Cooperative Monitoring Scheduling of PTZ Cameras with Splitting Vision and Its Implementation for Security Surveillance

You-Chiun Wang[1,*], Ding-Yan Chen[1]

[1]Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan

Abstract

Pan-tilt-zoom (PTZ) cameras are widely used in surveillance systems. They are capable of remote directional control to direct the attention to objects. In the paper, we consider that PTZ cameras form a visual sensor network to monitor target objects. Each object is associated with a minimum monitoring time, where it has to be monitored by cameras in a period. As the monitoring quality of an object by a camera depends on the object's location in the camera's field of view (FoV), we propose a concept of splitting vision. Specifically, the FoV of a camera is sliced into three parts based on its viewing angle. A different weight is given when the object is in each part of FoV. Let the monitoring grade of a camera with respect to an object be the product of its weight, monitoring time, and the object's priority. Our problem asks how to schedule the rotation of cameras to monitor each object, such that the total grade is maximized, under the constraint of minimum monitoring time of objects. We develop an *efficient camera rotation scheduling (ECRS)* heuristic to support cooperative monitoring of objects by exploiting FoV overlap of cameras. Experimental results show that ECRS achieves a higher grade than other methods, which implies that it helps PTZ cameras provide better monitoring quality of objects with longer time. In addition, we implement a prototype system to realize the ECRS heuristic and also demonstrate the system for the scenario of security surveillance.

Keywords: Cooperative monitoring, PTZ camera, scheduling, surveillance, visual sensor network.

1. Introduction

Wireless sensor network (WSN) plays a key role in IoT (Internet of Things) [1], which helps people acquire data from the environment handily. A WSN comprises a large number of tiny but autonomous devices called *sensors*, each capable of detecting events, reporting its findings to a sink, and possibly reacting to events [2]. Owing to their flexibility and convenience, a variety of applications are developed for WSNs, for example, mobile surveillance [3], intelligent buildings [4], grocery shopping [5], health care [6], light control [7], and pollutant monitoring [8].

Recently, pan-tilt-zoom (PTZ) cameras are popularly applied to surveillance applications, which form a visual WSN to monitor target objects [9]. Due to its hardware nature, the sensing coverage of a PTZ camera, also known as *field of view (FoV)*, is directional. In particular, the FoV is often described as a sector with a radius (i.e., sensing distance) r_s and a viewing angle θ. Moreover, with the help of machinery (e.g., a stepper motor), a PTZ camera can rotate a lot horizontally, which allows it

* Corresponding author. E-mail address: ycwang@cse.nsysu.edu.tw
Tel.: +886-7-5252000#4323; Fax: +886-7-5254301

adaptively directing the attention to the monitoring objects. Thus, a PTZ camera can be treated as a *rotatable and directional (R&D) sensor*, whose rotation capability offers spatiotemporal monitoring of the environment [10].

In this paper, we are interested in studying how to schedule the rotation of PTZ cameras to make them cooperatively monitor a set of target objects and provide better monitoring quality. Specifically, the time axis is sliced into periods, during which each object needs to be monitored by at least one PTZ camera for a minimum time. Since the monitoring quality of an object with respect to a PTZ camera depends on the object's position in the camera's FoV, we propose a concept of *splitting vision*, where the FoV of a camera is divided into three parts by its viewing angle. A different weight is associated with each part. Then, we define the *monitoring grade* of a PTZ camera for an object as the product of its weight, monitoring time, and the object's priority. Our problem asks how to schedule the rotation and also staying time of PTZ cameras to monitor each object, such that the total grade can be maximized, under the constraint of minimum monitoring time of objects.

To solve the problem, we propose an *efficient camera rotation scheduling (ECRS)* heuristic, which first finds out possible FoVs for each PTZ camera based on the locations of objects. Among these candidate FoVs, ECRS iteratively picks one FoV to maximize the monitoring grade and decides the time that its camera stays in the FoV to monitor objects. Through simulation, we show that our ECRS heuristic achieves a higher monitoring grade than other methods, so it can provide better monitoring quality of objects with longer time. Moreover, a prototype system is also implemented to verify the practicability of ECRS, which can be used in the scenario of security surveillance.

The rest of this paper is organized as follows. The next section surveys related work. We formulate the PTZ camera scheduling problem and propose the ECRS heuristic in Sections 3 and 4, respectively. Then, Section 5 presents performance evaluation, followed by system implementation in Section 6. Finally, Section 7 concludes the paper and gives future work.

2. Related Work

2.1. Deployment Strategies for Omnidirectional WSNs

There have been many deployment strategies proposed for omnidirectional WSNs, as the event detection capability of a WSN highly depends on its distribution of sensors [11]. The work [12] divides the sensing field into grids and uses a simulated annealing method to deploy sensors such that the maximum distance error by the WSN is minimized. Wang et al. [13] deploy sensors in a strip-by-strip way, and [14] proves that such a deployment strategy spends the least number of sensors to provide full coverage. The study [15] considers deploying the fewest sensors in an indoor region that contains obstacles to guarantee sensing coverage and network connectivity. In [16], both Poisson and Gaussian distributions of sensors are evaluated, and a hybrid deployment solution is developed based on these two distributions.

Some studies aim to offer *k*-coverage of a sensing field. Specifically, [17] adopts a hexagon-like pattern to deploy sensors, while [18] uses the combination of 1-coverage and 3-coverage deployment to achieve *k*-coverage deployment. Considering that different regions of a sensing field may have different coverage requirements, [19] picks the least number of sensors to monitor *p*-percent of the sensing field. The work [20] considers deploying a 3D WSN and follows the idea of embedding the surface network to a planar topology to provide greedy routing in the WSN. Boubrima et al. [21] find the optimal locations of sensors and sinks in a city such that they can use the minimum number of sensors to monitor air pollution. However, the results of these research efforts cannot be applied to our problem due to different coverage model of sensors.

2.2. Deployment and Rotation Scheduling Methods for Visual WSNs

For visual WSNs, how to deploy sensors (i.e., cameras) is also critical [22]. The work [23] seeks to deploy the fewest cameras in a region such that the response time to detect the intrusion in the region is minimized. To do so, it models the FoV of each camera as a triangle and uses an approximating optimal visual sensor placement strategy [24] together with linear programming to deploy cameras. Phama et al. [25] give each subarea of the sensing field a different risk level, which depends on the event occurring in that subarea (e.g., detecting an intruder). Then, cameras are deployed in each subarea according to its risk level. The study [26] also divides the sensing field into subareas and assign each one a minimum coverage ratio. After randomly placing cameras, it adjusts the direction of each camera with the objective of meeting the coverage ratio of each subarea. In [27], a particle swarm optimization algorithm is proposed to solve the coverage problem of a visual WSN. It views cameras as a swarm and target objects as particles. Then, the direction of each camera is iteratively adjusted to maximize the coverage of all target objects. However, these studies do not take advantage of the rotation capability of PTZ cameras.

