Sink Deployment in Wireless Surveillance Systems

Michael Chien-Chun Hung and Kate Ching-Ju Lin
Research Center for Information Technology Innovation, Academia Sinica
128 Academia Road, Section 2, Nankang, Taipei 115, Taiwan
E-mail:{shinglee, katelin}@citi.sinica.edu.tw

Abstract—In this paper, we investigate the Access Point (AP) deployment problem in Wireless Surveillance Systems (WSSs). Cameras in a WSS could have various demanding upload streaming rates to the AP (usually deemed as a sink or data collector) according to their surveillance requirements, e.g., object detection frequency, video resolution, or the successful ratio of object tracking. Therefore, a suitable AP location must be determined in order to satisfy each camera’s demand. We first provide a mathematical formulation that defines the best location of the AP, and then propose a novel mechanism, called Spring-Cam, to exploit the concept of mass-spring systems in Physics to locate the suitable AP location with consideration of camera demands as well as transmission quality of the link between each camera and the AP. Simulation results show that Spring-Cam can use only a limited measurement overhead to find an AP location that achieves a performance comparable with exhausted searching the best location.

I. INTRODUCTION

In recent years, Wireless Surveillance Systems (WSSs) have grown to become an emerging research topic in the area of wireless networks. In a WSS, wireless cameras monitor a specified field of vision to track targeting objects or events, and transmit the surveillance video streams over wireless medium back to the Access Point (AP), which acts as the sink and data analyzer. Plenty of the related systems [1]–[3] have been developed and deployed in the real world.

In a WSS, the locations of cameras are usually determined based on the requirements specified by the surveillance applications, e.g., object tracking [4], [5]. Each camera may have different demands on the video streaming rate, depending on the frequency of occurrences of target detection or motion of video frames. We notice that the demanding streaming rate of a camera is closely determined by environmental conditions, like motion or density of monitored objects. For instance, a camera might require a high uploading throughput if it monitors a high-motion and dense hotspot, e.g., the entrance of a train station, because that camera might capture more information and generate a stream with a higher video bit-rate. In addition, since the locations of cameras are usually deployed in fixed locations, it is very likely that the demanding streaming rate of each camera can be estimated by a long-term measurement.

Different from commercial wireless networks that have dynamic throughput demands from various mobile clients, a WSS can deploy its AP based on the prior information about the static location and estimated traffic demand of each camera. Hence, given the deployment of cameras, another essential problem for a WSS is how to guarantee that video streams collected in the AP have a high enough visual quality for post processing, e.g., event detection or object tracking. Intuitively, a WSS should enable each camera to obtain an uploading throughput proportional to their demanding streaming rates. Therefore, the main interest of this work is to deploy the AP in a suitable location with consideration of each camera’s demand as well as the quality of wireless link between each camera and the AP. Note that our work is complementary to previous works on camera deployment; that is, given the camera locations determined by surveillance requirements, our goal is to select the best AP location that allows the AP to receive a satisfactory video quality.

Nowadays, most wireless networks, e.g., 802.11, support multiple transmission bit-rates, each of which has a different modulation and channel coding rate when packets are transmitted in the physical layer. Some auto-rate algorithms [6], [7] are proposed to enable each wireless camera to select its transmission bit-rate that can achieve the maximal data rate when it occupies the channel. However, according to the previous study [8], the actual throughput that a camera can obtain is not only determined by its transmission bit-rate, but also affected by the bit-rates used by neighboring wireless cameras. Even worse, the achievable throughput decreases significantly when neighboring cameras use a low transmission bit-rate. This is because every camera has equal probability to occupy the channel, while sending at a lower bit-rate would occupy the channel for a much longer time, which, in turn, decreases the channel access opportunity for those nodes using a higher rate.

In order to find out the optimal AP location, we can apply an exhausted search algorithm that selects the best AP location by testing every possible location. However, in doing so, the system developer might need to spend a significant time to measure the performance of each individual location. To reduce such overhead, we propose a novel mechanism, Spring-Cam, which adopts the mass-spring operations to determine the potential AP location. The system developer can further search a better AP location by measuring only a small number of other locations surrounding the potential AP location. Simulation results show that the potential AP location solved by Spring-Cam can achieve a fair performance without any measurement overhead; in addition, applying extra location search surrounding the potential AP location can gain substantial improvement with a reasonable measurement overhead.

