Computers, Environment and Urban Systems 78 (2019) 101396

Contents lists available at ScienceDirect

Computers, Environment and Urban Systems

journal homepage: www.elsevier.com/locate/ceus

Camera planning for area surveillance: A new method for coverage inference and optimization using Location-based Service data

Zhigang Han[a,c,d,e], Songnian Li[b,*], Caihui Cui[c,e], Hongquan Song[a,c,e], Yunfeng Kong[a,c], Fen Qin[a,c,d]

[a] Key Laboratory of Geospatial Technology for the Middle and Lower Yellow River Regions (Henan University), Ministry of Education, Kaifeng, China
[b] Department of Civil Engineering, Ryerson University, 350 Victoria St., Toronto, ON M5B 2K3, Canada
[c] College of Environment and Planning, Henan University, Kaifeng, China
[d] Institute of Henan Spatio-temporal Bigdata Industrial Technology, Henan University, Zhengzhou, China
[e] Institute of Urban Bigdata, Henan University, Kaifeng, China

ARTICLE INFO

Keywords:
Video surveillance
Camera planning
Spatial optimization
Location-based service
Maximal coverage location problem

ABSTRACT

Along with the rapidly growing volume of public security events, efficient camera planning and configuration methods have been one of the most crucial challenges in the video surveillance field. How to allocate different types of surveillance cameras in an area is one of the fundamental problems; however, limited methods have been available for generating the deployment parameters of cameras. The purpose of the paper is to explore camera planning based on multi-source Location-based Service data. The main idea is to infer the camera coverage by the building footprints, Point of Interests (POI) and social network record (WeChat) data, and to optimize the camera placement using the Maximal Coverage Location Problem-Complementary Coverage (MCLP-CC) model. Based on the probability of cell monitored with the calculation of viewshed analysis, the candidate location with max probability is selected. The essential spots in the surveillance area are uncovered by the combination of the kernel density estimation of POIs and WeChat data. The inference algorithm of the location, the field of view angle, orientation yaw, and visible distance parameters are proposed using the candidate location and critical spots in the viewshed polygon. The MCLP-CC is modeled and implemented by Python scripts and Gurobi software. The experiment shows that the proposed method can generate the detailed camera parameters including location, the field of view angle, orientation yaw, and visible distance with the lower occlusion and overlapping ratio for camera coverage. We believe that the integration of the coverage inference and optimization methods into the existing GIS platform will promote a variety of innovative applications in the camera planning area.

1. Introduction

In the recent decade, the growing urgent requirement for public security creates new challenges for optimal camera planning (i.e., optimal camera location, coverage, orientation, etc.) in urban public spaces or infrastructure like railway stations, shopping malls, city squares, and residential areas (Liu, Sridharan, & Fookes, 2016; Murray, Kim, Davis, Machiraju, & Parent, 2007). The camera planning is a process of generating detailed parameters such as location, azimuth, and visible distance, etc. for camera deployment (Liu et al., 2016). The proper camera location and other configuration parameters can not only provide cost savings with the same or higher level of utility but also bring significant benefits to subsequent video analysis and detection tasks. In a specific public area, how to scientifically and low-costly allocate a certain number of different types of surveillance cameras at particular locations, improve monitoring coverage and reduce monitoring blind or obscured spots and overlapped area, has become a fundamental problem in deploying video surveillance systems.

Camera planning has received increased attention across a number of disciplines in recent years as the widespread deployment of video surveillance systems in urban areas (Liu et al., 2016). A large part of the research in camera planning has been focusing on developing an effective optimization method. The traditional spatial optimization methods, such as MCLP and its extension of BCLP (Backup Coverage Location Problem), have been used for camera coverage optimization (Dell'Olmo, Ricciardi, & Sgalambro, 2014; Murray et al., 2007). With the advantage of optimization and operation research, the researchers introduced multi-objective genetic algorithms (Kim, Murray, & Xiao, 2008), particle swarm optimization (Xu, Lei, & Hendriks, 2011), artificial bee colony algorithms (Chrysostomou & Gasteratos, 2012), etc.

* Corresponding author.
E-mail addresses: zghan@henu.edu.cn (Z. Han), snli@ryerson.ca (S. Li).

https://doi.org/10.1016/j.compenvurbsys.2019.101396
Received 9 February 2019; Received in revised form 22 August 2019; Accepted 23 August 2019
Available online 03 September 2019

for camera placement optimization. However, camera planning can be affected by quite a number of factors, including camera parameters (i.e., position, orientation, etc.) and static and dynamic objects (i.e., buildings, people's activities hot spots, etc.) in the scene. The static objects are the primary targets in existing camera planning method. Unfortunately, the dynamic objects are ignored, and so far these factors have not been fully used to camera planning in spite of their potential to enhance planning efficiency.

In recent years, along with mobile Location-based Service (LBS) technologies that have been rapidly developed, large amount of multi-source LBS data can be obtained from Internet, public services and social networks, including points of interest (POIs), mobile phone data, social network records, building footprints, etc. These data can indicate building occlusions and critical spots in surveillance areas. For example, the building footprints show the wall, which will obscure the field of view of the camera. The LBS data such as check-in data can uncover the hotspots of crowd activities, which is also the key area should be covered by the camera. By detecting the hotspot areas of crowd activities, the orientation parameters of surveillance cameras can be inferred, which can effectively improve the coverage of video surveillance on the location of crowd activities in the surveillance area and enhance the function of the video surveillance. In this paper, we propose a camera planning method using these multi-source LBS data to infer and optimize camera coverage for area surveillance. Our process focuses on dealing with the following two key issues identified for camera planning: (1) camera coverage inferring. Based on the coverage modeling and viewshed calculation, the two types of camera coverage (circle and fan) are inferred using building footprints, POIs, and WeChat data; (2) camera coverage optimization. The MCLP-CC (Maximal Coverage Location Problem complementary coverage) method is used to optimize the camera coverage by considering partial coverage of the demand unit area using the facility service area method in GIS. The primary aim of this study is to explore the camera planning based on multi-source LBS data, which can be used to generate the optimal camera configuration parameters including camera position, orientation, radius and FOV (field of view) angle in detail.

The paper is organized as follows. Section 2 presents related works. Section 3 proposes the methodology, including three main steps: coverage modeling and viewshed calculation, coverage inferring, and coverage optimization. The experiment with the method and the results obtained are discussed in Section 4. Section 5 concludes the paper with a brief summary and discussions.

2. Related works

Early work on camera planning can go back to the famous art gallery problem in computational geometry field. It is the assignment of finding the minimum number of guards in an art gallery to achieve the maximum visual coverage at different positions (Chvatal, 1975; Murray et al., 2007). As the subject of much theoretical work, the efficient algorithm to generate an optimized camera position for area surveillance has been the intense subject of many research areas such as visual sensor network, multimedia, and GIS. For example, Binary Integer Programming (BIP) methods (Gonzalez-Barbosa, García-Ramírez, Salas, & Hurtado-Ramos, 2009), MCLP or BCLP (Murray et al., 2007), Genetic Algorithms (Feng, Liu, & Wang, 2014; Kim et al., 2008), Particle Swarm Optimization (Conci & Lizzi, 2009; Fu, Zhou, & Deng, 2014; Morsly, Aouf, Djouadi, & Richardson, 2012), agent-based camera placement method (Nam & Hong, 2014), Semidefinite Programming (Zhao, 2011), and Distributed Mean Shift algorithm (Wang, Wu, & Long, 2013) have been used for camera planning with different task-specific requirements and real-world constraints. More related to our work, Liu et al. (Liu, Sridharan, Fookes, & Wark, 2014) introduced a TDSA (Trans-Dimensional Simulated Annealing) algorithm to serves as a good example to show how to get the optimal camera locations and orientations. However, the camera FOV angle is fixed. Hörster (Hörster & Lienhart, 2006) identified the entrances points that have a higher level of the essence than others for coverage optimization. But the other related spatial information such as POIs and the other key spots in an area is not fully used in their method. A more synthetic review of the state-of-the-art of camera planning was carried out in the literature of Costa and Guedes (2010) and Liu et al. (2016).

Camera planning for area surveillance includes the location decision process. It is important to determine the best facility locations among many candidates for all of the public or private sectors (Church & Murray, 2009). For service facilities, the location decision process aims to identify the best facility location to get the maximal areal coverage. The MCLP, originally defined by Church and ReVelle (1974), has often been used for this purpose. The MCLP method assumes that the demand is represented as a few points. These demand points are either completely covered or not by the service area of a facility. Thus, the MCLP method is a type of binary coverage. It is widely used to site service facilities in a series of fields, including camera placement (Murray et al., 2007), banking facility location (Xia et al., 2010), police patrol areas (Curtin, Hayslett-McCall, & Qiu, 2010), Wi-Fi equipment (Lee & Murray, 2010), urban fire stations (Murray, 2013; Yao, Zhang, & Murray, 2019), medical drone launch sites (Pulver & Wei, 2018), etc. As the binary coverage of the demand with the facility, the MCLP method ignores the partial coverage of the polygon demands by the service area, and it is found that the modeling results depend on the scale and unit definition (e.g., Modifiable Areal Unit Problem, or MAUP) (Murray, 2018; Wong, 2009). To consider partial coverage, a number of MCLP extensions including MCLP-implicit, MCLP-explicit, MCLP-Complementary Coverage (MCLP-CC) were introduced (Murray, Tong, & Kim, 2010; Tong, 2012). For example, by the combination of point-oriented or area-oriented demand, Tong and Wei (2017) proposed a new integrated model named MCLP-Mixed Representation (MCLP-MR). Wei's studies (Wei, 2016) indicated that the MCLP-CC model can get the largest facility coverage with reasonable computational efforts.

As another variable of the MCLP method, coverage model definition, namely, how do define the camera coverage area, is another key problem in camera planning. Mavrinac and Chen (2013) classified the coverage models into geometric coverage models and topological models. The former models the physical area or volume of the camera scene which can be inferred with the camera and the scene parameters. The latter describes the topology of mutual coverage overlap for a multi-camera system to track moving targets. Some of the previous studies had used disc/circle shape or viewshed polygon to describe the camera coverage (Liu et al., 2016). This is not an issue when omni-directional cameras are used for surveillance. However often times, the perspective cameras are mainly used to reduce building occlusion and construction costs. A number of researchers have attempted to use the isosceles triangle or fan shape, but many of them have a fixed angle. Wang et al. (Wang, Liu, Zhang, & Wang, 2017) introduced a coverage inference algorithm for the camera based on multistage grid subdivision by considering obstacles in the scene, but the dynamic objects in the scene such as people's activities are not fully utilized to infer the camera coverage. The importance and the hot spots of the people's activities as the key monitor target of the camera in an area can be indicated by the multi-source LBS data such as building footprints, POI, and social network data which could be collected on the internet.

In summary, there are a number of methods for camera planning in the area, and the MCLP-related models have been used to site service facilities in a range of fields. The review of existing literature indicates that limited effort has been made in integrating different camera parameters such as camera location, the field of view angle, orientation yaw, and visible distance with the lower occlusion and overlapping ratio in camera planning. To tackle this issue, the building footprints, the POIs, and the social network records data and the MCLP-CC model are used to infer and optimize the camera coverage in this study.