A number of studies deal with rotation scheduling of PTZ camera in a visual WSN. Supposing that each object is assigned with a threshold of monitoring probability, [28] considers how to schedule the rotation of PTZ cameras so as to make sure that each object can be monitored with a probability higher than its threshold. Given a set of objects, [29] discusses how to deploy the minimum cameras and decide their rotation, such that each object is covered by a fixed ratio of time in a period. The problem is NP-hard and two heuristics are proposed based on the distributions of objects. Li et al. [30] adopt the game theory to schedule camera rotation, whose goal is to find a solution to the problem of direction set *k*-cover for minimum coverage breach in a visual WSN. The study [31] translates the scheduling problem of PTZ cameras into a maximum vertex-weight matching problem, so as to pair each camera with a target object. In [32], a *generalized R&D sensor deployment (GRSD)* problem is formulated to use cameras to monitor a set of heterogeneous objects. It considers how to use the fewest PTZ cameras to achieve temporal coverage by making each object be t_i-time covered. The problem is also NP-hard, so a heuristic is developed based on the positions and coverage requirements of objects. Nevertheless, none of these studies addresses the concept of splitting vision. This motivates us to slice the FoV of each PTZ camera and associate each part a different weight, so as to provide different monitoring quality for objects.

3. PTZ Camera Scheduling Problem

3.1. Environmental Assumptions

In this paper, we use PTZ cameras that can freely rotate horizontally and direct their attention to target objects. During the monitoring process, a PTZ camera may stay in some FoVs briefly to monitor objects. Suppose that a set $\hat{C} = \{c_0, c_1, c_2 \dots, c_n\}$ of PTZ cameras are installed in the sensing field. Each camera has the same working period T (including both rotation and monitoring time) and can rotate 360 degrees. Let δ be the total rotation time for a camera c_i in a period to switch among FoVs. Besides, let $t_{i,k}^{cam}$ denote the time that camera c_i stays in FoV $f_{i,k}$. Then, we can derive that

$$T = \delta + T' \geq \delta + \Sigma\{t_{i,k}^{cam} | \forall f_{i,k} \in \hat{F}_i\}, \quad (1)$$

where T' is the total monitoring time and $\hat{F}_i$ is the set of c_i's FoVs. Fig. 1 shows an example. Since a PTZ camera will rotate 360 degrees in a period, we have $\delta = 2\pi/V_A$, where V_A is the angular velocity of PTZ cameras.

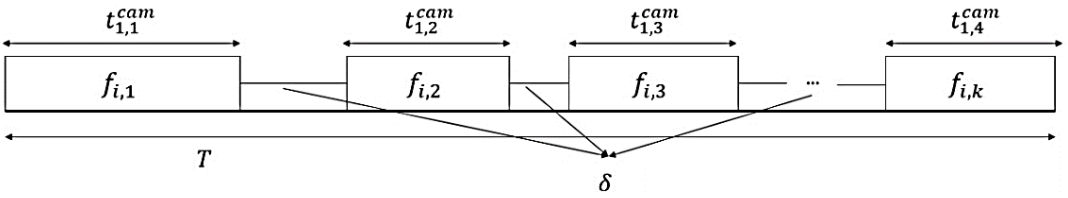


Fig. 1 The behavior of a PTZ camera in a period

A PTZ camera has to stop for a short time in order to capture a clear image to identify the monitoring object [33]. Besides, [34] points out that we should put an upper bound on the time that the PTZ camera switches between two monitoring objects, so as to avoid some objects uncovered for too long time. Thus, given a set $\hat{O}$ of target objects, we associate each object $o_j \in \hat{O}$ a minimum monitoring time t_j^{obj} and also a priority p_j to indicate its importance. Suppose that a subset $\hat{O}' \subseteq \hat{O}$ of objects are covered by a subset $\hat{C}' \subseteq \hat{C}$ of cameras. Then, we should guarantee that

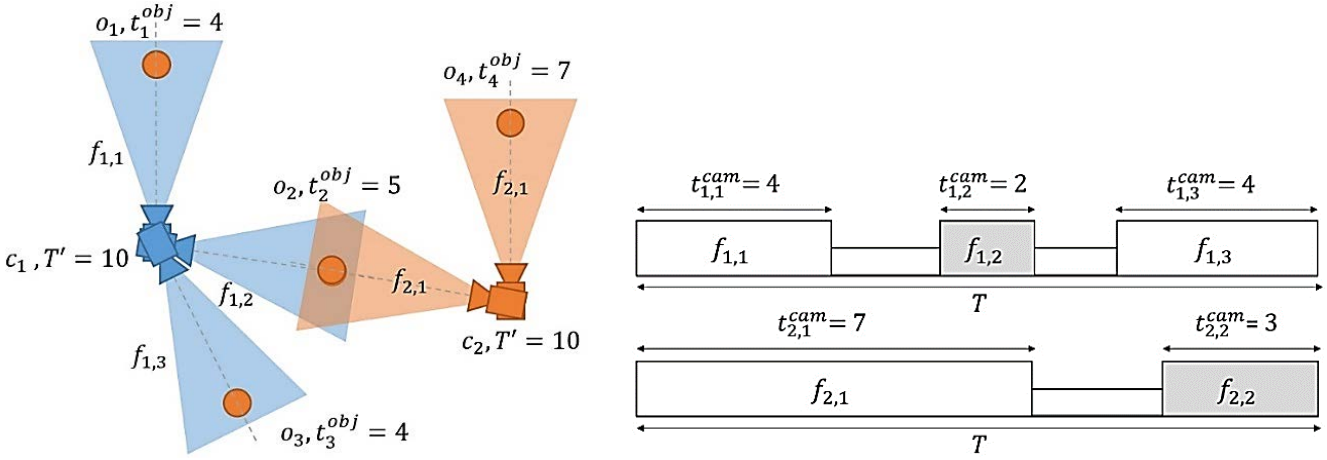
$$\Sigma_{c_i \in \hat{C}'} T' \geq \Sigma_{o_j \in \hat{O}'} t_j^{obj}. \quad (2)$$

In other words, the overall monitoring time of cameras in $\hat{C}'$ cannot be shorter than the sum of the minimum monitoring time of each object in $\hat{O}'$, or the t_j^{obj} constraint will be violated.