The contributions of this work are summarized as follows.

• We introduce the AP deployment problem in WSSs, which aims to best satisfy the demanding uploading
throughput of each camera by deploying the AP at the most suitable location.

- A novel mechanism, Spring-Cam, is proposed to adopt the concept of mass-spring systems in Physics to locate the best AP location with a reasonable search overhead.

The remainder of the paper is organized as follows. Section II briefly describes related works. Section III presents the system overview and describes the problem, while Spring-Cam is proposed in Section IV. We evaluate the performance of Spring-Cam via simulations in Section V. Finally, Section VI concludes this paper.

II. RELATED WORK

Consistent labeling [9] in WSSs is to recognize multiple tracks of the same object when it appears in the field of another camera, or shows-up in the field of the same camera after disappearing for a while. Numerous works [9]–[12] investigate how to identify multiple tracks of the same object across single or multiple cameras, and label them consistently.

Camera placement is related to the problems in the area of image processing. Some previous works of photogrammetry [13]–[15] analyze the 3D measurement error propagation, and develop a camera deployment strategy that can reduce the overall measurement error. Some other works [4], [5] study how to form a cluster of cameras that can cooperatively track certain objects, and propose mechanisms to deal with cluster hand-off; specifically, the system must re-form a new cluster if the targeting objects are moving out of the monitoring area of the original cluster. The work [5] analyzes the optimal placement of multiple cameras as well as selecting a group of cameras to track the targeting objects. Note that our work is complimentary to all above studies because we focus on enhancing transmission quality in a WSS given the camera deployment and post-processing requirements.

Enhancing network communication efficiency is another issue in WSSs. In [12], the communication traffic is eliminated by an idle probability-based broadcast protocol that can reduce redundancy and avoid data collision. Some other works aim to by an idle probability-based broadcast protocol that can reduce the issue in WSSs. In [12], the communication traffic is eliminated.

The remainder of the paper is organized as follows. Section II briefly describes related works. Section III presents the system overview and describes the problem, while Spring-Cam is proposed in Section IV. We evaluate the performance of Spring-Cam via simulations in Section V. Finally, Section VI concludes this paper.

III. SYSTEM OVERVIEW

We consider a wireless surveillance system (WSS) including a set of cameras $\mathcal{V}$ that upload video streams to an access point (AP). Suppose all cameras in $\mathcal{V}$ are deployed in the pre-defined locations based on [12] for certain surveillance requirements, e.g., the successful ratio of object tracking. In this paper, the design goal is to determine the location of the AP such that the overall satisfaction of all cameras in a WSS can be maximized. The metric overall satisfaction is defined as

$$\text{metric} = \sum_{i \in \mathcal{V}} \frac{u_i}{d_i}$$

where $d_i$ denotes the demanding streaming rate of camera $i \in \mathcal{V}$ and $u_i$ represents the achievable uploading throughput that can be flowed from camera $i$ into the AP.

The 802.11 standard provides multiple bit-rates, which are collected as a set $\mathcal{R}$. Each camera $i \in \mathcal{V}$ independently uses certain auto-rate selection algorithms, e.g., [6], [7], to choose its best transmission bit-rate $r_i \in \mathcal{R}$ according to quality of the link between itself and AP. In general, link quality is mainly affected by the distance of that link due to channel fading. Note that even though multi-path fading and interference may also affect the link quality, their impacts are relatively minor. Besides, they heavily depend on environmental conditions, which make the prediction of their impacts more challenging. Therefore, we ignore multi-path fading and interference in our formulation, and consider them by measuring the actual link quality in the advanced search phase.