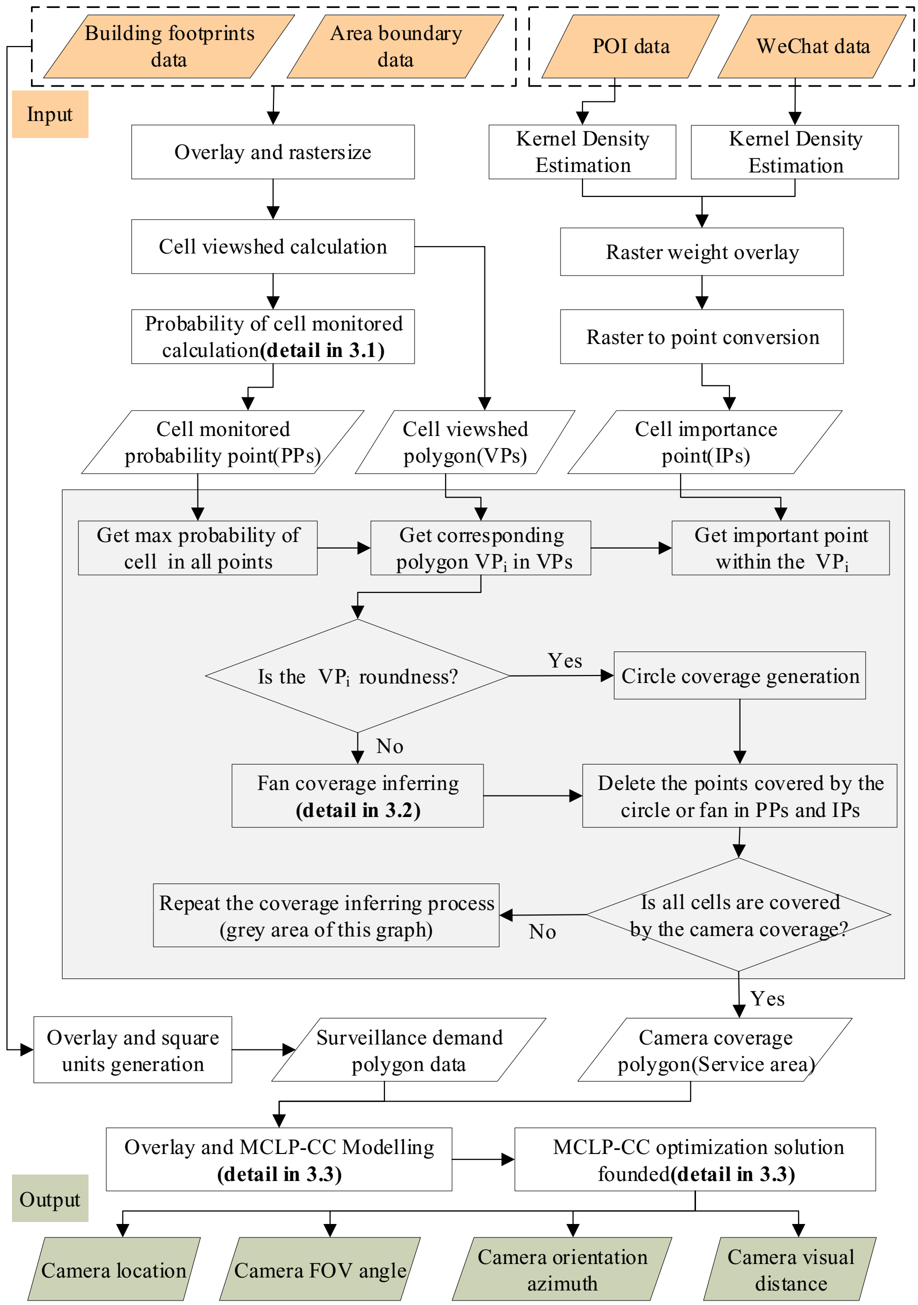

Fig. 1. The camera coverage inferring and optimization method.

3. Methodology

This section proposes a general method for camera coverage inferring and optimization method as shown in Fig. 1. We emphasize the generation of multiple camera parameters including location, FOV angle, orientation azimuth, and visible distance to support camera configuration and deployment through optimization. We envision that more and more multi-source LBS data is valuable for camera coverage inference. Our method consists of two main parts: the camera coverage inferring and the coverage optimization. The four types of data such as building footprints, surveillance area boundary, POIs, and Tencent WeChat records are the input data.

3.1. Coverage modeling and monitored probability calculation

As required by the MCLP optimization method, the camera coverage model must be formulated first as the service area of the facility in MCLP, which could be inferred by the multi-source LBS data. The scene captured by each camera corresponds to the real-world area. The camera's field of view can be determined according to the intrinsic and external parameters of the camera. Ordinary surveillance cameras include omnidirectional cameras, perspective cameras, and Pan-Tilt-Zoom (PTZ) cameras. To simplify the analysis, we ignore the time factor of the pan-tilt-zoom action of PTZ camera and assign it as an omnidirectional camera.

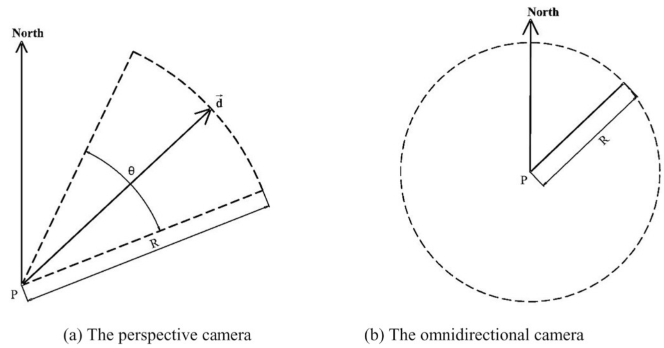

Fig. 2. 2D coverage model of the camera.

The omnidirectional cameras could cover the complete area within a certain distance when the monitoring area is unobstructed. On the two-dimensional plane, the omnidirectional camera theoretical field of view is a circular area with the camera position as the center (P) and the visible distance as the radius (R) (see Figure2b). The perspective cameras are usually set up to a fixed orientation and angle of the surveillance area. Its planar field of view is a fan shape as Fig. 2a shows. The fan parameters include: (1) P is the fan vertex, i.e., the camera position; (2) R is the fan radius, i.e., the farthest visible distance in the video image; (3) d is the fan azimuth angle, i.e., the angle from the north direction to the principal optic axis in a clockwise direction; and (4) θ is the fan angle, i.e., the horizontal angle for the camera FOV. It can be seen that in the process of video surveillance optimization, both types of cameras need to infer their mounting position and visible distance. In addition, it is also necessary to determine the orientation and horizontal viewing angle of the perspective cameras.

The surveillance cameras in an area are usually mounted in a fixed position with higher visibility. In GIS, Viewshed analysis is often applied to calculate the visible area of the point. There are a number of traditional viewshed algorithms including inter-visibility based on the LOS (line of sight), the reference plane algorithm, and the Xdraw algorithm, etc. (Floriani & Magillo, 2003). The data used in these algorithms is often the terrain models such as digital elevation model (DEM). However, it is the issue that there is mainly occlusion between the buildings in the urban surveillance area. We utilize the shadow-casting algorithm by the line of sight for resolving the buildings occlusion problem to calculate the cell viewshed. The shadow-casting algorithm is used to compute the dungeon area that is visible to the player in the roguelike game (Lippert, 2011). In our study, the surveillance area is rasterized with cells, and the buildings are marked as obstacle cells. To overcome the problem with the multiple visits of the same cells, the shadow-casting recursion algorithm is used to find visible cells by the line of sight (LOS) method. The algorithm divides the plane into eight octants and scans cells row-by-row or column-by-column to find visible cells. When the scan encounters a blocked cell, the new scan process is recursively initiated in the next row or column of the current row or column. During each scan, the cell visibility is determined by the start LOS and end LOS (Bergström, 2017). By execution of the shadow-casting recursion, the viewshed cell set of the specific location is generated within the max visible distance.

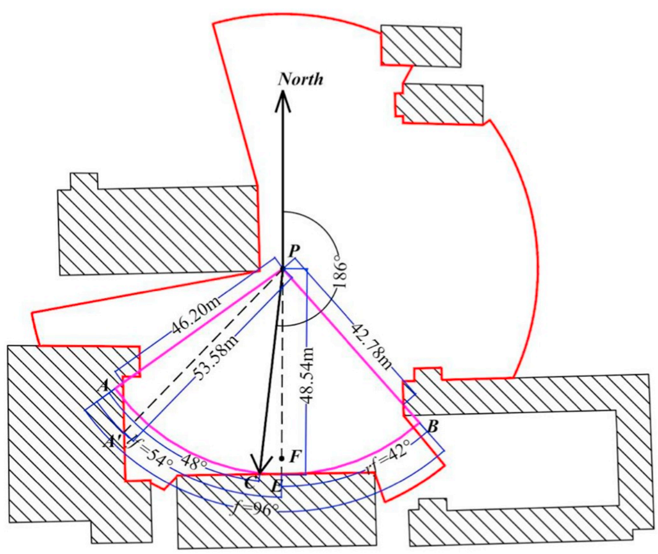

Fig. 3. The fan coverage inference algorithm example.

Based on cell viewshed calculation, the probability of cell monitored is proposed to indicate the cell visibility. It is defined as the probability that each cell in the area is monitored by other cells. The formula is as follows:

$$P(g \mid D) = c_g/n_D,\ c_g = \sum_{i=1}^{n} v_i | g \in VS_i,\ i \neq g \tag{1}$$

where $P(g|D)$ is the probability of cell monitored, c_g is the cell counts where the cells are covered by the other cell's viewshed VS_i, n_D is the total cell numbers of the surveillance area, and v_i is the value of cell g in VS_i, if g is visible then $v_i = 1$, else $v_i = 0$. The probability of each cell monitored in the area is calculated using Formula (1), and then converted into point data as *PPs* (Probability Points). Each cell's viewshed VS_i is converted into viewshed polygon as *VPs* (Viewshed Polygons). These two data are used as the input data of camera coverage inferring procedure.

3.2. Camera coverage inferring

Following the definition of the coverage model and the viewshed analysis, the camera coverage parameters including camera location, FOV angle, orientation, and visible distance can be inferred, and the camera monitoring area is generated. We determine camera field of view parameters first, and then generate the coverage area or polygon using these parameters. For the omnidirectional camera, the camera location is the circle center (P) which could be selected from the *PPs* by comparing the probability of cell monitored. Once P is located and marked, a corresponding viewshed polygon, VP_i, is selected to calculate its roundness index using the formula $rn_i = 4s\pi/l^2$, where rn_i is the roundness index of the viewshed polygon, s is the polygon area, l is the polygon perimeter (Mathworks, 2018). If the rn_i is greater than 0.90, the object is circular in shape. After the circular viewshed polygon is selected, the camera coverage can be generated as the circle by buffering the P with max visible distance.

To infer the FOV parameters of the perspective camera such as fan vertex, angle, radius, and azimuth include more steps. The camera location is also the fan vertex (P) which could be determined by the same method of the omnidirectional camera. Then the important spot within the corresponding viewshed polygon should be identified by the static or dynamic objects in an area. In this study, we collected the POI and social network record data from the LBS service. A POI is a point location containing the information about name, address, etc., which can indicate the static objects such as entrance, gate and other hot spots. The social network record data, WeChat (similar to Twitter) data in this case, could uncover the dynamic objects such as people's activity hot spots. The POIs and WeChat records are points data, whose density can be estimated by the KDE analysis. To combine the two types of data, the raster weighted overlay method is used by normalizing the KDE values. The formula is as $kv_{all} = \Sigma_{i=1}{}^{2} w_i v_i$, where kv_{all} is the combination of two KDE values, w_i is the weight, and v_i is the nondimensionalization KDE value. Then the raster data is converted into *CPs* which indicate the cell monitored importance. The CP_i could be marked using the spatial within relationship query method by the corresponding viewshed polygon VP_i. Upon the completion of the above steps, the fan parameters could be inferred using Algorithm 1 as follows.