3.2. FoV and Splitting Vision

For a PTZ camera, its current FoV is defined by the sector area whose bisector is aligned with the camera's optical axis and has a viewing angle θ. As the PTZ camera can rotate, it has multiple FoVs. Thus, we denote by $f_{i,k}$ the kth FoV of a PTZ camera c_i, and $\hat{F}_i$ the set of $f_{i,k}$. Besides, all cameras have the same viewing angle, so their FoVs have the same size.

Since the FoVs of two cameras may overlap, some studies [35][36] encourage us exploiting this property to share the monitoring time of a target object by two or more cameras. We can take Fig. 2 as an example, where both cameras c_1 and c_2 cover object o_2. Either c_1 or c_2 cannot solely satisfy the minimum monitoring time t_2^{obj} of o_2. In this case, we can make them cooperatively monitor o_2 (at different times), so as to meet the t_2^{obj} constraint, as shown in Fig. 2(b).



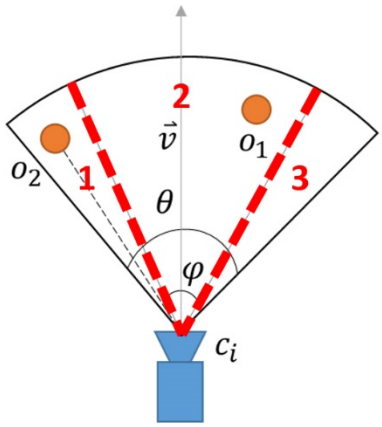
(a) FoVs of two PTZ cameras c_1 and c_2 (b) Rotation and monitoring scheduling of both c_1 and c_2

Fig. 2 Cooperative monitoring of two PTZ cameras

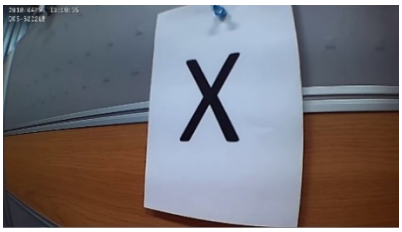
On the other hand, when the monitoring object locates in different parts of a FoV, the image quality provided by the PTZ camera may be different. Fig. 3 gives an example. When the object (marked as 'X') locates in o_1, the camera can offer better monitoring quality (i.e., the image is clearer). To reflect the above phenomenon, we propose the concept of splitting vision by dividing the FoV into three parts, as shown in Fig. 3(a). Moreover, we associate each part with a weight as follows:

$$\omega = \begin{cases} 1, \ if \ \left|arccos\left(\frac{\overrightarrow{c_i o_j} \cdot \vec{v}}{|\overrightarrow{c_i o_j}||\vec{v}|}\right)\right| \leq \frac{\varphi}{2} \\ \alpha, \ if \ \frac{\varphi}{2} < \left|arccos\left(\frac{\overrightarrow{c_i o_j} \cdot \vec{v}}{|\overrightarrow{c_i o_j}||\vec{v}|}\right)\right| \leq \frac{\theta}{2}, 0 < \alpha < 1 \end{cases} \quad (3)$$

Let us take Fig. 3(a) as an example, where $\vec{v}$ is the optical axis of a camera. Since the included angle between $\overrightarrow{o_1c_i}$ and $\vec{v}$ is smaller than $\varphi/2$, o_1 locates in part 2 and has a weight of 1. For o_2, as the included angle is between $\varphi/2$ and $\theta/2$, it locates in part 1 (or 3) and has a weight of $\alpha \in [0,1]$.



(a) Splitting vision



(b) Object locates in o_1

(c) Object locates in o_2

Fig. 3 Monitoring quality for an object when it locates in different parts of a FoV

3.3. *Problem Formulation*

Suppose that a PTZ camera c_i stays in each FoV $f_{i,k}$ for $t_{i,k}^{cam}$ time. We define its monitoring grade by

$$g_i = \Sigma_{o_j \in \hat{O}} \omega_{i,j} \times t_{i,k}^{cam} \times p_j, \quad (4)$$

Where $\omega_{i,j}$ is c_i's weight for object o_j. Then, we formulate the PTZ camera scheduling problem as follows:

Objective: Maximize $\Sigma_{c_i \in \hat{C}}\, g_i$ (5)

Constraints:

$t_j^{obj} > 0, \forall o_j \in \hat{O}$ (6)

$t_{i,k}^{cam} > 0, \forall f_{i,k} \in \hat{F}_i, \forall c_i \in \hat{C}$ (7)

$\hat{C} \neq \emptyset, \hat{O} \neq \emptyset$ (8)

$T = \delta + T' \geq \delta + \Sigma\{t_{i,k}^{cam} | \forall f_{i,k} \in \hat{F}_i\}$ (9)

$\Sigma_{c_i \in \hat{C}}, T' \geq \Sigma_{o_j \in \hat{O}}, t_j^{obj}$ (10)

Here, Eq. (5) means to maximize the overal minitoring grade. Eq. (6) indicates that each object has a positive t_j^{obj} time. Eq. (7) points out that a camera will stay in each selected FoV for a non-zero time. Clearly, both $\hat{C}$ and $\hat{O}$ cannot be empty, as shown in Eq. (8). The meanings of Eqs. (9) and (10) have been discussed in Section 3.1. In fact, the above formulation follows the format of mixed integer programming, so the PTZ camera scheduling problem is NP-hard in essence.