On the other hand, due to performance anomaly [8], contention among all nodes in the interference area results in that all the nodes would ultimately obtain a similar throughput, no matter what bit-rate it utilizes. Based on the previous work [17], when all cameras continuously contend for wireless bandwidth, due to performance anomaly, the achievable throughput can be derived as

$$u_i = \hat{u} \cdot p_i = \frac{p_i}{\sum_{j \in \mathcal{V}} \frac{1}{r_j}}, \forall i \in \mathcal{V}$$

where $\hat{u}$ represents the approximated achievable uploading throughput that the AP can receive from each camera in a long-term without considering link reliability, and $p_i$ is the link reliability between camera $i$ and the AP. As a result, the achievable uploading throughput $u_i$ of a link that delivers data at a higher bit-rate is degraded by other links choosing a lower bit-rate. Based on Eq. (2), the performance metric defined in Eq. (1) can be further reformulated as

$$\text{metric} = \sum_{i \in \mathcal{V}} \frac{u_i}{d_i} = \sum_{i \in \mathcal{V}} \frac{\hat{u} \cdot p_i}{d_i} = \hat{u} \cdot \sum_{i \in \mathcal{V}} \frac{p_i}{d_i}.$$  

The value of link reliability can be estimated based on some theoretical formulations according to the SNR value at the
receiver (AP), while in practical the actual link quality would be affected by other environmental factors, e.g., multi-path fading, which makes it hard to get the accurate estimate of $p_i$. To avoid such uncertainty, we eliminate the link reliability factor in our mathematical analysis, and focus on maximizing the following metric, as shown in Eq. (4). The link reliability factor will be taken into account at the last stage, i.e., advanced search.

$$\text{metric}' = \hat{u} \sum_{i \in V} \frac{1}{d_i} \tag{4}$$

Therefore, our goal is to determine the best location of the AP so as to maximize the achievable uploading throughput $\hat{u}$ and, in turn, enhance the overall satisfaction consequently. We see a step further into Eq. (2), and observe that maximizing $\hat{u}$ is equivalent to minimizing $\sum_{i \in V} \frac{1}{r_i}$. Then, by AM-GM Inequality Property, we get that

$$\sum_{i \in V} \frac{1}{r_i} \geq |V| \cdot \left( \prod_{i \in V} \frac{1}{r_i} \right)^{\frac{1}{|V|}} \tag{5}$$

The equilibrium holds when $(r_1 = r_2 = \ldots = r_{|V|})$. Based on the AM-GM Inequality Property, we obtain that the maximum $\hat{u}$ exists when all cameras select the same transmission bit-rate $r_i$, and the overall satisfaction can be maximized consequently. On the other hand, recall that each camera independently selects it bit-rate according to the link quality, which is closely correlated to the distance between itself and the AP. Hence, our problem can be transformed to determining the location of the AP that has similar distances to all the cameras such that all cameras can transmit at a similar bit-rate and achieve the best achievable uploading throughput.

This problem is quite similar to that of finding the circumcenter of a polygon if we consider each camera as a vertex. However, according to the geometry property, the circumcenter only exists in some special polygons, e.g., regular simple polygons, triangles, and rectangles, while other general polygons might not have a circumcenter. In those general cases, since the circumcenter might not exist, we cannot simply use the circumcenter as the AP location. Thus, in the following section, we will propose a mechanism to determine the AP location that has similar distances to all cameras in order to enhance each camera’s achievable uploading throughput $\hat{u}$. Despite that our relaxed problem (metric') might not result in optimal solution of the original goal in Eq. (1), we can greatly reduce the measurement overhead in the advanced search phase by using the AP location solved based on Eq. (4) as an initial measurement point.

### IV. Framework of Spring-Cam

In this section, we first describe the idea of the proposed mechanism, Spring-Cam, and detail how it determines the location of the AP in a WSS.

#### A. Spring-Cam in a nutshell

According to Section III, our goal is to find the AP location that has similar distances to all cameras. We can thereby reformulate the AP deployment problem as minimizing the variance of link distances as follows.

$$\frac{1}{|V|} \sum_{i \in V} (z_i - \bar{z})^2 \tag{6}$$

, where $z_i$ represents the distance between camera $i$ and the AP, and $\bar{z} = \frac{\sum_{i \in V} z_i}{|V|}$ is the average distance of all links. Ideally, if the circumcenter exists, the metric in Eq. (6) equals 0. However, if the circumcenter does not exist, the objective is to find an AP location with a similar distance $z_i$ for each camera $i \in V$ such that $\forall z_i$ approaches $\bar{z}$ and Eq. (6) can be minimized.