Algorithms 1: Perspective Camera Coverage Parameters Inference
Input: camera location *P*, viewshed polygon VP_i, the important cell CP_i, angle incremental step *a*
Output: coverage parameters including fan vertex, FOV angle, visible distance, azimuth yaw
Process:
Generate vector *z* from *P* to CP_i, record the azimuth angle of *z* as *y*
Intersect *z* and VP_i, obtain the intersection *M*
Calculate distance *R* from *P* to *M* as visible distance
For *ii* in (0, 90/*a*) step 1
Chang angle from *y* by add(as q_1) and subtract(as q_2) the multiple of *a* with *ii*
Find the intersection N from *P to* CP_i *along with angle* q_1 *and* q_2
Calculate distance *d* from *P* to *N*
If d < R then record the angle *q* and break
Calculate the azimuth yaw by the angle bisector between q_1 *and* q_2
Calculate the FOV angle by the discrepancy between q_1 *and* q_2

As can be seen from the algorithm above, the visible distance *R* is determined by the intersection of the viewshed polygon boundary and the vector *z* from camera location *P* to the important point CP_i within the viewshed polygon. The two angles, q_1 and q_2 are calculated by increment or decrement according to a certain step *a* starting with the azimuth of vector *z and* ending with increment or decrement of 90°. The distance *d* between *P* and intersection calculation is similar to the above. The angle expansion is broken where the *d* is less than R by taking into account the occlusion of the building to the viewshed. The FOV angle and the camera orientation is determined by the discrepancy and bisector between q_1 *and* q_2. Fig. 3 shows the algorithmic process, where $R = 48.54$ m, *P* is the camera location, *F* is the point *CPi* with max kv_{all} in the viewshed (red lines) of cell *P*, *lf* is the angle *q1*(54°), *rf* is the angle *q2*(42°), and the angle step *a* is 2°. The inferred fan FOV angle $f = 96°$, and the camera orientation yaw is the direction of angle *f* bisector (186°). Once the perspective camera parameters had been decided upon, the fan coverage of the camera could be generated as the Fig. 3 shows (magenta lines) by the Bresenham line and circle drawing algorithm (Hughes et al., 2014).

The candidate camera inference process is shown in Fig. 4. After the probability of cell monitored (*P*) of all the cells are calculated and converted to PPs, the PP_i with the largest *P* value in all PPs is selected, and the viewshed polygon (VP_i) of the PP_i is retrieved. Then the point IP_i with the max *kv* value covered by the VP_i is determined (Fig. 4a). The roundness index of VP_i (rn_i) is calculated. If rn_i is greater than 0.9, the PP_i is the candidate position of the omnidirectional camera. We can get the intersection point of the line from PP_i to IP_i and the VP_i, and the distance from PP_i to the intersection point is the circle radius. We delete all PPs and IPs covered by the circular FOV (Fig. 4b). Otherwise, the point is the candidate position of the perspective camera, on the basis of the PP_i, IP_i, and VP_i (Fig. 4c), the parameters such as the radius, azimuth, and angle of the fan FOV are inferred using Algorithms 1, and all the PPs and IPs contained in the fan FOV are deleted (Fig. 4d). The algorithm does not need to set the threshold. The selection of PP_i and IP_i is based on the rank of the monitored probability and *kv* value. By selecting the PP_i with the max *P* value and its corresponding VP_i from the remaining PPs, the IP_i with max *kv* value of the IPs covered by the VP_i is queried. This process is repeated until the PPs set is empty, and the full coverage of the surveillance area is completed, the candidate camera positions and parameters are inferred. In order to avoid local optimums, the PP outside the neighbourhood range of the specific PP_i is preferentially searched, and the coverage ratio is reduced to 90% in order to decrease the variable size of the optimizing model.

3.3. Camera coverage optimization

As the two types of camera coverage, circle and fan are generated, and the service area of cameras is determined subsequently. Then the camera placement could be optimized using the MCLP-CC model. The model is outlined in detail by Tong (2012) and is briefly summarized as follows:

$$\text{Maximize } \sum_i \lambda_i \tag{2}$$

Subject to

$$\lambda_i \le \sum_{j \in N_i} b_{ij} x_j \ \forall\, i \tag{3}$$

$$\lambda_i \le w_i \ \forall\, i \tag{4}$$

$$\sum_j x_j \le p \tag{5}$$

$$x_j \in \{0, 1\} \ \forall\, j \tag{6}$$

where *i* is the index for demand unit, i.e., the monitored point, cell or polygon, *j* is the index for potential camera locations, w_i is the expected service at point *i*, *p* is the number of cameras to site, λ_i is the total number of service received by *i*, the b_{ij} is the amount of service provided to *i* by the camera *j*, and N_i is the set of the cameras that are able to provide some service to *i*.

To obtain the parameters of the MCLP-CC method, the demand unit should be determined first. As Tong and Wei (2017) suggested that the partial coverage should be considered. Thus, the difference overlay is executed with the surveillance area polygon and the building footprint polygon to get the demand region for camera surveillance. The fishnet polygon of rectangular units for the demand region is created by the rectangle length and width parameters. The fishnet polygons are taken as the continuous demand unit, which shows the complete or partial coverage by the facility service area. When determining the demand unit and inferring the coverage circle or fan polygon of the camera, the polygon intersecting overlay is calculated to model MCLP-CC and obtain the parameters of MCLP-CC method including the total number of services received, the amount of service provided by the camera, the cameras set that can provide some service to demand, etc. The final stage of the method is the MCLP-CC calculation and the camera coverage optimization solution founding. As a result, the camera is configured with higher coverage ratio, lower overlap rate, and occlusion

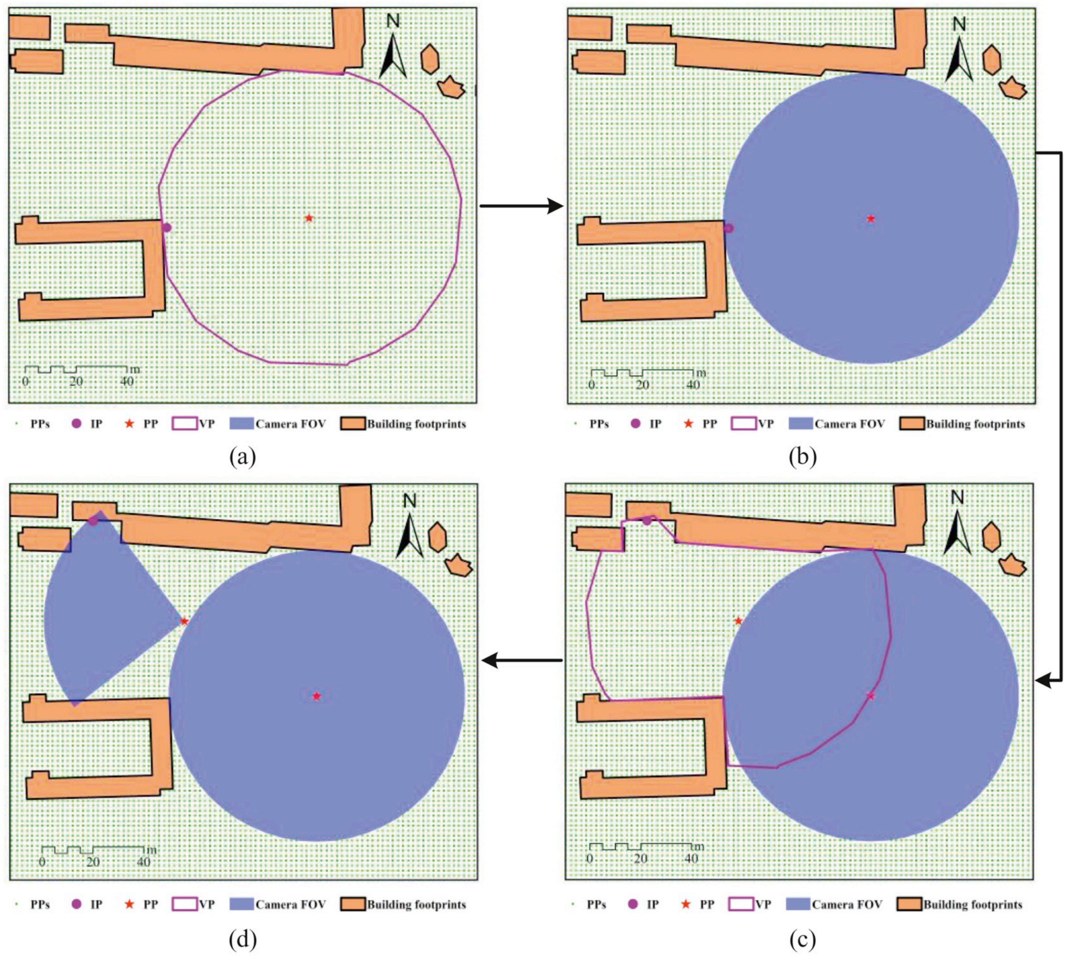

Fig. 4. The example of the potential camera inference process.

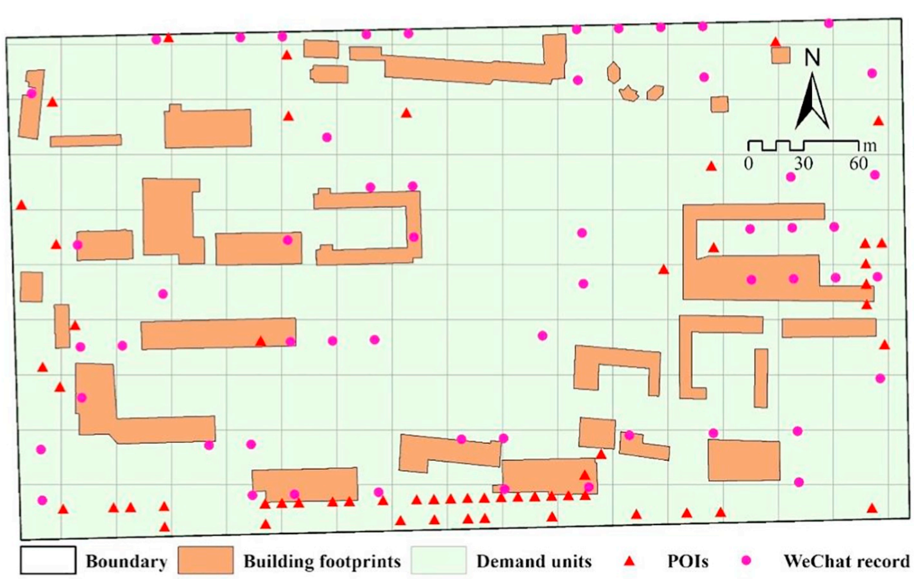

Fig. 5. Case study area with POIs, WeChat records and demand units shown.

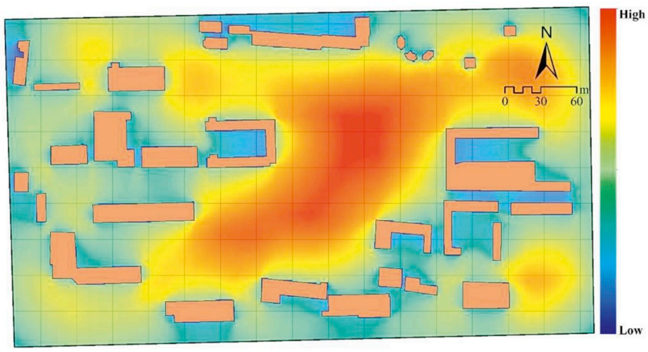

Fig. 6. The probability of cell monitored.

rate, and detailed parameters including location, FOV angle, visible distance, and camera orientation.