4. The Proposed ECRS Heuristic

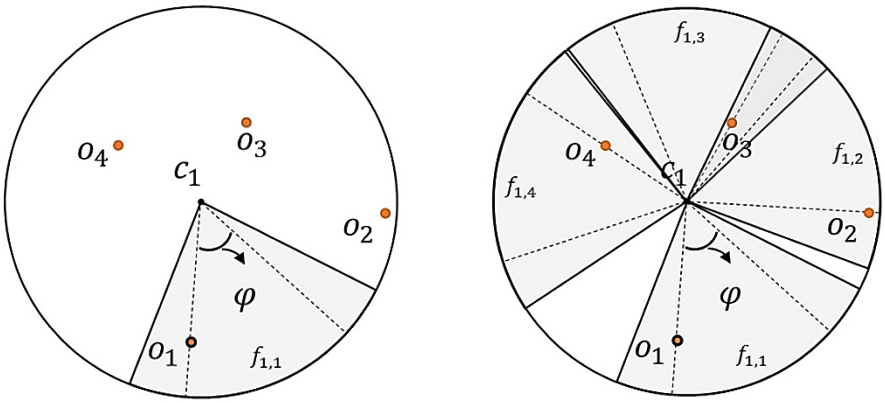
Our ECRS heuristic contains four steps. In step 1, we identify all FoVs of each PTZ camera based on the distribution of objects in $\hat{O}$. Then, step 2 picks an object from $\hat{O}$ and finds the corresponding FoV, while step 3 decides its $t_{i,k}^{cam}$ and updates

the available monitoring time of the camera. Both steps are repeated until the termination conditions are met in step 4. Below, we discuss the detailed design in each step.

4.1. Step 1: Identify FoVs of a Camera

For each PTZ camera c_i, we first find out all of its FoVs, where each FoV should cover at least one object in $\hat{O}$. To do so, we model the *pontential sensing coverage* of the camera as a disk whose center locates at the camera's position and radius is the sensing distance r_s. Then, for each object $o_j \in \hat{O}$ located in the disk, we expand the sector area from $\overrightarrow{c_i o_j}$ with an angle of $(\theta - \varphi)/2$ clockwise (denoted by A_1) and the sector area from $\overrightarrow{c_i o_j}$ with an angle of $(\theta + \varphi)/2$ counterclockwise (denoted by A_2). In this way, a FoV is found by combining both A_1 and A_2. Then, we record the objects covered by the FoV.

Fig. 4(a) gives an example. For object o_1, we expand sector area A_1 from $\overrightarrow{c_1 o_1}$ with an angle of $(\theta - \varphi)/2$ clockwise and also sector area A_2 from $\overrightarrow{c_1 o_1}$ with an angle of $(\theta + \varphi)/2$ counterclockwise. Then, FoV $f_{1,1}$ will be the union of A_1 and A_2. Fig. 4(b) shows the four FoVs found by the above procedure.



(a) Identify FoV $f_{1,1}$ (b) The final result

Fig. 4 An example of finding FoVs of a PTZ camera

4.2. Step 2: Pick an Object and Find the Corresponding FoV

For each object $o_j \in \hat{O}$, we define t_j^{remain} as its residual required time to be monitored by PTZ cameras. In the beginning, t_j^{remain} is set to t_j^{obj} (i.e., the minimum monitoring time). On the other hand, for each camera $c_i \in \hat{C}$, we also define T_i^{remain} to be its available monitoring time. From the discussion in Section 3.1, T_i^{remain} is initially set to T_i'.

We then pick an object and find the corresponding FoV based on a greedy approach. In particular, among all objects in $\hat{O}$ that have positive t_j^{remain} values, we choose the object o_j whose priority p_j is the largest. In case of tie, we select the object with the maximum t_j^{remain} value. Then, among all cameras in $\hat{C}$ whose FoVs can cover o_j, we select the camera c_i that has the maximum T_i^{remain} time. Afterwards, if o_j is covered by multiple FoVs of c_i, we then select the FoV as follows:

$$f_{i,k} = arg_{f_{i,k} \in \hat{F}_i} max \sum_{o_j \in \hat{O}_{i,k}} \omega_{i,j} \times t_{i,k}^{cam} \times p_j, \quad (11)$$

where $\hat{O}_{i,k}$ denotes the set of objects covered by $f_{i,k}$. In other words, we will select the FoV $f_{i,k}$ that achieves the highest monitoring grades. However, since we do not know $t_{i,k}^{cam}$ yet (this paramter will be calculated in step 3), we can replace $t_{i,k}^{cam}$ in Eq. (11) by $min\{t_j^{remain}, T_i^{remain}\}$ (i.e., the minimum value between the residual required monitoring time of object o_j and the available monitoring time of camera c_i).

4.3. Step 3: Decide $t_{i,k}^{cam}$ and Update the Available Monitoring Time

After choosing FoV $f_{i,k}$ and object o_j, we calculate the time $t_{i,k}^{cam}$ for camera c_i to stay in $f_{i,k}$ to monitor o_j. However, $f_{i,k}$ may include objects other than o_j. Thus, we consider the *longest* residual required monitoring time:

$$t_{i,k}^{max} = max_{o_v \in \hat{O}_{i,k}} \{t_v^{remain}\}. \quad (12)$$

Then, the following pseudocode helps us decide $t_{i,k}^{cam}$ and also update the available monitoring time T_i^{remain} of c_i:

```
if T_i^remain > t_{i,k}^max
  t_{i,k}^cam = t_{i,k}^max;
  T_i^remain = T_i^remain − t_{i,k}^max;
else
  t_{i,k}^cam = T_i^remain;
  T_i^remain = 0;

for o_v in Ô_{i,k}
  if t_v^remain > t_{i,k}^cam
    t_v^remain = t_v^remain - t_{i,k}^cam;
  else
    t_v^remain = 0;
end
```

Specifically, if the available monitoring time is larger than $t_{i,k}^{max}$, we can directly set $t_{i,k}^{cam}$ to $t_{i,k}^{max}$ and also deduct T_i^{remain} by $t_{i,k}^{max}$. Otherwise, it means that there remains insufficient monitoring time for camera c_i. In this case, $t_{i,k}^{cam}$ should be set to T_i^{remain} and we then set T_i^{remain} to zero, so c_i will not participate in the scheduling process in the next round. Since FoV $f_{i,k}$ covers each object o_v in $\hat{O}_{i,k}$, we need to update its residual required monitoring time t_v^{remain}. Note that when $t_{i,k}^{cam}$ is no smaller than t_v^{remain}, it means that c_i (possibly with other cameras) can satisfy the minimum monitoring time of o_v. Thus, we set t_v^{remain} to zero, so the object will not be considered in the next round.

4.4. Step 4: Termination Conditions

Both steps 2 and 3 are repeated until 1) every camera has run out of monitoring time or 2) the minimum monitoring time of each object has been satisified. For the second condition, if a camera still has available monitoring time, it refers to Eq. (11) (by replaing $t_{i,k}^{cam}$ with T_i^{remain}) to select the FoV $f_{i,k}$ and then spends time T_i^{remain} on staying in $f_{i,k}$, so as to maximize its monitoring grade.

