus on optimizing the framework to reduce computational costs, particularly in a 3D environment where the optimization process tends to be more resource intensive.

REFERENCES

Abraham, A., and Jain, L. (2005). Evolutionary Multiobjective Optimization, Evolutionary Multiobjective Optimization: Theoretical Advances and Applications, 1-6.

Afroz, Z., Shafiullah, G. M., Urmee, T., Shoeb, M. A., and Higgins, G. (2022). Predictive modelling and optimization of HVAC systems using neural network and particle swarm optimization algorithm, Building and Environment, 209, 108681.

Albahri, A., and Hammad, A. (2016). Optimizing Surveillance Camera Placement in Buildings Using BIM.

Albahri Ameen, H., and Hammad, A. (2017). Simulation-Based Optimization of Surveillance Camera Types, Number, and Placement in Buildings Using BIM, Journal of Computing in Civil Engineering, 31(6), 04017055.

Alciatore, D., and Miranda, R. (1995). A winding number and point-in-polygon algorithm, Glaxo Virtual Anatomy Project Research Report, Department of Mechanical Engineering, Colorado State University.

Allam, A. S., Bassioni, H., Ayoub, M., and Kamel, W. (2023). Investigating the performance of genetic algorithm and particle swarm for optimizing daylighting and energy performance of offices in Alexandria, Egypt, Smart and Sustainable Built Environment, 12(3), 682-700.

Altahir, A., Asirvadam, V., Hamid, N., and Sebastian, P. (2022). Using artificial intelligence search in solving the camera placement problem, 109-127.

Altahir, A. A., Asirvadam, V. S., Hamid, N. H., Sebastian, P., Saad, N., Ibrahim, R., and Dass, S. C. (2017). Modeling Multicamera Coverage for Placement Optimization, IEEE Sensors Letters, 1(6), 1-4.

Assareh, E., Behrang, M. A., Assari, M. R., and Ghanbarzadeh, A. (2010). Application of PSO (particle swarm optimization) and GA (genetic algorithm) techniques on demand estimation of oil in Iran, Energy, 35(12), 5223-5229.

Beorkrem, C., Rodríguez, F. J. C., Karduni, A., and Radnia, A. A Big Data Approach to BIM Models.

Bourikas, L., Costanza, E., Gauthier, S., James, P., Kittley-Davies, J., Ornaghi, C., Rogers, A., Saadatian, E., and Huang, Y. (2016). Camera-based window-opening estimation in a naturally ventilated office, Building Research and Information, 46.

Chan, L. S. (2024). Piping network optimization for district heating system using an enhanced Genetic Algorithm searching method, Journal of Building Engineering, 95, 110078.

Chen, X., Zhu, Y., Chen, H., Ouyang, Y., Luo, X., and Wu, X. (2021). BIM-based optimization of camera placement for indoor construction monitoring considering the construction schedule, Automation in Construction, 130, 103825.

Cheng, J. C. P., Kwok, H. H. L., Li, A. T. Y., Tong, J. C. K., and Lau, A. K. H. (2022). BIM-supported sensor placement optimization based on genetic algorithm for multi-zone thermal comfort and IAQ monitoring, Building and Environment, 216, 108997.

Chvátal, V. (1975). A combinatorial theorem in plane geometry, Journal of Combinatorial Theory, Series B, 18(1), 39-41.

Črepinšek, M., Liu, S.-H., and Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey, ACM Comput. Surv., 45(3), Article 35.

Dahmane, W. M., Brahmia, M. e.-A., Dollinger, J. F., and Ouchani, S. (2020). A BIM-based framework for an Optimal WSN Deployment in Smart Building, 2020 11th International Conference on Network of the Future (NoF), 110-114.

David, P., Idasiak, V., and Kratz, F. (2007). A Sensor Placement Approach for the Monitoring of Indoor Scenes, Smart Sensing and Context, 110-125.

Deep, K., Singh, K. P., Kansal, M. L., and Mohan, C. (2009). A real coded genetic algorithm for solving integer and mixed integer optimization problems, Applied Mathematics and Computation, 212(2), 505-518.

Goldstein, E. (2021). CCTV Helps Both Patients And Staff.

Guo, K., and Zhang, L. (2021). Multi-source information fusion for safety risk assessment in underground tunnels, Knowledge-Based Systems, 227, 107210.

Han, Z., Li, S., Cui, C., Song, H., Kong, Y., and Qin, F. (2019). Camera planning for area surveillance: A new method for coverage inference and optimization using Location-based Service data, Computers, Environment and Urban Systems, 78, 101396.

Hassanat, A., Almohammadi, K., Alkafaween, E. a., Abunawas, E., Hammouri, A., and Prasath, V. S. (2019). Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach, Information, 10(12), 390.

Hikvision (5MP). Model no. DS-2CD2655FWD-IZS, Varifocal Bullet Network Camera.

Horster, E., and Lienhart, R. (2006). Approximating Optimal Visual Sensor Placement, 2006 IEEE International Conference on Multimedia and Expo, 1257-1260.

Hsiao, S.-J., and Sung, W.-T. (2021). Realization of IoT Integration System of LED Based on Particle Swarm Optimization, Intelligent Automation \& Soft Computing, 27(2), 499--517.