4. Experiment and results

4.1. Data collection and processing

We used the building footprint data, POI data, and social network record (WeChat) data from online web services for LBS including Baidu Map and Tencent Easygo platform. The building footprints indicate the building area in 2D plan map, which can be collected through the visual interpretation of high-resolution remote sensing images. Recently, these building footprints data in a city have already been collected and uploaded to the online web service for LBS such as Google map, Baidu map, and OpenStreetMap. We used the building footprint data from the Baidu map. For POI data in the surveillance area, the LBS provided the web service Application Programming Interface (API) to retrieve the POI data near the user-defined location. We used Place API from Baidu Map Open Platform (http://lbsyun.baidu.com/) to record the POI data in study area. The fourth dataset was the real-time density information for WeChat user. The data is recorded and visualized on the Tencent ‘Easygo’ Open Big Data Platform for the public with the real-time crowdedness. As the largest social network platform of China, the WeChat data could indicate that the user activities hot spot for the dynamic object in the area. We implemented the web crawler that uses an Easygo API (https://heat.qq.com/heatmap.php) to collect the WeChat data at base transceiver stations of smartphones and Wi-Fi hotspots covering the surveillance area at a specific time.

A case study in Zhengzhou, China were performed to demonstrate the application of the introduced camera coverage inferring and optimization method for the camera planning improvement and parameters configuration of cameras. The total area of the surveillance region that has buildings with different sizes is $0.1324\,km^2$. The total building footprint area is $0.0197\,km^2$; thus the demand area for monitoring is $0.1127\,km^2$. To get the demand units and reduce the optimization

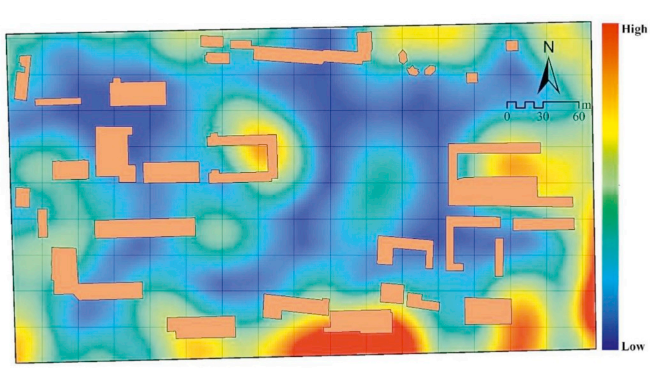

Fig. 7. The weighted overlay of POI and WeChat KDE values.

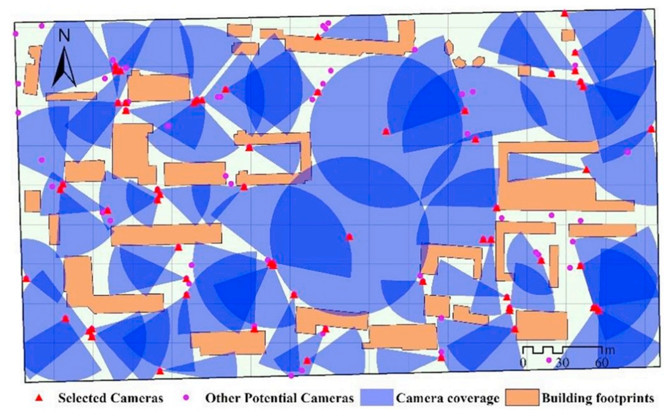

Fig. 8. The location and coverage of selected cameras and potential cameras position with 80.02% coverage.

variable counts, we divided the surveillance region into 170 squares sub-polygons with $30 \times 30\,m$ size. We gathered the WeChat user crowdedness data and POI data at 12:00 am, December 14, 2017, in order to fully show the people's activities. To weaken the edge effect of KDE, the surveillance area is expanded by half of its short side length (273 m) to select POIs and WeChat records points. There are 639 POIs and 242 WeChat records in the extended area(74 POIs and 55 WeChat records in the surveillance area, see Fig. 5).

4.2. Model implementation

In the follow-up phase of the experiment, the candidate camera's location and coverage area must be inferred by the above dataset. We rasterized the surveillance area into raster cell with $2 \times 2\,m$ size to refine the viewshed area, and marked the buildings cell as code 0 and the non-buildings cell as 1. Each cell viewshed area is generated using the shadow-casting algorithm with max visible distance 60 m. The distance is the farthest visible distance on the cell, which the current monitoring camera can identify the target in the image. According to the occlusion of the camera FOV, the inferred distance is less than or equal to 60 m. The probability of cell monitored is calculated as shown in Fig. 6. To obtain the important spot in the cell viewshed, the KDE values of POIs and WeChat records are calculated separately using the bandwidth value of 50 m. The POI data indicates the critical location of static targets such as buildings, and the WeChat data uncover the hotspot of the dynamic targets such as crowd. Both of them should be the critical targets for surveillance and we consider they have the same level of importance. As a result, these two KDE values are overlaid with the same weight of 0.5 (see Fig. 7). If we increase the weight of the POI or the WeChat record, the orientation parameters of a single camera will be altered; however, the overall coverage of multiple cameras will be changed little. The cell viewshed area, cell monitored probability, and the combination of KDE data is converted into polygons and points. Based on these data, we utilized the coverage inferring method in Section 3.2 to generate the candidate camera location and its coverage circle or fan. To eliminating the FOV with a small coverage area and a large overlap portion, the coverage ratio of 90% is set as the break condition of the loop process in the algorithm. In this case, a total of 107 candidate camera positions were generated (Fig. 8) with a coverage area of $0.1016\,km^2$.

The MCLP-CC was modeled by overlaying the candidate camera coverage (service area) and the demand units using areal interpolating. We developed the procedure using the Python language and the ArcPy library in ArcGIS to formulated the MCLP-CC model. The model was solved using Gurobi linear programming solver. We performed the calculation on an Intel Core i7 3520M CPU @ 2.90 GHZ with 16.0 GB of random access memory.

Table 1
Optimization process of different counts of cameras.

Camera numbers	Coverage ratio(%)	Objective (m^2)	Iterations	Time (s)	Camera numbers	Coverage ratio(%)	Objective (m^2)	Iterations	Time (s)
1	10.00	11,276	192	0.0070	32	64.95	73,200	260	0.0220
2	17.75	20,011	198	0.0070	33	66.25	74,666	267	0.0220
3	20.43	23,023	212	0.0120	34	66.51	74,968	287	0.0240
4	22.99	25,916	203	0.0170	35	66.79	75,283	294	0.0220
5	26.02	29,331	205	0.0160	36	67.42	75,987	291	0.0250
6	28.58	32,216	209	0.0150	37	68.55	77,260	283	0.0260
7	30.87	34,798	225	0.0160	38	68.55	77,260	284	0.0240
8	33.24	37,463	220	0.0190	39	69.24	78,045	295	0.0350
9	35.40	39,900	208	0.0150	40	69.98	78,874	314	0.0230
10	37.52	42,290	222	0.0160	41	71.13	80,166	296	0.0260
11	39.46	44,472	232	0.0160	42	71.06	80,096	304	0.0230
12	40.96	46,166	223	0.0250	43	71.81	80,934	299	0.0230
13	42.91	48,361	206	0.0140	44	72.30	81,486	295	0.0230
14	44.58	50,247	214	0.0150	45	72.63	81,858	310	0.0250
15	46.17	52,035	207	0.0150	46	72.95	82,221	316	0.0240
16	49.28	55,548	211	0.0140	47	73.94	83,334	296	0.0220
17	49.43	55,713	206	0.0150	48	73.94	83,334	289	0.0220
18	50.95	57,429	207	0.0140	49	74.11	83,529	311	0.0230
19	52.31	58,958	218	0.0160	50	74.41	83,870	305	0.0270
20	53.46	60,250	233	0.0180	51	74.76	84,257	302	0.0270
21	56.51	63,691	222	0.0170	52	75.42	85,007	286	0.0230
22	56.27	63,424	228	0.0170	53	75.70	85,325	300	0.0270
23	57.43	64,724	237	0.0140	54	76.32	86,022	306	0.0210
24	58.19	65,580	245	0.0200	55	76.32	86,018	314	0.0320
25	59.11	66,622	258	0.0180	56	77.13	86,931	331	0.0700
26	60.25	67,910	251	0.0190	57	78.74	88,746	364	0.0450
27	61.19	68,962	256	0.0200	58	77.69	87,566	308	0.0390
28	61.56	69,387	258	0.0210	59	79.02	89,061	390	0.0450
29	62.62	70,575	261	0.0200	60	79.46	89,557	314	0.0650
30	63.72	71,814	268	0.0260	61	79.56	89,668	322	0.0370
31	64.23	72,392	265	0.0300	62	80.02	90,186	369	0.0440

4.3. Results

To reduce the optimization problem size, the amount of cameras is set between 1 and 107 given that with 107 cameras 90.15% of the area can be covered. The 2 circles and 105 fans are generated using the Python scripts and ArcGIS ArcPy package. We executed the optimization process using Gurobi solver by setting the different camera number. As Fig. 8 shows, the 2 circles and 60 fans are configurated and drawn on the map which the coverage ratio of the surveillance area is 80.02%. The optimization results with different camera numbers are reported in Table 1.

To evaluate the result, the coverage, occlusion, and overlapping ratio are defined in Formulae (7)–(9):

$$CAR = \frac{Area(\cup Inner(CP_i)) - Area((\cup CP_i) \cap (\cup BF_j))}{(S_a - S_b)} \tag{7}$$

$$COR = \frac{Area((\cup Inner(CP_i)) \cap (\cup BF_j))}{(S_a - S_b)} \tag{8}$$

$$CVR = \frac{(\sum_i S_{CP_i} - Area(\cup Inner(CP_i)))}{Area(\cup Inner(CP_i))} \tag{9}$$

where *CAR* is the coverage ratio of the cameras, *COR* is the occlusion

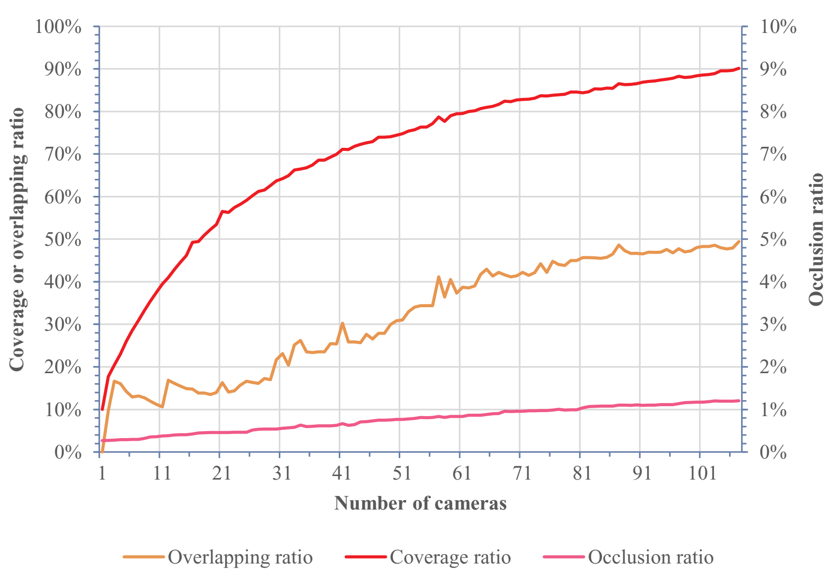

Fig. 9. The overlap (*CVR*), coverage (*CAR*), and occlusion (*COR*) ratio graph for different number of cameras.