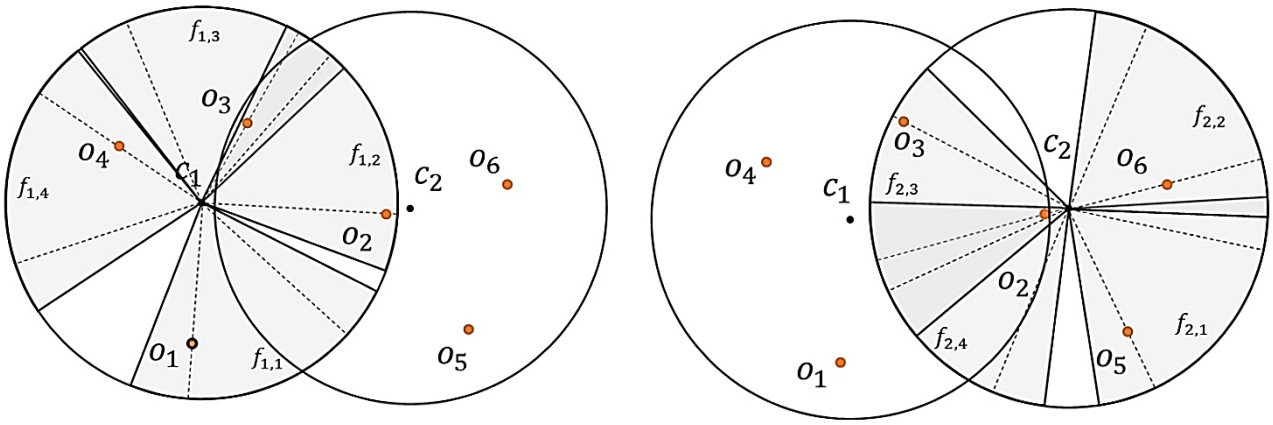


Fig. 5 FoVs of two PTZ cameras c_1 and c_2

Let us use the example in Fig. 5 to show how ECRS works, where we have two cameras c_1 and c_2 to monitor six objects. Suppose that the working period T is 10 seconds. Table 1 gives the status in each round by ECRS. Specifically, since objects

o_1 and o_5 have the largest priority (i.e., 10) but o_5 has longer t_j^{remain} time, we choose FoV $f_{2,1}$ to cover o_5, with staying time of 6 seconds in round 1. Then, we select FoV $f_{1,1}$ to cover o_1. In round 3, we choose object o_3 to monitor. Since both cameras c_1 and c_2 can cover o_3, we select c_1 because it has longer available monitoring time. In round 4, we use FoV $f_{1,4}$ to cover object o_4. Finally, object o_6 is covered by FoV $f_{2,2}$ in round 5.Thus, the minimum monitoring time of each object can be met.

Status			Round 1		Round 2		Round 3		Round 4		Round 5	
Object	p_j	t_j^{remain}	t_j^{remain}		t_j^{remain}		t_j^{remain}		t_j^{remain}		t_j^{remain}	
o_1	10	4	-	4	$f_{1,1}$	0	-	0	-	0	-	
o_2	1	2	-	2	-	2	$f_{1,2}$	0	-	0	-	
o_3	5	4	-	4	-	4	$f_{1,2}$	0	-	0	-	
o_4	5	2	-	2	-	2	-	2	$f_{1,4}$	0	-	
o_5	10	6	$f_{2,1}$	0	-	0	-	0	-	0	-	
o_6	1	4	-	4	-	4	-	4	-	4	$f_{2,2}$	0

(a) Update t_j^{remain} of each object

Status		Round 1		Round 2		Round 3		Round 4		Round 5	
Camera	T_i^{remain}	FoV	$t_{i,k}^{cam}$	FoV	$t_{i,k}^{cam}$	FoV	$t_{i,k}^{cam}$	FoV	$t_{i,k}^{cam}$	FoV	$t_{i,k}^{cam}$
c_1	10	-	-	$f_{1,1}$	4	$f_{1,2}$	4	$f_{1,4}$	2	-	-
c_2	10	$f_{2,1}$	6	-	-	-	-	-	-	$f_{2,2}$	4

(b) Selected camera and FoV in each round

	o_1	o_2	o_3	o_4	o_5	o_6
c_1	4	4	4	2	×	×
c_2	×	×	×	×	6	4

(c) Monitoring time of objects by each camera

Table 1. An example to show how ECRS works

5. Performance Evaluation

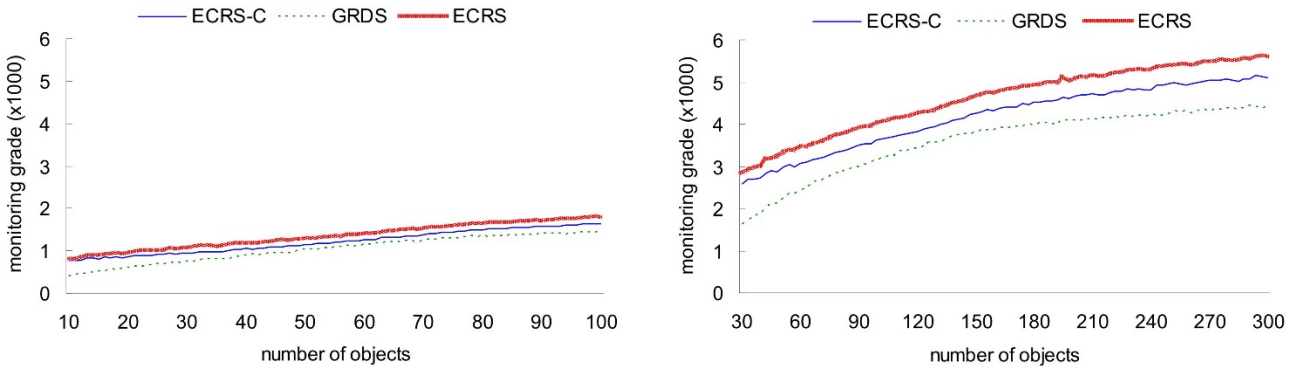
We develop a simulator in Python to evaluate performance of the ECRS heuristic. The sensing field is a square whose width is 400, inside which a number of target objects are randomly placed. We divide these objects into three groups, each with the same number of objects. For the objects in groups 1, 2, and 3, we set their priorities to 1, 5, and 10, respectively. Then, we use the Poisson-disk sampling algorithm [37] to deploy PTZ cameras, where the distance between any two cameras is kept in $[r_s/2, r_s]$. Therefore, each camera can cover a similar number of objects. The working period T of each camera is set to 10 seconds. To find a feasible solution, we have to make sure that

$$\sum_{o_j \in \hat{O}} t_j^{obj} \leq T \times N_C, \qquad (13)$$

where N_C is the total number of cameras. We compare ECRS with the GRSD heuristic [32] discussed in Section 2.2. Besides, we also create an ECRS-C method for comparison, where it finds each FoV by aligning the camera's optical axis and an object in step 1. For each experiment, we repeat the simulation for 100 times and take their average.