Kenichi, Y., and Hitoshi, K. (2008). Optimum camera placement considering camera specification for security monitoring, 2008 IEEE International Symposium on Circuits and Systems (ISCAS), 2114-2117.

Kennedy, J., and Eberhart, R. (1995). Particle swarm optimization, Proceedings of ICNN'95 - International Conference on Neural Networks, 4, 1942-1948 vol.1944.

Kim, J., Ham, Y., Chung, Y., and Chi, S. (2019). Systematic Camera Placement Framework for Operation-Level Visual Monitoring on Construction Jobsites, Journal of Construction Engineering and Management, 145(4), 04019019.

Kim, K., Murray, A., and Xiao, N. (2008). A Multi-Objective Evolutionary Algorithm for Surveillance Sensor Placement, Environment and Planning B: Planning and Design, 35, 935-948.

Li, H., Love, P. E. D., and Ogunlana, S. (1998). Genetic algorithm compared to nonlinear optimization for labour and equipment assignment, Building Research & Information, 26(6), 322-329.

Li, M., Chen, H., Shi, X., Liu, S., Zhang, M., and Lu, S. (2019). A multi-information fusion “triple variables with iteration” inertia weight PSO algorithm and its application, Applied Soft Computing, 84, 105677.

McArthur, J. J., Shahbazi, N., Fok, R., Raghubar, C., Bortoluzzi, B., and An, A. (2018). Machine learning and BIM visualization for maintenance issue classification and enhanced data collection, Advanced Engineering Informatics, 38, 101-112.

Murray, A. T., Kim, K., Davis, J. W., Machiraju, R., and Parent, R. (2007). Coverage optimization to support security monitoring, Computers, Environment and Urban Systems, 31(2), 133-147.

Nam, Y., and Hong, S. (2014). Optimal placement of multiple visual sensors considering space coverage and cost constraints, Multimedia Tools and Applications, 73(1), 129-150.

Panda, S., and Padhy, N. P. (2008). Comparison of particle swarm optimization and genetic algorithm for FACTS-based controller design, Applied Soft Computing, 8(4), 1418-1427.

Schmitt, M., and Wanka, R. (2015). Particle swarm optimization almost surely finds local optima, Theoretical Computer Science, 561, 57-72.

Shi, Y., and Eberhart, R. (1998). A modified particle swarm optimizer, 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), 69-73.

Shukla, A., Pandey, H. M., and Mehrotra, D. (2015). Comparative review of selection techniques in genetic algorithm, 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 515-519.

Si, P., Wu, C., Zhang, Y., Jia, Z., Ji, P., and Chu, H. (2017). Barrier Coverage for 3D Camera Sensor Networks, Sensors, 17(8).

Soh, R., and Ahmad, N. (2019). The Implementation of Video Surveillance System at School Compound in Batu Pahat, Johor, 119-134.

Suresh, M. S. S., Narayanan, A., and Menon, V. (2020). Maximizing Camera Coverage in Multicamera Surveillance Networks, IEEE Sensors Journal, 20(17), 10170-10178.

Syu, S.-R., and Peng, C.-H. (2022). Optimizing Placements of 360-degree Panoramic Cameras in Indoor Environments by Integer Programming, arXiv preprint arXiv:2211.07296.

Taylor, G. (1994). POINT IN POLYGON TEST, Survey Review, 32(254), 479-484.

Vasconcelos, J. A., Ramirez, J. A., Takahashi, R. H. C., and Saldanha, R. R. (2001). Improvements in genetic algorithms, IEEE Transactions on Magnetics, 37(5), 3414-3417.

Wang, D., Tan, D., and Liu, L. (2018). Particle swarm optimization algorithm: an overview, Soft Computing, 22.

Wang, H., Wang, G., and Li, X. (2021). Image-based occupancy positioning system using pose-estimation model for demand-oriented ventilation, Journal of Building Engineering, 39, 102220.

Wang, J., Song, X., and El-Latif, A. A. A. (2022). Single-Objective Particle Swarm Optimization-Based Chaotic Image Encryption Scheme, Electronics, 11(16), 2628.

Wang, X., Gu, H., Zhang, H., and Chen, H. (2019). Novel RPSO Based Strategy for Optimizing the Placement and Charging of a Large-Scale Camera Network in Proximity Service, IEEE Access, 7, 16991-17000.

Xie, Z., Liu, X., Li, Y., Zhang, H., and Xiang, Q. (2023). Camera placement optimization for CCTV in rail transit using BIM, Measurement and Control, 56(9-10), 1499-1509.

Xu, Y., and Lei, B. (2009). Particle swarm optimization to improve the coverage of a camera network, MIPPR 2009: Medical Imaging, Parallel Processing of Images, and Optimization Techniques, 7497, 312-317.

Xu, Y., Lei, B., Shuifa, S., Fangmin, D., and Chilan, C. (2010). Three particle swarm algorithms to improve coverage of camera networks with mobile nodes, 2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), 816-820.

Yixuan, W. (2022). Indoor Optimal Design of Building Renovation Environment Space Layout Based on Genetic Algorithm, Procedia Computer Science, 208, 539-545.

Zhang, Y., Luo, H., Skitmore, M., Li, Q., and Zhong, B. (2019). Optimal Camera Placement for Monitoring Safety in Metro Station Construction Work, Journal of Construction Engineering and Management, 145(1), 04018118.