OPTIMIZING SURVEILLANCE CAMERA PLACEMENT IN BUILDINGS USING BIM

Ameen Hamza Albahri[a] and Amin Hammad[b]

[a]Ph.D. Candidate, Faculty of Engineering and Computer science, Concordia University, Canada
[b]Professor, Faculty of Engineering and Computer science, Concordia University, Canada
E-mail: a_albah@encs.concordia.ca, hammad@ciise.concodia.ca

Abstract

Surveillance cameras are becoming indispensable tools due to their ability to ensure security in buildings. However, many organizations claim that purchasing, installing and maintaining these cameras are costly. Furthermore, current practices have conducted different 2D and 3D techniques to find the optimum camera placement inside buildings. These techniques have many limitations which can affect the camera coverage. On the other hand, the technology of Building Information Modelling (BIM) is becoming a well-known method that can help in various tasks related to the Architectural, Engineering and Construction (AEC) industry. This study aims to find the optimum placement of a fixed camera inside a BIM model. The proposed method includes: (1) using BIM for specifying constraints; (2) identifying decision variables; and (3) integrating simulation and optimization.

Keywords –

Surveillance cameras; BIM; Optimization; Genetic Algorithm

1 Introduction

The issue of optimizing the camera placement in buildings is becoming significant and it requires more attention by the scientific community. This paper presents a simulation-based method to optimize the placement of a single fixed camera in a building using the technology of Building Information Modeling (BIM). The optimization should ensure the maximum coverage for a specific area. The method is adopting Genetic Algorithm (GA) to address the optimization problem.

The remaining of this work includes the following. Section 2 includes the literature review. Section 3 explains the proposed method. Section 4 implements the proposed method and validates it by conducting a case study. Section 5 summarizes the method and future work.

2 Literature Review

Chelliger [2] outlines that identifying risks and establishing their mitigation standards are the most important features of an efficient security system. Wilkis [3] highlighted that no security system can ensure complete coverage. One of the most significant and serious issues is that surveillance cameras might miss hidden areas behind objects (e.g. columns). These hidden areas can become a vulnerability of an asset. Thus, numerous studies have taken place in order to improve the placement of these surveillance cameras.

The study of Bigdeli et al. [4] was conducted to evaluate the performance of surveillance cameras in railway platforms. The objective of this study is to guarantee clear detection of people during events (e.g. crowded periods). The method showed a real-life trial system that is composed of diverse video analytic systems for detecting important events. The work highlighted that the process of placing various types of surveillance cameras is time consuming.

Furthermore, Kelly [5] mentioned that the placement of surveillance cameras is becoming expensive. Nam and Hong [6] optimized the placement of multiple cameras in indoor spaces taking into consideration the cost constraints. The proposed method includes four steps: space modeling, agent modeling, generating trajectories and selection of optimal camera placement. However, this study did not take into account the importance of a clear segmentation process for the monitoring area, which can facilitate achieving accurate calculation of the camera coverage. Also, their way of finding the important areas by using the path-finding algorithm is inaccurate, since some of these areas might include constraints such as privacy areas.

Yabuta and Kitazawa [7] used a 2D method to find the optimum camera placement in a specific area. Their study specified a number of important parameters related to surveillance cameras as the following: visual distance, Field of View (FOV) and resolution of the camera. Their method showed some weaknesses as the following: (1) the 2D method of calculating the coverage is inefficient in cases where hidden areas are available behind objects because of the variety of elevations; (2) their segmentation process divided the monitoring area into irregular shapes, which will result in inaccurate calculations. This is because small shapes behind obstacles might be missed during the camera coverage process.

Moreover, Chen et al. [8] conducted a study to optimize the real angle of depression of surveillance cameras. The study aimed to validate the applicability of Building Information Modeling (BIM) in improving the performance of surveillance cameras. The study highlighted the role of BIM in detecting obstacles inside the model that are able to affect the camera coverage and placement. The study adopted a 2D method for calculating the camera coverage by using a tool in Revit called "the Fill Region Tool". Although their method proved the ability of BIM in optimizing the real angle of depression for a number of surveillance cameras, it did not consider the optimization of the camera placement.

3 Proposed method

3.1 Using BIM for specifying constraints

Several elements can impact the optimization process as the following:

(1) The placement area of the camera (e.g. ceiling).
(2) The monitored area including the existence of different floor elevations.
(3) The existence of different important areas in the monitored area.
(4) The existence of privacy areas such as bathrooms.
(5) The constraints on the monitored area (e.g. columns) and the placement area (e.g. HVAC system). Other constraints are light sources, shiny surfaces and ceiling signboards which can also affect the visibility of the camera.