We first measure the effect of different numbers of objects and cameras on the monitoring grade. In this experiment, we set θ to 60 degrees and φ to 20 degrees. Besides, α is set to 0.3 in Eq. (3) to decide the weight of each part in a FoV. Fig. 6(a) gives the result when there are 10 cameras to monitor 100 objects, while Fig. 6(b) presents the result when there are 30 cameras

to monitor 300 objects. Obviously, the monitoring grade increases as the number of objects grows (referring to Eq. (5)). Since the GRDS heuristic does not consider splitting vision, it always results in the lowest monitoring grade. On the other hand, our ECRS heuristic finds out FoVs of each camera by exploiting splitting vision. Besides, it greedily picks the object with the highest priority and also the largest t_j^{remain} value to serve first, so ECRS can achieve much higher monitoring grade than other methods. Such a phenomenon is more significant when there are 300 objects and 30 camera, as shown in Fig. 6(b).

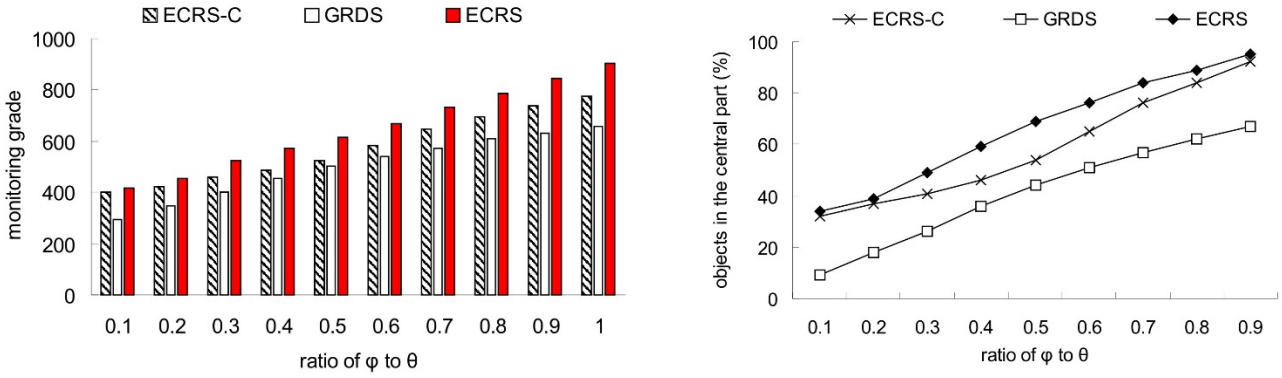


(a) 10 cameras and 100 objects

(b) 30 cameras and 300 objects

Fig. 6 Comparison on monitoring grades under different numbers of cameras and objects

Then, we evaluate the effect of splitting vision by angle φ. In this experiment, there are 20 cameras used to monitor 120 objects, and the viewing angle θ is set to 120 degrees. Fig. 7(a) shows the effect of φ on the monitoring grade. It is obvious that the monitoring grade increases in each method when we enlarge φ. The reason is that the central part (i.e., part 2 in Fig. 3(a)) of each FoV increases, so it can cover more objects and thus improves the monitoring grade, as shown in Fig. 7(b). In fact, ECRS can let more objects be covered by the central part of each FoV when φ increases, as comparing with other two methods. From Fig. 7(a), we observe that our ECRS heuristic always keeps the highest monitoring grade, even though we remove the effect of splitting vision by setting φ to θ (i.e., the ratio is 1). This result demonstrates the superiority of ECRS over other methods.



(a) monitoring grade

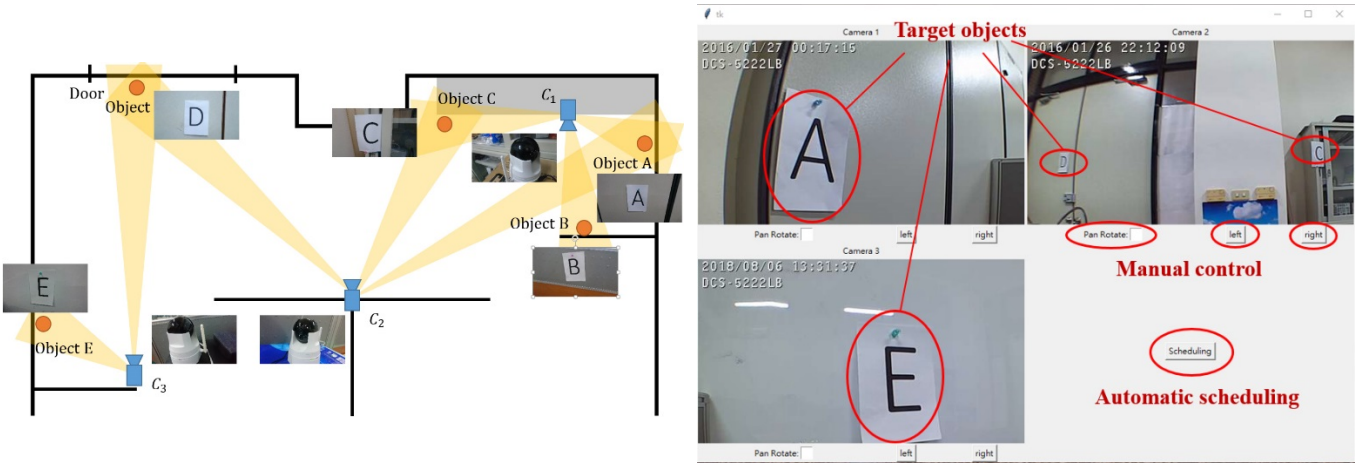
(b) percentage of objects in the central part

Fig. 7 Effect of splitting vision by angle φ

6. System Implementation

To implement the ECRS heuristic, three D-Link 5222L PTZ cameras [38] are installed to monitor five objects in an office, as shown in Fig. 8(a). Each camera provides megapixel HD 720p resolution and can be remotely controlled. In particular, we use multithreading along with HTTP request to synchronously control these cameras. However, the maximum rotation angle of a camera is limited to 340 degrees, so we design a zeroing mechanism. In the beginning, we make each camera rotate to the end

counterclockwise and set this direction as a reference of zero degree. After finishing one period of the monitoring job, we also ask a camera to rotate back to the above reference direction. In this way, we can imitate the behavior of rotating 360 degrees.