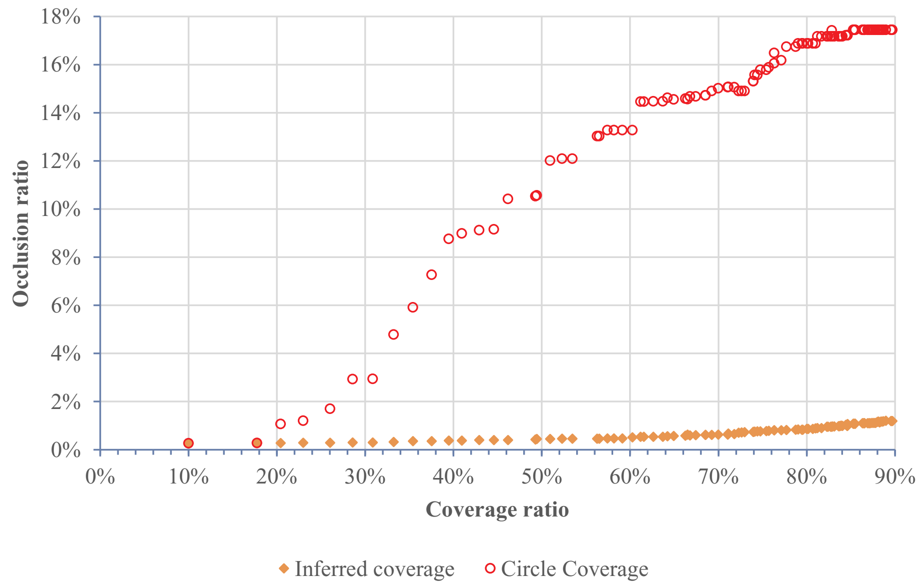

Fig. 10. The occlusion ration of two types of coverage with the same coverage ratio.

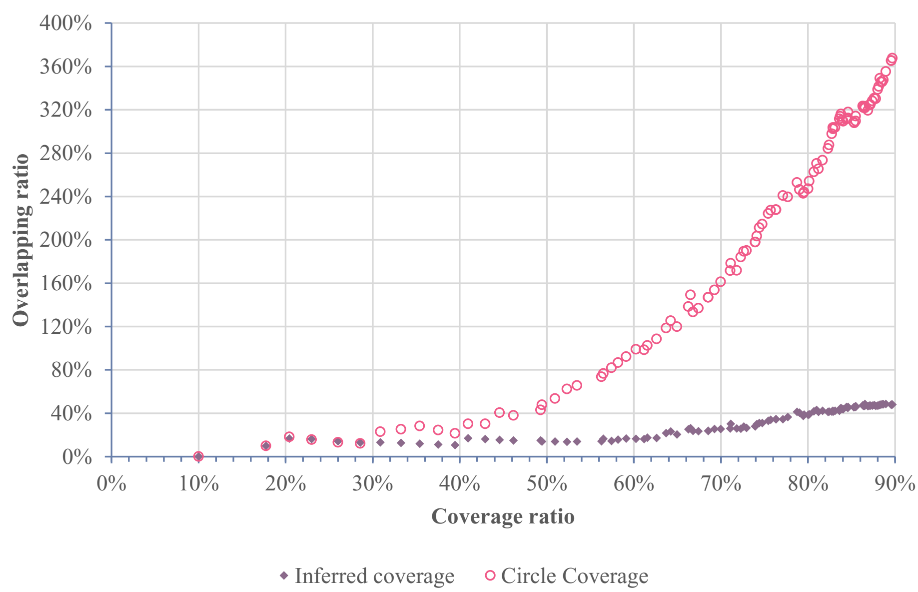

Fig. 11. The overlapping ration of two types of coverage with the same coverage ratio.

Table 2
Statistical results of *CAR*, *COR*, and *CVR* for each coverage type.

Coverage type	*CAR* (%)				*COR* (%)				*CVR* (%)			
	Ave	Min	Max	Dev	Ave	Min	Max	Dev	Ave	Min	Max	Dev
Inferred coverage	70.16	10.0	90.15	18.96	0.77	0.27	1.20	0.28	31.97	0	49.43	13.43
Circle coverage	88.54	10.0	99.71	18.90	14.43	0.27	17.44	4.34	205.48	0	381.42	114.86

ratio of camera coverage by the buildings, *CVR* is the overlapping ratio of the cameras, CP_i is the coverage fan or circle of camera *i*, BF_j is the *jth* building footprint polygon, S_a, S_b, and S_{CP_i} is the area of the surveillance region, building footprints polygon; The coverage fan or circle of camera *i*. *Inner* and *Area* are the functions to obtain the coverage polygon CP_i inside the surveillance region and its area value. It can be illustrated from Fig. 9 that as the number of cameras grows, the overall trend of *CAR*, *COR*, and *CVR* value increase synchronously. Their max values are 90.15%, 1.20%, and 49.43% respectively as the number of cameras is 107. It is apparent that the *COR* is very small, and what stands out in this figure is the rapid growth of the *CAR* but the slowly increasing of the other two ratios, especially the *COR*. It is the inevitable result that our method defined the two types of coverage model including circle and fan instead of the circle only and utilized the multi-source LBS data including building footprints, POI, and WeChat data to infer the camera coverage.

We used the same camera location and visible distance parameters and calculate the ratio of the circle coverage models, i.e., all the

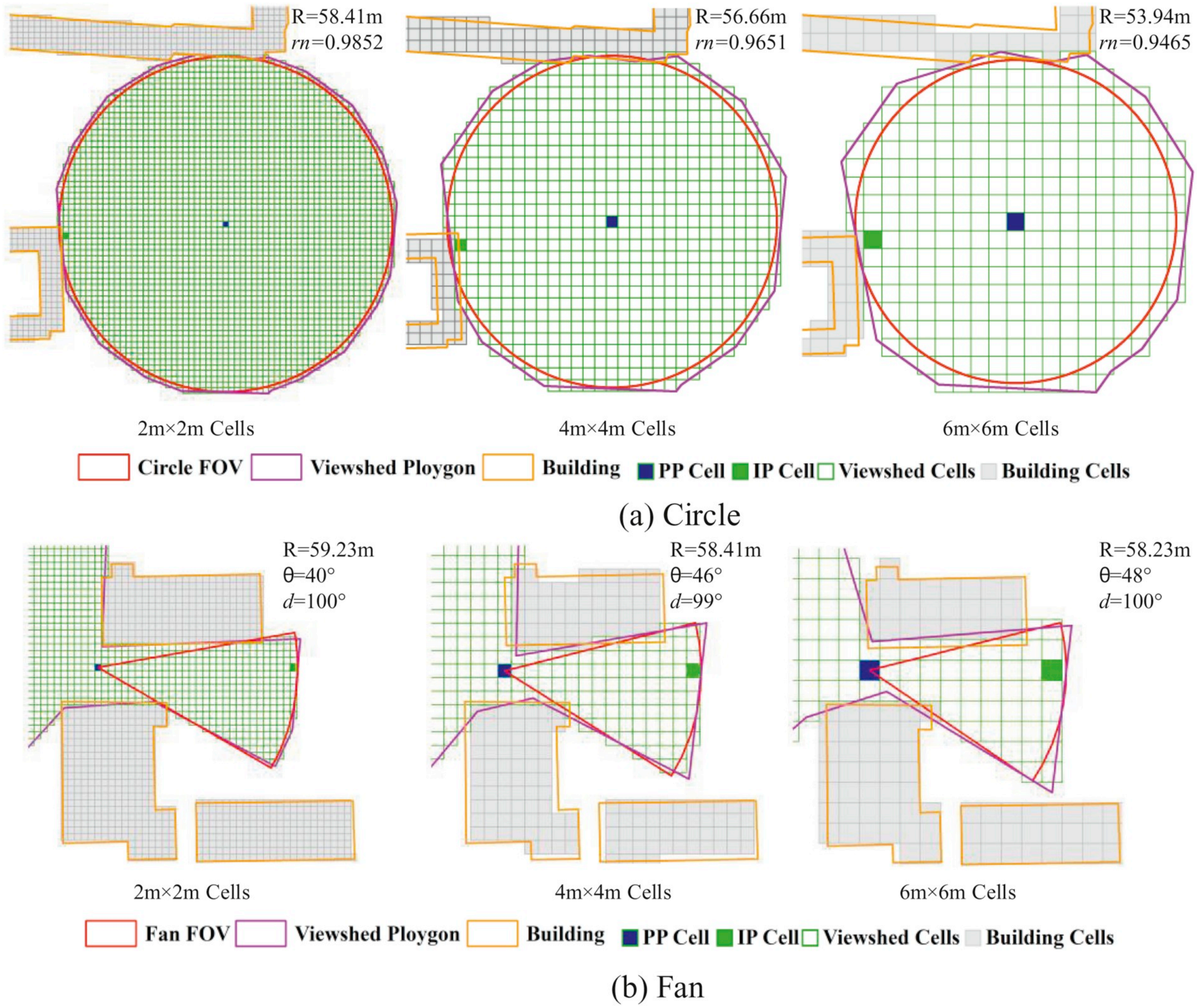

Fig. 12. The inferred FOV and parameters with different cell size.

cameras are the omnidirectional camera. Then we compared the results with our inferred coverage. The graphs in Figs. 10 and 11 show a significant discrepancy between the *COR* and *CVR* values of two types of camera coverage with the same *CAR* value. The trend of *COR* and *CVR* is same, i.e., as the value of circle coverage grow rapidly, the value of inferred coverage increases slowly. Table 2 presents the summary statistics for *COR* and *CVR* of two types of coverage. There are 13.66% and 16.24% discrepancy between average and max occlusion ratio, and 173.51% and 331.99% discrepancy in average and max overlapping ratio. The *CAR* is lower than circle coverage, and the main reason behind it is there is a multitude of occlusion areas by the buildings with circle coverage. This indicates that the overall occlusion and overlapping ratio of the inferred coverage reported significantly less than the circular coverage with the same numbers of cameras.

5. Discussions

5.1. Coverage inference with different cell size

In this study, the line-of-sight algorithm is applied to find the viewshed by discretizing the surveillance area into grid cells. The cell size is a key parameter, which may cause MAUP problems and produce different optimization results. To analyze the effect of the raster cell size on the camera coverage inference and optimization, we infer the camera parameters with three raster cell sizes such as 2 m, 4 m, and 6 m. From the perspective of a single camera, as the cell size increases, the inferred camera parameters change accordingly. (1) FOV for omnidirectional cameras. The expansion of the cell size will result in a reduction of the roundness index of the view polygon. As shown in Fig. 12a, at 2 m × 2 m, the roundness index is 0.9852, and as the cell size is 4 m and 6 m, it is 0.9651 and 0.9465, respectively. The inferred radius of the circular FOV is also gradually reduced, which is 58.41 m, 56.66 m, and 53.94 m, respectively, resulting in a gradual decrease in the coverage area. (2) FOV for the perspective cameras. The enlargement of the cell size causes the counts of cells covered by building to be reduced, which will increase in the fan angle. As shown in Fig. 12b, with the cell size from 2 m, 4 m, and 6 m, the inferred fan angle increases from 40° to 46° and 48°, respectively. The fan radius also decreased from 59.23 m to 58.41 m and 58.23 m. As the difference of the cell centre, the fan azimuth also changes slightly. Similar to the circular field of view, although the fan angle is increasing, the fan radius is gradually reduced, the coverage area is decreased from 1775.66 m^2 to 1726.83 m^2 and 1716.20 m^2. It can be seen that as the cell size expands, the location of selected PP and IP point changes slightly, and the viewshed error increases. The boundary of the inferred circular or fan FOV and the view polygon are more mismatched, which in turn leads to a reduction in the inferred coverage area.