BIM is able to identify the locations of different constraints located on the placement or the monitored areas. This ability can help in providing the valid coordinates of the camera placement to the optimization process. This can decrease the time of the optimization process by making the camera placement restricted to valid positions and avoiding the positions of constraints. This study is focusing on four types of constraints:

(1) Geometrical constraints on the ceiling (e.g. ceiling signboards).
(2) Operational constraints (e.g. in places where the camera is exposed to vibration generated by the HVAC system).
(3) Logical constraints (e.g. in places where the camera is facing walls).
(4) Legal constraints (e.g. in case if the monitoring area includes privacy areas).

3.2 Identifying decision variables

The decision variables of the camera placement include the position and the orientation of the camera. The camera position is represented by the values of *X, Y* and *Z* coordinates, while the camera orientation is represented by the values of *Pan* and *Tilt* angles. The *Y* variable (the height) will be fixed during the optimization process in a way to ensure the maximum height of the camera. The solution of the GA includes one gene that gives a specific pose of the single camera on the ceiling including five variables *X, Y, Z, Pan* and *Tilt* angles.

Moreover, it is important to note that a bigger population size results in better optimization results and longer computation time, and vice versa. Parameters related to the crossover and mutation processes should be also specified in the GA. For simplicity, this study adopted the rectangular shape for the placement area.

The Preference-Inspired Co-evolutionary Algorithm (PICEA) is used [9]. The general algorithm aims to generate candidate solutions for a specific number of generations. Every generation includes a number of parents which are exposed to crossover and mutation processes in order to produce new offsprings. Parents and offsprings are then combined together and the members of the resulting population are evaluated based on their fitness. The resulting population is then subjected to a truncation selection in order to produce new generation of parents. This process is repeated until the termination criterion is achieved (i.e. maximum number of generations).

3.3 Integrating simulation and optimization

In order to apply GA for the camera placement process, the following steps are applied:

(1) BIM is used to identify the search space of the placement area and exclude the constraints.
(2) These feasible coordinates are then sent to the optimization process in order to generate the decision variables.
(3) As shown in Figure 1, GA will initialize the first generation (*g*) which has specified population size (*P*) of solutions.
(4) The first solution (i.e. gene) of the first generation is sent to a coverage simulation process in order to calculate the fitness. This process will continue until all solutions of the current generation are

evaluated.

(5) GA will then produce the next generation by applying the crossover and mutation. This process will continue until the GA finishes from all generations.

(6) Finally, the optimum solutions are specified and simulated in BIM.

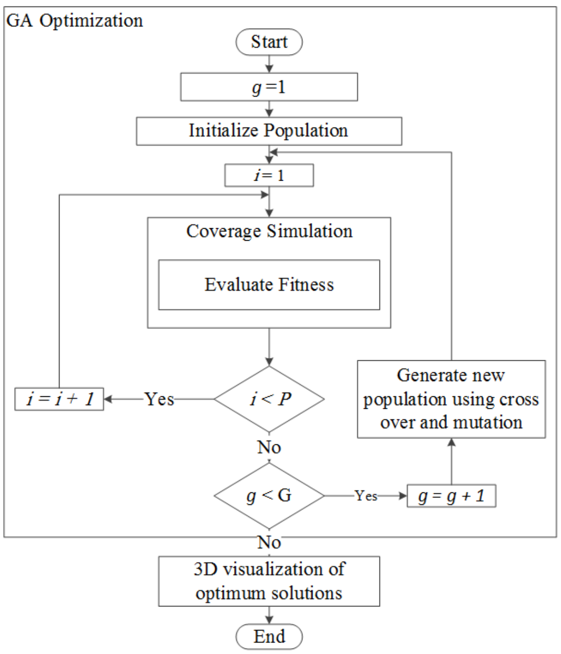

Figure 1 Proposed method

The previous process includes integration between two systems. One represents the optimization process, while the other is responsible of calculating the coverage. In order to calculate the camera coverage, the following steps should be conducted: (1) accessing the data from the BIM, (2) adjusting the camera features (i.e. FOV, the camera height and clipping planes (far and near points)), (3) placing the camera automatically based on the data of each solution in the population, (4) generating cells on the targeted surface (i.e. the floor), and (5) calculating the coverage using Equation 1, [10] and Equation 2 [1]. The output will be sent to the GA to evaluate the fitness as previously discussed. It is also important to mention that the objective function is aiming to maximize the coverage by applying Equations 1 to 7.

$$CC_{a_i} = \sum_{v=1}^{n`} CC_{iv} \times IV_i \qquad \text{Equation 1}$$

$$Coverage\ of\ camera\ K_x = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{v=1}^{n`} \frac{CC_{iv} \times IV_i}{C_{ij} \times IV_i} \qquad \text{Equation 2}$$

Subject to: $0 \le X \le L$, L is the maximum value of the length — Equation 3

Cameras should be on the ceiling or on the upper part of the wall

$0 \le Y \le H$, H is the maximum value of the height — Equation 4

$0 \le Z \le W$, W is the maximum value of the width — Equation 5

$0 \le Pan \le 360°$ — Equation 6

$0 \le Tilt < 90°$ — Equation 7

where C_{ij} represents the cell j in important area i, and j = 1:n, IV_i is the importance value assigned all cells in area i, and i = 1:m, CC_{iv} is the covered cell v in area a_i and v = 1:$n`$, K_x is camera x.