(a) System deployment (b) GUI

Fig. 8 System implementation of ECRS

Fig. 8(b) illustrates our graphical user interface (GUI), which is implemented by the tkinter component in Python. In the GUI, we provide three separated screens to show the videos captured by each PTZ camera. In addition, a user can manually control each camera by asking it to rotate left (i.e., counterclockwise) or right (i.e., clockwise) with a certain degree. When the user presses the "scheduling" button (in the bottom right concern), these three cameras will automatically rotate to monitor each object, which follows the scheduling result of our ECRS heuristic. This system also provides a prototype for security surveillance in an indoor environment.

7. Conclusion and Future Work

Thanks to the rotation capability of PTZ cameras, they are widely used in indoor surveillance scenarios. By considering a new concept of splitting vision, this paper models the rotation scheduling problem of PTZ cameras in a visual WSN by linear programming and also develops the ECRS heuristic to efficiently solve the problem. ECRS first finds out FoVs of cameras to monitor objects and iteratively pairs objects and FoVs with the goal of increasing the monitoring grade. It also makes two cameras with overlapped FoVs cooperatively monitor the same objects. Simulation results in Python demonstrates that ECRS outperforms other methods in terms of the monitoring grade, which means that it can offer better monitoring quality for longer time. Moreover, we also implement ECRS by using three PTZ cameras to monitor objects in an office.

We give some directions for future work. First, the transmission of video streaming consumes lots of network bandwidth. When some video frames are lost, we can partially retransmit necessary frames to improve the video quality [39]. In addition, different data compression techniques [40] can be also applied to save network bandwidth. Second, we consider only fixed objects in this paper. When objects will move inside the sensing field, how to efficiently track them is a challenge. There are two possible solutions. One is to assume that the mobility of objects follows some regular patterns [41], so we can schedule the rotation of cameras based on these mobility patterns. Alternatively, objects can be associated with RFID (radio frequency identification) tags [42], so as to notify cameras of the positions of objects. Third, since PTZ cameras are used in surveillance applications, how to provide secure transmission of sensing data is critical [43]. This security issue will be especially important for IoT scenarios, where the monitoring data will be continuously sent to a data center network for storage and processing [44]. Finally, as the technology of mobile platforms (e.g., robots or vehicles) is mature, it is interesting to mount PTZ cameras on these platforms to make them become mobile sensors [45]. Some challenging issues will be also arisen, for example, how to

tactically and quickly dispatch these mobile sensors to detect abnormal events [46].

References

[1] I. Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow, and P. Polakos, "Wireless sensor network virtualization: a survey," IEEE Communications Surveys & Tutorials, vol. 18, pp. 553-576, 2016.

[2] Y. C. Wang, W. C. Peng, and Y. C. Tseng, "Energy-balanced dispatch of mobile sensors in a hybrid wireless sensor network," IEEE Transactions on Parallel and Distributed Systems, vol. 21, pp. 1836-1850, 2010.

[3] Y. C. Tseng, Y. C. Wang, K. Y. Cheng, and Y. Y. Hsieh, "iMouse: an integrated mobile surveillance and wireless sensor system," IEEE Computer, vol. 40, pp. 60-66, 2007.

[4] L. W. Yeh, Y. C. Wang, and Y. C. Tseng, "iPower: an energy conservation system for intelligent buildings by wireless sensor networks," International Journal of Sensor Networks, vol. 5, pp. 1-10, 2009.

[5] Y. C. Wang and C. C. Yang, "3S-cart: a lightweight, interactive sensor-based cart for smart shopping in supermarkets," IEEE Sensors Journal, vol. 16, pp. 6774-6781, 2016.

[6] A. Alaiad and L. Zhou, "Patients' adoption of WSN-based smart home healthcare systems: an integrated model of facilitators and barriers," IEEE Transactions on Professional Communication, vol. 60, pp. 4-23, 2017.

[7] Y. C. Wang and W. T. Chen, "An automatic and adaptive light control system by integrating wireless sensors and brain-computer interface," IEEE International Conference on Applied System Innovation, 2017, pp. 1399-1402.

[8] Y. C. Wang and G. W. Chen, "Efficient data gathering and estimation for metropolitan air quality monitoring by using vehicular sensor networks," IEEE Transactions on Vehicular Technology, vol. 66, pp. 7234-7248, 2017.

[9] C. Ding, B. Song, A. Morye, J. A. Farrell, and A. K. Roy-Chowdhury, "Collaborative sensing in a distributed PTZ camera network," IEEE Transactions on Image Processing, vol. 21, pp. 3282-3295, 2012.

[10] Y. C. Wang, "Mobile sensor networks: system hardware and dispatch software," ACM Computing Surveys, vol. 47, pp. 12:1-12:36, 2014.

[11] Y. C. Wang, K. Y. Cheng, and Y. C. Tseng, "Using event detection latency to evaluate the coverage of a wireless sensor network," Computer Communications, vol. 30, pp. 2699-2707, 2007.

[12] F. Y. S. Lin and P. L. Chiu, "A near-optimal sensor placement algorithm to achieve complete coverage/discrimination in sensor networks," IEEE Communications Letters, vol. 9, pp. 43-45, 2005.

[13] Y. C. Wang, C. C. Hu, and Y. C. Tseng, "Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks," IEEE Wireless Internet Conference, pp. 114-121, 2005.

[14] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. H. Lai, "Deploying wireless sensors to achieve both coverage and connectivity," ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 131-142, 2006.

[15] Y. C. Wang, C. C. Hu, and Y. C. Tseng, "Efficient placement and dispatch of sensors in a wireless sensor network," IEEE Transactions on Mobile Computing, vol. 7, pp. 262-274, 2008.

[16] Y. Wang, M. Wilkerson, and X. Yu, "Hybrid sensor deployment for surveillance and target detection in wireless sensor networks," IEEE International Wireless Communications and Mobile Computing Conference, pp. 326-330, 2011.