The maximum coverage optimization was performed using the MCLP-CC model based on the inferred camera parameters. The difference of the average CAR, CVR, and COR between three cell size are highlighted in Fig. 13. With the expansion of cell size, the average CAR increased from 70.16% to 74.34% and 75.32% with 90% coverage.

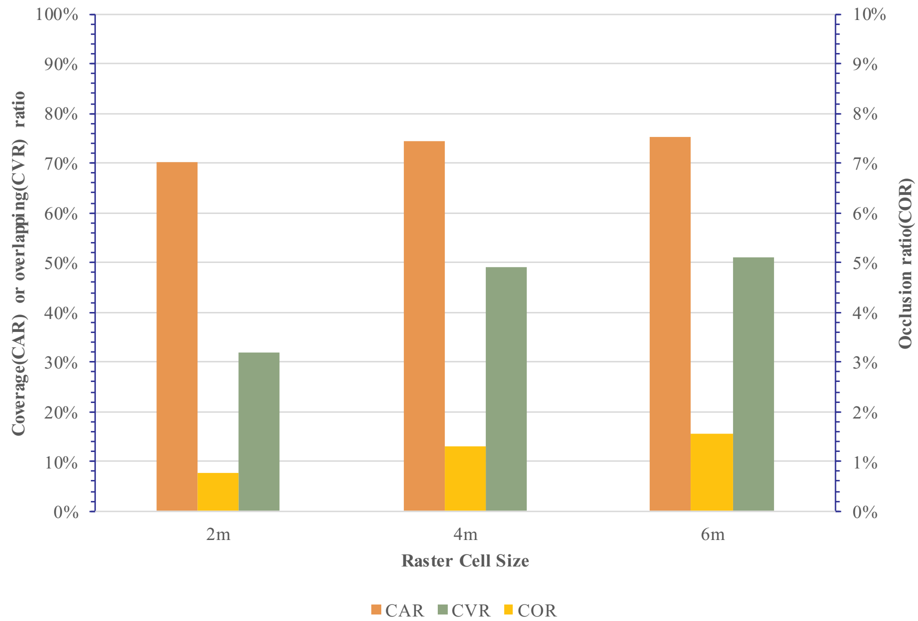

Fig. 13. The difference of average CAR/CVR/COR with three cell size.

Since the coverage area of a single camera is decreasing, more surveillance cameras are needed in the case of obtaining higher coverage. At 4 m and 6 m cell size, there are 136 and 150 cameras to cover the 90% of surveillance area. Thus, the CVR and COR are also increased, the average CVR is increased from 31.97% to 50.95%, and the average COR increased from 0.77% to 1.55%.

Comparison of coverage areas for optimization results with different cell sizes is based on the Generalized Intersection over Union (*GIoU*) ratio proposed by Rezatofighi et al. (2019). The *GIoU* is an indicator for evaluating detection errors in the field of object detection, which is defined in Formulae (10)–(11):

$$GIoU = IoU - \frac{Area(C - A \cup B)}{Area(C)} \tag{10}$$

$$IoU = \frac{Area(A \cap B)}{Area(A \cup B)} \tag{11}$$

where A, B is the polygon, which means the union coverage area of the optimized cameras with corresponding cell size, *IoU* is the intersection over union ratio of the A and B, C is the smallest convex hull that encloses both A and B. *GIoU* ranges from −1 to 1. It is an improvement of the *IoU*, which not only reflects the overlap area of the two optimization results but also describes the degree of separation. Fig. 14 illustrates the *GIoU* and *IoU* ratio of coverage area between 2 m and 4 m or 6 m cell size. In general, with the expansion of the cell size, the *GIoU* is lower when the number of cameras is less than 10, and the minimum values are 0.08 and − 0.09 when the cell of 4 m and 6 m is compared with the cell of 2 m. In the case of an increasing number of cameras, *GIoU* has shown a steady increasing trend. At 107 cameras, the *GIoU* is 0.59 and 0.60 at 4 m versus 2 m and 6 m versus 2 m. With a small number of

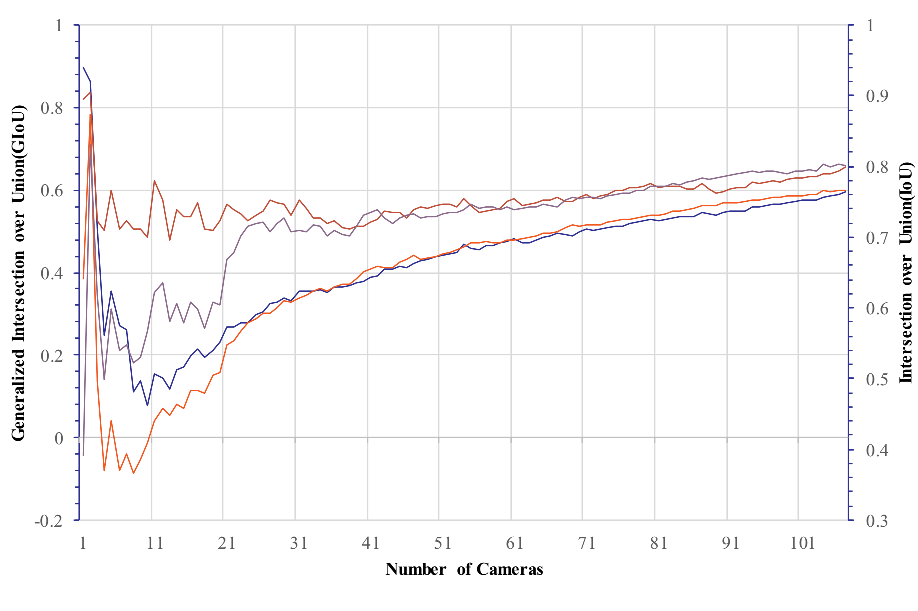

Fig. 14. The *GIoU* and *IoU* graph with different cell size. *GIoU*1 and *IoU*1 is 4 m × 4 m versus 2 m × 2 m , *GIoU*2 and *IoU*2 is 6 m × 6 m versus 2 m × 2 m.

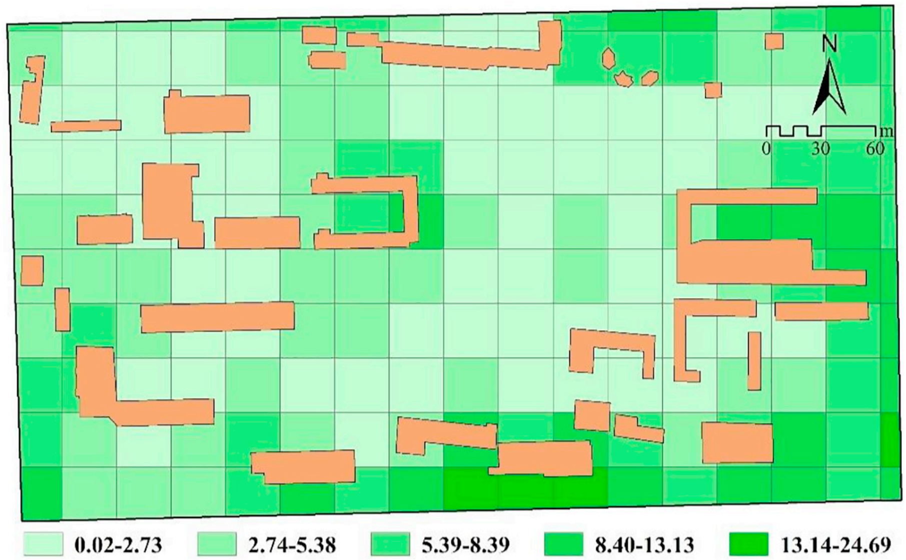

Fig. 15. The average weighted POI and WeChat KDE values of the demand units.

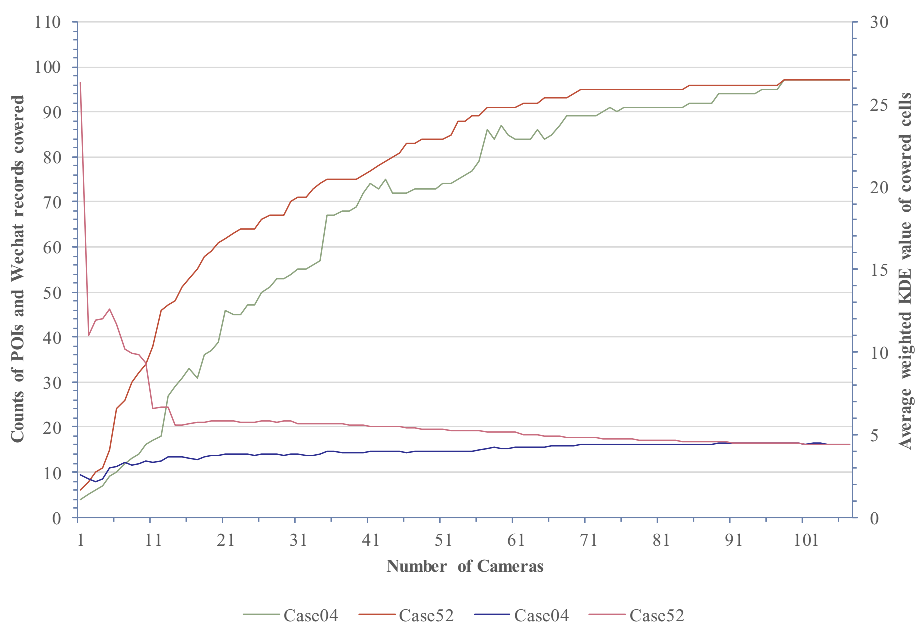

Fig. 16. The weighted POI and WeChat KDE values and the counts of POIs and WeChat records covered.

cameras, the *IoU* of the optimization result is smaller as the cell is enlarged. For example, for a single camera, the *IoU* with a 4 m versus 2 m and 6 m versus 2 m is 0.89 and 0.39 respectively. However, as the number of cameras increases, the *IoU* tends to be stable. At 107 cameras, the *IoU* of the two sizes of cells has reached 0.80. Generally, the larger the cell size, the smaller the *GIoU* and *IoU* are when the camera number is small. It is shown that different cell sizes will affect the coverage optimization results. The smaller the number of cameras, the more obvious the difference, and the larger cells will result in a lower *GIoU* ratio; as the number of cameras increases, the optimization results polygons are constantly increasing in overlapping area and their degree of separation is decreasing.

5.2. Coverage optimization with different importance of the demand units

From the point of maximum coverage, camera planning is the process to optimize the candidate camera based on its coverage area; all of the demand units are equally important. In this study, the key locations of the surveillance area are uncovered from the POI and WeChat data. It also reflects the importance of different locations to some extent in the surveillance area. To reflect the importance of different demand units in the optimization process, the average weighted KDE, i.e. kv_{all} value of the cells covered by each demand unit is calculated (Fig. 15). The camera coverage optimization is performed through the average weighted KDE value as the weight. The comparison between the optimization result (Case52) and Section 4 result (ignoring the importance of the demand unit, Case04) is shown in Fig. 16.