4 Implementation and case study

A case study is applied in order to validate the proposed method. The case study is implemented on the ground floor of the Engineering and Visual Art (EV) building in Concordia University, as shown in Figure 2. This figure shows the current locations of seven surveillance cameras installed inside this model.

Figure 2 BIM model aligned with FOVs in the game engine [1]

This case study is validating the proposed method using BIM, game engine tool and Matlab for optimizing the placement of single fixed camera in the BIM model. In addition, the floor has some physical constraints such as columns that have an impact on the camera coverage. Autodesk Revit was used to develop the BIM model [11]. The BIM model is sent to Unity3D [12] using FBX format for the purpose of simplifying the programming processes which used C# language.

In Unity3D, the ground floor of the BIM model was divided into important areas as shown in Figure 3. The important areas include cells which have equal cubic shapes. These cells are assigned with different importance values based on their positions inside the model. The importance values can help in defining the optimum position of the camera. These cells can offer accurate results of the camera coverage by adjusting their sizes as discussed in Albahri and Hammad [1]. Also, a grid system is developed in order to facilitate the camera placement process.

As previously discussed, the optimization process includes integration between the GA tool of Matlab [13] and Unity3D. The GA is set to 65 generations. The population size of each generation is 200 candidate solutions. GA will send these generations one by one in the form of a text file to Unity3D.

Inside Unity3D, this text file is processed as the following: (1) Unity3D runs the placement process for the selected 3D camera based on the given solution in the text file; (2) Then, it calculates the coverage of the selected camera; and finally (3) The results of the camera coverage are saved in text file and sent to Matlab in order to continue running the next generations.

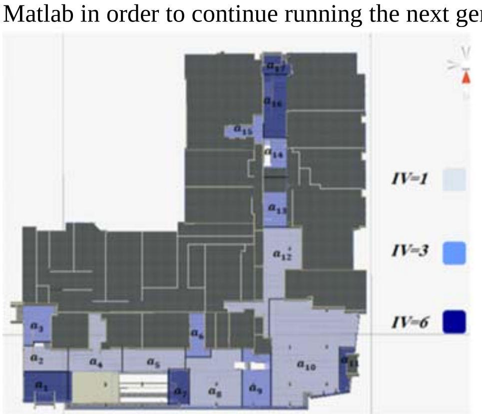

Figure 3 Plan view showing the important areas with their Importance Values (IV) [1]

It is important to mention that the optimization process is conducted on the areas from a_5 - a_8, as shown in Figure 4. These areas were covered by cells that have different importance values (i.e. value of 1 for the lowest importance, value of 3 for the medium importance and value of 6 for the highest importance).

Moreover, the placement surface has the dimensions of 2500 cm on the *Z* axis and 1400 cm on the *X* axis. The placement surface is divided by a number of grids that can define the camera placement. In this case study, 56 grids were generated on the *X* axis, and 100 grids on the *Z* axis. The selected camera has the FOV = 60°.

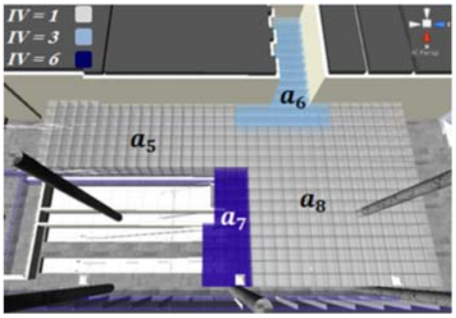

Figure 4 Areas a_5 to a_8 on the monitoring area

Table 1 shows the best 10 solutions with their coverage results. For example, in solution 10, the GA found a camera position at *X* = 100 cm; *Y* = 500 cm; *Z* = 2500 cm; pan angle = 205°; and tilt angle = 61°. This solution offered camera coverage of 65%. Solution 1 has the optimal coverage of 90%, as shown in Figure 5, with coordinates *X* = 1250 cm, *Y*=500 cm, *Z* = 1550 cm, pan angle = 115° and tilt angle = 62°. Figure 6 shows the global fitness, best fitness and the mean fitness curves. The best fitness curve displays a fluctuation in the camera coverage from the 8th generation to the 27th generation. This is because GA is randomly searching for various solutions which lead to various fitness results. The curve faced coverage of 82% in the 28th generation and continued increasing until it reached 90%, which is the global fitness value, in the 58th generation. The GA showed a slight increasing of the coverage from the 54th generation to the 65th generation with coverage values from 88% to 90%, respectively.