[17] T. Sun, L. J. Chen, C. C. Han, and M. Gerla, "Reliable sensor networks for planet exploration," IEEE International Conference on Networking, Sensing and Control, pp. 816-821, 2005.

[18] Y. C. Wang and Y. C. Tseng, "Distributed deployment schemes for mobile wireless sensor networks to ensure multilevel coverage," IEEE Transactions on Parallel and Distributed Systems, vol. 19, pp. 1280-1294, 2008.

[19] Y. Li, C. Ai, Z. Cai, and R. Beyah, "Sensor scheduling for *p*-percent coverage in wireless sensor networks," Cluster Computing, vol. 14, pp. 27-40, 2011.

[20] C. Wang, H. Jiang, T. Yu, and C. S. Liu, "SLICE: enabling greedy routing in high genus 3-D WSNs with general topologies," IEEE/ACM Transactions on Networking, vol. 24, pp. 2472-2484, 2016.

[21] A. Boubrima, W. Bechkit, and H. Rivano, "A new WSN deployment approach for air pollution monitoring," IEEE Annual Consumer Communications and Networking Conference , pp. 455-460, 2017.

[22] Y. Charfi, N. Wakamiya, and M. Murata, "Challenging issues in visual sensor networks," IEEE Wireless Communications, vol. 16, pp. 44-49, 2009.

[23] Y. Morsly, M. S. Djouadi, and N. Aouf, "On the best interceptor placement for an optimally deployed visual sensor network," IEEE International Conference on Systems, Man and Cybernetics, pp. 43-51, 2010.

[24] E. Horster, and R. Lienhart, "Approximating optimal visual sensor placement," IEEE International Conference on Multimedia and Expo, pp. 1257-1260, 2006.

[25] C. Phama, A. Makhoulb, and R. Saadic, “Risk-based adaptive scheduling in randomly deployed video sensor networks for critical surveillance applications,” Journal of Network and Computer Applications, vol. 34, pp. 783-795, 2011.
[26] C. Han, L. Sun, J. Guo, and C. Chen “Rotatable sensor scheduling for multi-demands of coverage in directional sensor networks,” IEEE International Conference on Computer Communication and Networks, pp. 1-8, 2016.
[27] S. Peng, Y. Xiong, M. Wu, and J. She, “A new method of deploying nodes for area coverage rate maximization in directional sensor network,” Annual Conference of the IEEE Industrial Electronics Society, pp. 8452-8457, 2017.
[28] P. Zhou, J. Wu, and C. Long, “Probability-based optimal coverage of PTZ camera networks”, IEEE International Conference on Communications, pp. 218-222, 2012.
[29] Y. C. Wang, Y. F. Chen, and Y. C. Tseng, “Using rotatable and directional (R&D) sensors to achieve temporal coverage of objects and its surveillance application,” IEEE Transactions on Mobile Computing, vol. 11, pp. 1358-1371, 2012.
[30] J. Li, K. Yue, W. Y. Liu, and Q. Liu, “Game-theoretic based distributed scheduling algorithms for minimum coverage breach in directional sensor networks,” International Journal of Distributed Sensor Networks, vol. 10, pp. 1-12, 2014.
[31] L. Guo, Y. Zhu, D. Li, and D. Kim, “PTZ camera scheduling for selected area coverage in visual sensor networks,” IEEE International Conference on Distributed Computing Systems, pp. 379-388, 2015.
[32] Y. C. Wang and S. E. Hsu, “Deploying R&D sensors to monitor heterogeneous objects and accomplish temporal coverage,” Pervasive and Mobile Computing, vol. 21, pp. 30-46, 2015.
[33] A. Mittal, L. S. Davis, and S. N. Lim, “Constructing task visibility intervals for a surveillance system,” ACM International Workshop on Video Surveillance & Sensor Networks, pp. 141-148, 2005.
[34] N. Basilico, N. Gatti, and F. Amigoni, “Developing a deterministic patrolling strategy for security agents,” IEEE/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 565-572, 2009.
[35] M. Alaei, and J. M. Barcelo-Ordinas, “Node clustering based on overlapping FoVs for wireless multimedia sensor networks,” IEEE Wireless Communication and Networking Conference, pp. 1-6, 2010.
[36] Y. C. Wang and S. E. Hsu, “An efficient deployment heuristic to support temporal coverage of heterogeneous objects in rotatable and directional (R&D) sensor networks,” IEEE Vehicular Technology Conference, pp. 1-5, 2014.
[37] R. Bridson, “Fast Poisson disk sampling in arbitrary dimensions,” SIGGRAPH sketches, pp. 22, 2007.
[38] D-Link. [Online]. Available: https://eu.dlink.com/uk/en/products/dcs-5222l-pan-tilt-zoom-cloud-camera
[39] Y. C. Wang, “Profit-based exclusive-or coding algorithm for data retransmission in DVB-H with a recovery network,” International Journal of Communication Systems, vol. 28, pp. 1580-1597, 2015.
[40] Y. C. Wang, “Data compression techniques in wireless sensor networks,” Pervasive Computing, Nova Science Publishers, 2012.
[41] W. H. Yang, Y. C. Wang, Y. C. Tseng, and B. S. P. Lin, “Energy-efficient network selection with mobility pattern awareness in an integrated WiMAX and WiFi network,” International Journal on Communication Systems, vol. 23, pp. 213-230, 2010.
[42] Y. C. Wang and S. J. Liu, “Minimum-cost deployment of adjustable readers to provide complete coverage of tags in RFID systems,” Journal of Systems and Software, vol. 134, pp. 228-241, 2017.
[43] Y. C. Wang and Y. C. Tseng, “Attacks and defenses of routing mechanisms in ad hoc and sensor networks,” Security in Sensor Networks, CRC Press, 2006.
[44] Y. C. Wang and S. Y. You, “An efficient route management framework for load balance and overhead reduction in SDN-based data center networks,” IEEE Transactions on Network and Service Management, vol. 15, no. 4, pp. 1422-1434, 2018.
[45] Y. C. Wang, F. J. Wu, and Y. C. Tseng, “Mobility management algorithms and applications for mobile sensor networks,” Wireless Communications and Mobile Computing, vol. 12, pp. 7-21, 2012.
[46] Y. C. Wang, “A two-phase dispatch heuristic to schedule the movement of multi-attribute mobile sensors in a hybrid wireless sensor network,” IEEE Transactions on Mobile Computing, vol. 13, pp. 709-722, 2014.