In the case of the POI and WeChat records covered with the same camera counts, the number of Case52 is higher than Case04, the minimum of the former is 6, and the mean is 77; while the latter has a minimum of 4, and the mean is 69. The average difference between the two cases is 8. It can be seen that case 52 is significantly different from case04 in the average kv_{all} values of cells covered by different numbers of camera. In Case 52, this value generally shows a downward trend, and in the case of a small number of cameras, the cells that have the higher KDE Value are preferred in the optimization process. For

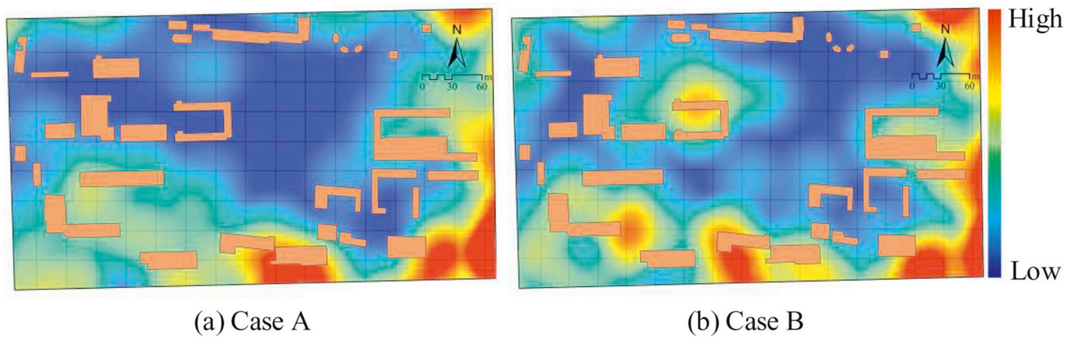

Fig. 17. The combination KDE of POIs and WeChat records at different times.

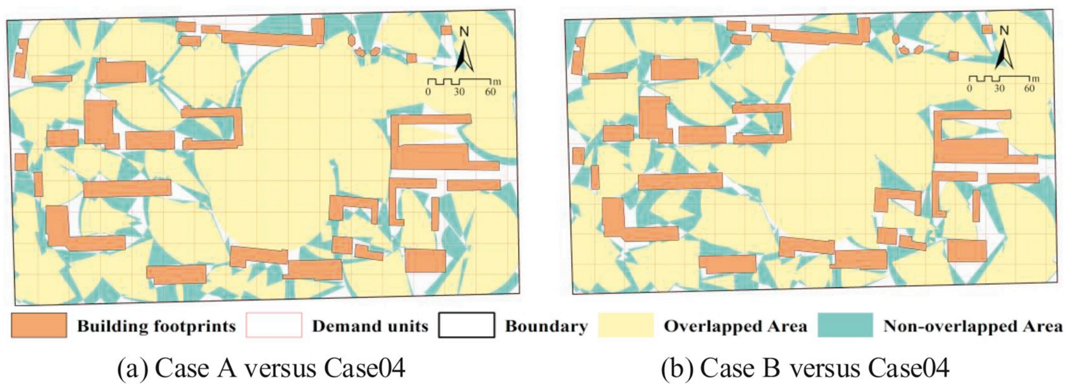

Fig. 18. The coverage area comparison of the Case A and Case B versus Case04 with 80% coverage.

example, in 1 camera, it covers the demand unit with the highest average kv_{all} value. The mean value of the covered cells is 26.32. As the number of cameras increases, the value becomes smaller and tends to be stable.

The result of Case04 is just the opposite. In general, the average kv_{all} values of cells covered by the camera shows a slight upward trend. The smaller the number of cameras, the smaller the average kv_{all} value; but the increasing step is small. As the importance of the demand unit is considered in the optimization, the average coverage ratio of the optimization result is also reduced from 70.16% of Case04 to 60.61% of Case52; when there is only one camera, the coverage is decreased from 10.00% to 0.55%. The average coverage ratio, the numbers of POI and WeChat records covered, and average kv_{all} value tend to be consistent as the number of cameras increases. It can be seen that modeling the importance of the demand unit will have a significant impact on the optimization result when the number of cameras is small, but it is convergence when the number of cameras and coverage ratio is increased.

5.3. Coverage inference and optimization with different times of LBS data

In Section 4, considering the characteristics of crowd activities, the WeChat record data at 12:00 am on a weekday was selected for calculation. To analyze the influence of LBS data on the optimization results, the two other different moments WeChat record data were selected to infer the camera position and field of view and optimize the camera coverage. Case A selects the same time on the rest day, which is the WeChat record data of 12:00 am, December 16, 2017, for inference and optimization. A total of 290 WeChat records were obtained based on the expanded range (60 in the surveillance area). In order to analyze the results at different times on the same day, Case B selects the off-hours in the afternoon of the same day, which is the WeChat record data of 18:00 pm, December 14, 2017. There are 274 WeChat records on the expanded range (62 in the surveillance area).

The KDE were performed on the WeChat record points at two moments and overlapped with the POI KDE value. As shown in Fig. 17, the overall distribution of combination KDE at different times has similarities to some extent. The hotspot areas are distributed on the south and east sides of the area; however, the difference between the weekday and the rest day is more significant. According to the combination KDE of the two cases, the camera FOV inference and coverage optimization were carried out. In the case of coverage of 80%, it is generated 71 cameras (including 2 circles and 69 sectors, with a monitoring coverage of 80.15%) in Case A, and 61 cameras (including 2 circles, 59 sectors, the monitoring coverage rate is 80.30%) in Case B. The coverage area comparison of the two cases with the results of Section 4 (Case04) is shown in Fig. 18. It is apparent that Case A and Case B with Case04 have a higher overlapped area, and their *IoU* are 0.7806 and 0.7757, *GIoU* is 0.5211 and 0.5042 respectively. With the upward of coverage, the overlapped area gradually increases. It can be seen that the LBS data will change the parameters of a single camera, but the impact on the overall coverage is a bit small.

5.4. Coverage inference with simple geometric approach

As discussed in Section 3.1, the field of view for surveillance cameras include two types of circular and fan. In previous study cases, the viewshed polygon or fixed-angle camera are used for camera planning, and the optimization process locates only the positions of the cameras. (1) The viewshed polygon is an irregular polygon due to occlusion of buildings, which is different from the FOV of the surveillance camera. As shown in the left graph of Fig. 19, the omnidirectional camera is selected based on the viewshed polygon. There are 714 building cells and 701 occlusion cells in the circle FOV of the camera. Although the circle FOV covers these cells, they cannot be monitored by the camera. (2) The fixed-angle camera is used (middle graph of Fig. 19). As the fixed center angle with 70° of the fan FOV, there are 153 building cells and 190 occlusion cells in the three fixed-angle fan FOVs of the camera.

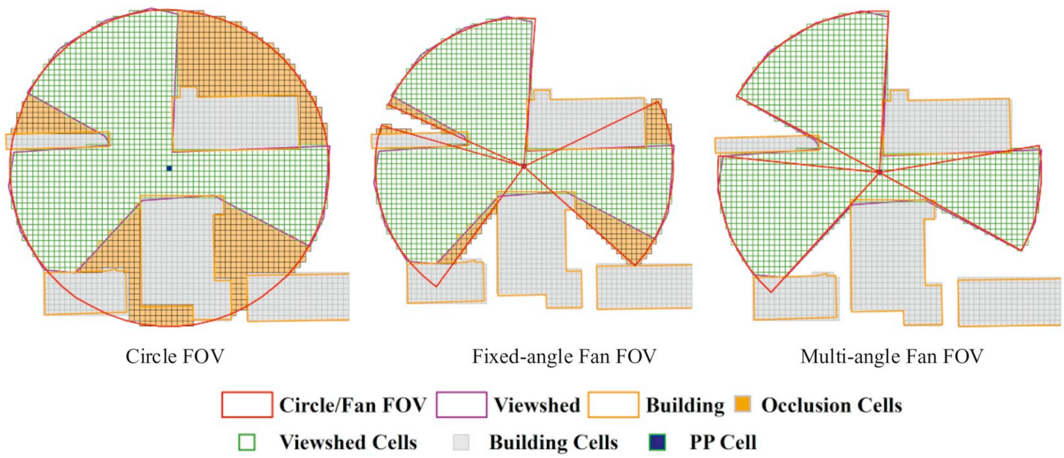

Fig. 19. The occlusion cells comparison of the circle, fixed-angle fan, and multi-angle fan FOV.

(3) Three cameras with 39°, 54°, and 65° center angles can be inferred according to the viewshed and LBS data using the proposed method in Section 3.2. The number of building units covered by the FOV is reduced to 9, and none of the cells are occluded (the right graph of Fig. 19). By applying building footprint data, POI data, and WeChat data, our method can infer many different types of monitoring cameras with detailed parameters such as camera location, FOV angle, orientation, and visible distance, which is beneficial to lessen the number of occluded areas and deployment costs.

5.5. Practical application discussions

The monitored probability is used to infer the candidate camera position in our method. The higher the probability that the cell is monitored, the wider the range of the field of view, where the deployed camera can monitor more areas and receive less occlusion. In actual surroundings, these well-viewed cells may be located in the center of the surveillance area, such as the center of an open square, and may be subject to managerial constraints if deployed directly at the inferred location. However, the method is based on cells for inference (cell area is 4m^2). For the sake of simplicity, the cell center point is selected when generating the camera position; but in fact, any position in the specific cell can be picked as candidate camera position. When deploying the camera, the camera can be deployed in the vicinity of the inferred position with the corresponding street lights, trees, etc., to free of the problem of physically impractical. At the same time, the deployment constraint can be further added to our method in the inference process to exclude the cell that cannot be deployed.

The camera parameters are unknown in the proposed method. These parameters are directly inferred based on the buildings, POIs, and WeChat records in the surveillance area. In practical applications, one of the possible scenarios is that there are corresponding types of surveillance cameras, and the parameters such as the angle and the monitoring distance are fixed. The method can also meet this camera planning requirement. According to the current primary surveillance camera type, there are seven kinds of camera FOV angles: 10°, 20°, 26°, 39°, 50°, 70°, 360° and six kinds of monitoring distances: 10 m, 20 m, 30 m, 40 m, 50 m, 60 m. After inferring the camera position, azimuth, and FOV angle used the method described in Section 3, the closest camera type combination could be selected from the above parameters. For example, if the calculated candidate camera has an angle of 98° and a radius of 51.74 m, two types of cameras can be picked as candidate cameras with a FOV angle of 70° and 26° and a radius of 50 m. In this way, the camera type parameters could be obtained, their FOV could be generated, and then the coverage optimization could be performed. As the narrowing of the camera's FOV angle, the number of cameras increased significantly to 179 with 90% coverage, and the corresponding average coverage decreased slightly to 68.12% compared with the results of Section 4. The overlap rate and the occlusion rate increased slightly, being 34.03% and 1.19%, respectively. However, despite that, the result is better than circular FOV. The method provides a solution for camera planning under the premise of known camera parameters.