We observe that this metric is similar to the mass-spring system in Physics, which aims to strike the balance and minimize the potential energy generated by all springs. In our problem, if $z_i$ is larger than $\bar{z}$, the AP should be placed closer to camera $i$; otherwise, the AP should be placed away from camera $i$. Motivated by such observation, we map the AP deployment problem to a mass-spring problem in Physics. We define a virtual spring that connects each camera to the AP. Each camera generates a directed force to the AP through this spring. We view $z_i$ as the current length $l^c$ of the spring of camera $i$, while $\bar{z}$ as the rest length $l^r$. Accordingly, the difference between $z_i$ and $\bar{z}$ represents the displacement $\Delta z$ of the virtual spring, which is proportional to the force applied on the AP. We set the elasticity coefficient $k$ of a spring to 1 without loss of generality, and, therefore, the force $f_i$ generated by the virtual spring of camera $i$ can be represented as

$$f_i = k_i \cdot \Delta z_i = k_i \cdot (l^c_i - l^r_i) = 1 \cdot (z_i - \bar{z}) \tag{7}$$

The magnitude of the spring-force is determined by the absolute value of $\Delta z$. In addition, the sign of $\Delta z$ determines how the AP should be moved. Specifically, if $\Delta z > 0$, the virtual spring utilizes a pulling force that "pulls" the AP closer to the camera. On the other hand, if $\Delta z < 0$, the camera "pushes" the AP away from itself. The direction of the spring force can be further computed by assessing the geometric vector $v_i$ from the AP to camera $i$, which is expressed as

$$v_i = \frac{(x_i, y_i) - (x_{\text{AP}}, y_{\text{AP}})}{\sqrt{(x_{\text{AP}} - x_i)^2 + (y_{\text{AP}} - y_i)^2}} \tag{9}$$

Note that we utilize the unit vector, instead of the normal vector, because this vector is used only to specify the direction of the spring-force. The unit vector is exploited to keep the magnitude of the spring-force equal to $k_i \cdot \Delta z_i$. Finally, the directed spring-force $\vec{F}_i$ of camera $i$ can be expressed as

$$\vec{F}_i = f_i \cdot \vec{v}_i. \tag{10}$$

After all the spring-forces are calculated, we can compute the net force $\vec{N}$ applied on the AP by simply summing up all the directed forces. That is,

$$\vec{N}_{\text{AP}} = \sum_{i \in V} \vec{F}_i. \tag{11}$$
Given the net force $\vec{N}_{AP}$ generated by spring-forces of all cameras, a WSS can move the location of the AP with the magnitude of net force, $|\vec{N}_{AP}|$, toward the direction specified by $\vec{N}_{AP}$. The larger the $|\vec{N}_{AP}|$ value is, the longer the distance that the AP should be moved away from the current location. The whole procedure continues until the termination condition is met.

### B. Spring-Cam Implementation Issues

In the last section, we briefly describe the rationale behind the design of Spring-Cam. We now discuss the implementation issues of realizing the proposed mass-spring system in WSSs. In order to adopt the mass-spring system to solve the AP deployment problem, we must address the following components.

- **Initialization**: What initial location should the AP be placed in the beginning of Spring-Cam’s operation?
- **Adjustment**: How does Spring-Cam adjust the location of the AP according to the net force generated by the virtual springs of cameras?
- **Termination**: What is the termination condition of Spring-Cam adjustment? How long does it take until Spring-Cam reaches the final state?
- **Advanced Search**: Can Spring-Cam be improved by searching around the potential AP location outputted by Spring-Cam? How good does it improve if we further search a better AP location based on the result of Spring-Cam?

We will discuss each of the above components subsequently.

1) **Initialization**: The initial AP location could be set at the corner of the area of interest, the average of the locations of all cameras, the average of the locations of all cameras weighted by their demand rates, or any random selected location. In general, Spring-Cam works well no matter where the initial AP location is because the adjustment procedure will update the location to approach a suitable AP location. However, a good selection of the initial point might result in a short convergence time of the adjustment procedure.

Figures 1 and 2 are the simulation results that analyze the impacts of various initialization strategies. In this simulation, we compare five initialization methods, i.e., the origin (corner), the central point of the monitored area, the average of the locations of all cameras, the average of the locations of all cameras weighted by their demand rates, or any random selected point. We can see that the origin and the random approaches result in a worse performance, while the others achieve similar results. In addition, since these two methods may start from a point that is far from the final potential AP location, they require more iterations to converge to the termination condition. Based on this observation, in this work, we set the average of the locations of all cameras as the starting point of our proposed Spring-Cam algorithm.