Table 1 Best 10 solutions of the optimization process

Solution #	Coverage (%)
1	90
2	89
3	88
4	86
5	84
6	82
7	78
8	74
9	69
10	65

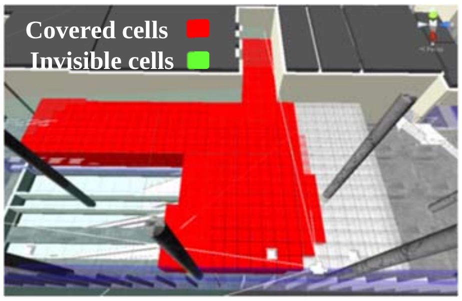

Figure 5 3D Simulation of solution 1

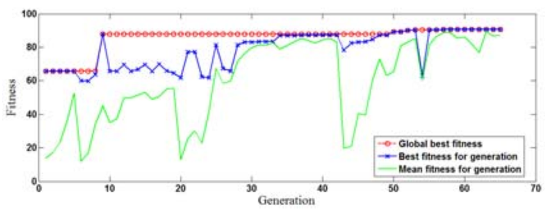

Figure 6 GA fitness progression of the case study

5 Summary and future work

This work presented a simulation-based optimization method for the placement of a single fixed camera inside a BIM model. The optimal placement of the camera aims to achieve the maximum coverage in a specific area inside a building. In addition, BIM is used to define the constraints on the ceiling in order to determine the valid positions of the camera placement. The proposed method adopted integration between BIM and a GA optimization process. The proposed method included the following: (1) using BIM for specifying constraints; (2) identifying decision variables; and (3) integrating simulation and optimization. A case study was developed to validate the method. This method can assist individuals who are responsible of the camera installation to accurately determine the suitable placement in the facility. Future work will include the placement of multiple camera types inside buildings.

Acknowledgement

Special thanks to the Saudi Arabian Cultural Bureau in Canada for their support during the planning and the implementation of this work.

References

[1] Albahri, A., and Hammad, A. (2015). Evaluating the Coverage of Surveillance Cameras Using BIM and Game Engine. In 2nd International Conference on Civil and Building Engineering Informatics (ICCBEI 2015), Tokyo.

[2] Chelliger, D., (2008). From the Ground up: Security for Tall Buildings. CRISB Report. Connecting Research in Security to Practice. In ASIS International. Alexandria, VA, USA.

[3] Wilkis, K. (2015). Who's Watching You? Video Surveillance-User Beware. Extracted on 23 Oct 2015 from: < http://blog.willis.com/2015/05/whos-watching-you-video-surveillance-user-beware/ >.

[4] Bigdeli, A., Lovell, B. C., Sanderson, C., Shan, T., and Chen, S. (2007). Vision Processing in Intelligent CCTV for Mass Transport Security. In IEEE Workshop on Signal Processing Applications for Public Security and Forensics. (SAFE'07), 1-4.

[5] Kelly, H. (2013), After Boston: The Pros and Cons of Surveillance Cameras, CNN Tech. Extracted at 25 Sep 2013 from: <http://www.cnn.com/2013/04/26/tech/innovation/security-cameras-boston-bombings/>.

[6] 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.

[7] Yabuta, K., and Kitazawa, H. (2008). Optimum Camera Placement Considering Camera Specification for Security Monitoring. IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2114-2117.

[8] Chen, H. T., Wu, S. W., and Hsieh, S. H. (2013). Visualization of CCTV Coverage in Public Building Space Using BIM Technology. Visualization in Engineering, 1(1):1-17.

[9] Purshouse, R.C., Jalba, C. and Fleming, P.J. (2011). Preference-Driven Co-evolutionary Algorithms Show Promise for Many-Objective Optimisation. In Evolutionary Multi-criterion Optimization (EMO), pp. 136-150.

[10] Amriki, K. A., and Atrey, P. K. (2014). Bus Surveillance: How Many and Where Cameras Should be Placed. Multimedia Tools and Applications, 71 (3): 1051-1085.

[11] Autodesk, (2014). Revit. Extracted on 16 Sep 2014 from: <http://www.autodesk.com/products/revit-family/overview>

[12] Unity (2015). Create and Connect With Unity 5. Extracted on: 28 Oct 2015 from: < https://unity3d.com/ >.

[13] De Freitas, A. (2012). Matlab Central: Open Genetic Algorithm Toolbox. Extracted at 15 Oct 2015 from: <http://www.mathworks.com/matlabcentral/fileexchange/37998-open-genetic-algorithm-toolbox>.