6. Summary and conclusions

The coverage inference and optimization on the camera planning area continues to be a methodological challenge. In this paper, we introduce a camera coverage inference method using multi-source LBS data and optimize the camera placement with the MCLP-CC model to enhance the camera planning. The related issues such as the camera coverage model, the probability of cell monitored calculation, the camera coverage inference algorithm, and the camera coverage optimization are detailed. The method is discussed in depth from four aspects: cell size, demand unit importance, LBS data at different times, and practical applications. The advantages of the proposed camera coverage inference and optimization solution are summarized as follows. In the first place, the detailed camera parameters including camera location, FOV angle, orientation yaw, and visible distance instead of single camera location or fixed type cameras are inferred using the multi-source LBS data. Based on these parameters, users can configure the different type of cameras to achieve the surveillance task. There is one more point that the vital spot of an area which is reflected by the static (building footprints) and dynamic objects (people's activities indicated by POI and social network record) are utilized in the camera coverage inference process. It is given rise to the lower occlusion ratio and overlapping ratio with the same coverage ratio. The last, but not the least, is that the data used in our method is easy to access which were collected from the online LBS web services. With the constraint of camera cost, the paper focuses on the use of LBS data to infer the camera parameters, and the MCLP-CC model is used to get the optimization scheme which generates the max coverage by few cameras to reduce the cost. For camera deployment that requires a specific overlap rate and without cost constraint, the method could be easily used for the overlapping coverage optimization by BCLP or MCLP-B (maximal covering location problem with backup) model (Church & Murray, 2018; Murray et al., 2007).

Further studies of the following two topics should be performed. First, the 3D camera coverage geospatial model should be investigated, and the general 3D coverage inference methods should be defined to describe coverage precisely more than just providing 2D FOVs. Modeling the 3D coverage of camera with 3DGIS will create a host of potential applications for solving real-world camera planning problems. Moreover, the camera height parameter should be considered for camera placement, which is omitted from our study. Another cardinal direction is to implement a GIS toolbox or web service interface for camera coverage inference and optimization process and evaluate the cost of the camera deployment solution. Meanwhile, the sensitivity analysis should be performed with the different weight settings for the overlay of KDE values of POIs and WeChat records.

We believe the method introduced in this paper, especially using the Location-based Service data, has application potentials in camera planning and deployment. Supported by data collection and processing methods of GIS, various innovative camera planning tools could be implemented for both video surveillance and the visual sensor network application.

Declaration of Competing Interest

None.

Acknowledgment

Special thanks go to the editor and anonymous reviewers of this paper for their constructive comments, which substantially improved the quality of this paper. This study was supported by the National Natural Science Foundation of China under Grant [grant numbers 41871316, 41601116]; the Key Project of Humanities and Social Sciences Key Research Base of the Ministry of Education, China under Grant [grant number 16JJD770019]; Science and Technology Foundation of Henan Province [grant number 172102110286]; and the Henan University Science Foundation for Young Talents.

References

Bergström, B. (2017). *FOV using recursive shadowcasting.* Retrieved*02-06*, 2018. http://www.roguebasin.com/index.php?title=FOV_using_recursive_shadowcasting.

Chrysostomou, D., & Gasteratos, A. (2012). Optimum multi-camera arrangement using a bee colony algorithm. *2012 IEEE International Conference on Imaging Systems and Techniques (IST)* (pp. 387–392). Manchester, UK: IEEE. https://doi.org/10.1109/IST.2012.6295580.

Church, R. L., & Murray, A. T. (2009). *Business site selection, location analysis, and GIS.* Hoboken, New Jersey: Wiley Online Library1–19.

Church, R. L., & Murray, A. T. (2018). Extended forms of coverage. *Location covering models: History, applications and advancements* (pp. 49–79). Gewerbestrasse, Switzerland: Springer.

Church, R. L., & ReVelle, C. (1974). The maximal covering location problem. *Papers of the Regional Science Association, 32*, 101–118.

Chvatal, V. (1975). A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory, Series B, 18*, 39–41.

Conci, N., & Lizzi, L. (2009). Camera placement using particle swarm optimization in visual surveillance applications. *2009 16th IEEE international conference on image processing (ICIP) (pp. 3485–3488)*Cairo, Egypt: IEEE. https://doi.org/10.1109/ICIP.2009.5413833.

Costa, D. G., & Guedes, L. A. (2010). The coverage problem in video-based wireless sensor networks: A survey. *Sensors, 10*, 8215–8247. https://doi.org/10.3390/s100908215.

Curtin, K. M., Hayslett-McCall, K., & Qiu, F. (2010). Determining optimal police patrol areas with maximal covering and backup covering location models. *Networks and Spatial Economics, 10*, 125–145. https://doi.org/10.1007/s11067-007-9035-6.

Dell'Olmo, P., Ricciardi, N., & Sgalambro, A. (2014). A multiperiod maximal covering location model for the optimal location of intersection safety cameras on an urban traffic network. *Procedia-Social and Behavioral Sciences, 108*, 106–117. https://doi.org/10.1016/j.sbspro.2013.12.824.

Feng, G., Liu, M., & Wang, G. (2014). Genetic algorithm based optimal placement of PIR sensors for human motion localization. *Optimization and Engineering, 15*, 643–656. https://doi.org/10.1007/s11081-012-9209-z.

Floriani, L., & Magillo, P. (2003). Algorithms for visibility computation on terrains: A survey. *Environment and Planning. B, Planning & Design, 30*, 709–728. https://doi.org/10.1068/b12979.

Fu, Y. G., Zhou, J., & Deng, L. (2014). Surveillance of a 2D plane area with 3D deployed cameras. *Sensors, 14*, 1988–2011. https://doi.org/10.3390/s140201988.

Gonzalez-Barbosa, J.-J., García-Ramírez, T., Salas, J., & Hurtado-Ramos, J.-B. (2009). Optimal camera placement for total coverage. *2009 IEEE International Conference on Robotics and Automation(ICRA) (pp. 844–848)*Kobe, Japan: IEEE. https://doi.org/10.1109/ROBOT.2009.5152761.

Hörster, E., & Lienhart, R. (2006). On the optimal placement of multiple visual sensors. *Proceedings of the 4th ACM international workshop on video surveillance and sensor networks* (pp. 111–120). Santa Barbara, California, USA: ACM. https://doi.org/10.1145/1178782.1178800.

Hughes, J. F., Van Dam, A., Foley, J. D., McGuire, M., Feiner, S. K., Sklar, D. F., & Akeley, K. (2014). *Computer graphics: Principles and practice.* New Jersey, USA: Pearson Education81–100.

Kim, K., Murray, A. T., & Xiao, N. (2008). A multiobjective evolutionary algorithm for surveillance sensor placement. *Environment and Planning. B, Planning & Design, 35*, 935–948. https://doi.org/10.1068/2Fb33139.

Lee, G., & Murray, A. T. (2010). Maximal covering with network survivability requirements in wireless mesh networks. *Computers, Environment and Urban Systems, 34*, 49–57. https://doi.org/10.1016/j.compenvurbsys.2009.05.004.

Lippert, E. (2011). Shadowcasting in C#. Retrieved 02-06, 2018 https://blogs.msdn.microsoft.com/ericlippert/2011/12/12/shadowcasting-in-c-part-one/.

Liu, J., Sridharan, S., & Fookes, C. (2016). Recent advances in camera planning for large area surveillance: A comprehensive review. *ACM Computing Surveys (CSUR), 49*, 6. https://doi.org/10.1145/2906148.

Liu, J., Sridharan, S., Fookes, C., & Wark, T. (2014). Optimal camera planning under versatile user constraints in multi-camera image processing systems. *IEEE Transactions on Image Processing, 23*, 171–184. https://doi.org/10.1109/TIP.2013.2287606.

Mathworks (2018). Identifying round objects. Retrieved 02-05, 2018, from http://www.mathworks.com/help/images/examples/identifying-round-objects.html.

Mavrinac, A., & Chen, X. (2013). Modeling coverage in camera networks: A survey. *International Journal of Computer Vision, 101*, 205–226. https://doi.org/10.1007/s11263-012-0587-7.

Morsly, Y., Aouf, N., Djouadi, M. S., & Richardson, M. (2012). Particle swarm optimization inspired probability algorithm for optimal camera network placement. *IEEE Sensors Journal, 12*, 1402–1412. https://doi.org/10.1109/JSEN.2011.2170833.

Murray, A. T. (2013). Optimising the spatial location of urban fire stations. *Fire Safety Journal, 62*, 64–71. https://doi.org/10.1016/j.firesaf.2013.03.002.

Murray, A. T. (2018). Evolving location analytics for service coverage modeling. *Geographical Analysis, 50*, 207–222. https://doi.org/10.1111/gean.12146.

Murray, A. T., Kim, K., Davis, J. W., Machiraju, R., & Parent, R. (2007). Coverage optimization to support security monitoring. *Computers, Environment and Urban Systems, 31*, 133–147. https://doi.org/10.1016/j.compenvurbsys.2006.06.002.

Murray, A. T., Tong, D., & Kim, K. (2010). Enhancing classic coverage location models. *International Regional Science Review, 33*, 115–133. https://doi.org/10.1177/2F0160017609340149.

Nam, Y., & Hong, S. (2014). Optimal placement of multiple visual sensors considering space coverage and cost constraints. *Multimedia Tools and Applications, 73*, 129–150. https://doi.org/10.1007/s11042-012-1266-y.

Pulver, A., & Wei, R. (2018). Optimizing the spatial location of medical drones. *Applied Geography, 90*, 9–16. https://doi.org/10.1016/j.apgeog.2017.11.009.

Rezatofighi, H., Tsoi, N., Gwak, J. Y., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union. *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2019). Long Beach, CA, USA: IEEE.

Tong, D. (2012). Regional coverage maximization: A new model to account implicitly for complementary coverage. *Geographical Analysis, 44*, 1–14. https://doi.org/10.1111/j.1538-4632.2011.00835.x.

Tong, D., & Wei, R. (2017). Regional coverage maximization: Alternative geographical space abstraction and modeling. *Geographical Analysis, 49*, 125–142. https://doi.org/10.1111/gean.12121.

Wang, M., Liu, X., Zhang, Y., & Wang, Z. (2017). Camera coverage estimation based on multistage grid subdivision. *ISPRS International Journal of Geo-Information, 6*, 110. https://doi.org/10.3390/ijgi6040110.

Wang, Q., Wu, J., & Long, C. (2013). On-line configuration of large scale surveillance networks using mobile smart camera. *2013 Seventh International Conference on Distributed Smart Cameras (ICDSC)*Palm Springs, CA, USA: IEEE. https://doi.org/10.1109/ICDSC.2013.6778238 1-6.

Wei, R. (2016). Coverage location models: Alternatives, approximation, and uncertainty. *International Regional Science Review, 39*, 48–76. https://doi.org/10.1177/2F0160017615571588.

Wong, D. (2009). The modifiable areal unit problem (MAUP). *The SAGE handbook of spatial analysis* (pp. 105–123). .

Xia, L., Yin, W., Dong, J., Wu, T., Xie, M., & Zhao, Y. (2010). A hybrid nested partitions algorithm for banking facility location problems. *IEEE Transactions on Automation Science and Engineering, 7*, 654–658. https://doi.org/10.1109/TASE.2010.2043430.

Xu, Y. C., Lei, B., & Hendriks, E. A. (2011). Camera network coverage improving by particle swarm optimization. *Journal on Image and Video Processing, 2011*, 3. https://doi.org/10.1155/2011/458283.

Yao, J., Zhang, X. X., & Murray, A. T. (2019). Location optimization of urban fire stations: Access and service coverage. *Computers, Environment and Urban Systems, 73*, 184–190. https://doi.org/10.1016/j.compenvurbsys.2018.10.006.

Zhao, J. (2011). *Camera planning and fusion in a heterogeneous camera network.* Lexington, Kentucky: University of Kentucky.