2) **Adjustment**: In order to minimize the distance variance as shown in Eq. (6), Spring-Cam repetitively adjusts the AP location until it converges to the termination condition. In each iteration, given the current location of the AP and the locations all cameras, Spring-Cam computes the net force applied on the AP based on Eq. (11). The net force is a vector composed of the magnitude and the direction. Basically, Spring-Cam adjusts the current location of the AP based on the net force; that is, the AP is set $\vec{N}_{AP}$ apart from the current location toward the direction of $\vec{N}_{AP}$. Specifically, the AP location $(x'_{AP}, y'_{AP})$ can be updated by simply adding the net force vector $\vec{N}_{AP}$ to the current AP location $(x_{AP}, y_{AP})$ as follows.

$$ (x'_{AP}, y'_{AP}) = (x_{AP}, y_{AP}) + \vec{N}_{AP} $$

(12)

However, the $|\vec{N}_{AP}|$ value varies with the number of cameras. The higher the number of the cameras, the larger the $|\vec{N}_{AP}|$ value could be. In order to eliminate this effect, we slightly modify Eq. (12), and divide it by the number of cameras in a WSS as follows.

$$ (x'_{AP}, y'_{AP}) = (x_{AP}, y_{AP}) + \frac{\vec{N}_{AP}}{|V|} $$

(13)

As a result, the distance variation can be reduced at each iteration. This procedure continues until the distance variation cannot be further minimized.

3) **Termination**: In the adjustment procedure, Spring-Cam attempts to reduce the distance variation defined in Eq. (6) during each iteration. We set the termination condition as the...
state in which the value of Eq. (6) cannot be reduced by the adjustment any more. Let \( z'_{i} \) denote the distance between camera \( i \) and the AP location generated in the previous iteration of the adjustment procedure, and \( z_{i} \) represent the distance between camera \( i \) and the newly-selected location at the current iteration. Spring-Cam’s adjustment procedure keeps moving the AP to a location that has smaller distance variation until the following termination condition is met.

\[
\frac{1}{|V|} \sum_{i \in V} (z_{i} - \bar{z})^2 = \alpha \cdot \frac{1}{|V|} \sum_{i \in V} (z'_{i} - \bar{z'})^2
\]

(14)

where \( \alpha \) is a parameter controlling how sensitive the adjustment procedure could be terminated, and can be set by system developers based on their requirements. Note that this termination condition does not guarantee that Spring-Cam halts at the optimum AP location in the area. Specifically, this method may return local optimum point. However, we adopt this approach because it is simple and easy to converge.

Figure 2 shows the average number of iterations required to converge to the termination condition. From the results we can see that, as compared to exhausted search over each possible location, Spring-Cam only takes a few steps to converge to the final location. Although Spring-Cam requires a few more iterations than that simply using the average location of all cameras, we will later verify that Spring-Cam achieves a better performance with such a reasonable overhead in Section V.

4) Advanced Search: Since we eliminate the link reliability factor \( p_{i} \) in Eq. 4, we now consider the actual link reliability by real measurements via advanced search. In order to enhance the overall satisfaction of WSS, Spring-Cam can further find a better AP location by searching the nearby area of the potential AP \((x_{AP}^{p}, y_{AP}^{p})\) solved in the adjustment procedure, to execute real measurement and determine the best result in terms of Eq. (1). For example, if we allow the system deployer to measure a 10m×10m square area, which has a center point \((x_{AP}^{p}, y_{AP}^{p})\), the system deployer must measure the receipt throughput of each camera in this 10m×10m area. It is evident that there is a trade-off between the improvement and the size of the testing area. The larger the testing area is, the better location perceiving a higher video quality it could locate, while it also increases the cost of measuring the quality of each distinct location in the targeting area. By conducting advanced search, Spring-Cam can find a better AP location by taking the actual link reliability into consideration.

In Section V, we will validate that Spring-Cam achieves a fair performance without the procedure of advanced search. Besides, the performance can be significantly enhanced if advanced search is applied to measure a limited size of the area surrounding the potential location. Since Spring-Cam has already spot a fair location of the AP, it only takes little effort to find out a better AP location. In some cases, Spring-Cam with advanced search can even find out the same best AP location as that determined by exhausted search.

V. PERFORMANCE EVALUATION

In this section, we conduct simulations to evaluate the performance of the proposed Spring-Cam. We set the topology area as a 350m×350m square field. We distribute cameras in the square field uniformly at random, and assign each camera a demanding streaming rate, which is selected at random from [500,10000] (Kb/s) uniformly. The performance metric applied here is the same as that defined in Eq. (1).

We compare the performance of Spring-Cam with that of the following AP deployment strategies. The first method is exhausted search over the whole area; that is, we put the AP on each coordinate, and measure the achievable uploading throughput of each camera for each different coordinate. Hence, this method requires 350 × 350 iterations, i.e., measurements, to find out the best AP location. We use this as the oracle and consider it as the optimal result of the AP deployment problem. The second method simply puts the AP at the average location of all cameras, which is also the starting point of Spring-Cam. By comparing Spring-Cam with the second method, we can demonstrate how much performance gain Spring-Cam can achieve.

For advanced search proposed in Spring-Cam, we apply exhausted search over the area around the potential AP location determined by Spring-Cam, i.e., \((x_{AP}^{p}, y_{AP}^{p})\). For instance, \(Spring-Cam+5\) returns the best AP location within the area of \((x_{AP}^{p}±5, y_{AP}^{p}±5)\), while it costs \((2 \cdot 5 + 1)^{2}\) measurements to find the local optimal location. We also conduct simulations to demonstrate the trade-off between the performance and the measurement overhead.

Figure 3 shows the performance of all methods for various number of cameras in a WSS. As the number of cameras in a WSS increases, the performance degrades gradually because more transmission links get involved in wireless channel contention. In addition, the slopes of degradation decrease as more cameras are deployed in the system. We can also learn from this figure that the performance gap between Spring-Cam and the average location is up to 23%, showing that Spring-Cam helps to find a better AP location without any measurement overhead. By further applying advanced search,
Spring-Cam is able to achieve a better performance that is close to the results of the oracle when the area of advanced search, i.e., the number of measurements, increases.

Figure 4 shows the average hit ratio of all methods. If the AP location generated by a method matches the optimal AP location determined by exhausted search (oracle), we call it “hit”; otherwise, we call it “fail”. Therefore, the hit ratio is defined as the proportion of the “hit” scenarios. The figure shows that increasing the area of advanced search also increases the probability of finding the same AP location as the oracle result. Another observation is that the hit ratio decreases as the number of cameras increases. This is because Spring-Cam might converge to a local minimum location when there are more virtual springs (i.e., cameras) generating the forces on the AP. However, the larger area is tested by the advanced search, the sooner Spring-Cam converges to a stable hit ratio.

We conduct simulations to study the trade-off between the hit ratio and the overhead required by the advanced search. Figure 5 depicts the result of dividing the hit ratio by the number of measurement overheads of each method. Despite the fact that Spring-Cam+5 has the highest Hit Ratio / Overhead, its value varies under different numbers of cameras. As the area of the advanced search gets larger, Spring-Cam obtains a flatter curve, which shows that Spring-Cam along with the wider advanced search can ensure the system deployer to achieve the stable performance gain.

VI. CONCLUSIONS
This paper investigated the AP deployment problem in WSSs. The objective is to deploy the AP in a location that can best satisfy the demanding uploading streaming throughput of each camera. Specifically, given a specific location of the AP, we analyze the achievable uploading throughput that the AP can receive from each camera. We then propose a mechanism, called Spring-Cam, which adopts the concept of mass-spring systems in Physics to generate a suitable AP location with a reduced measurement overhead. Simulation results show that Spring-Cam can find an AP location that achieves a performance comparable with exhausted search without any measurement overhead. By searching the area surrounding the potential AP location, Spring-Cam can significantly improve the performance in terms of the overall satisfaction with a reasonable measurement overhead. In the future, we will extend our work to support the deployment of multiple sinks (APs) over a multi-channel wireless network.

REFERENCES