

Article

## Surveillance of a 2D Plane Area with 3D Deployed Cameras

Yi-Ge Fu \*, Jie Zhou and Lei Deng

Department of Automation, Tsinghua University, Beijing 100084, China;

E-Mails: jzhou@tsinghua.edu.cn (J.Z.); dengl09@mails.tsinghua.edu.cn (L.D.)

\* Author to whom correspondence should be addressed; E-Mail: iamdafu@gmail.com;

Tel.: +86-138-1045-9761.

Received: 30 November 2013; in revised form: 16 January 2014 / Accepted: 17 January 2014 /

Published: 24 January 2014

---

**Abstract:** As the use of camera networks has expanded, camera placement to satisfy some quality assurance parameters (such as a good coverage ratio, an acceptable resolution constraints, an acceptable cost as low as possible, *etc.*) has become an important problem. The discrete camera deployment problem is NP-hard and many heuristic methods have been proposed to solve it, most of which make very simple assumptions. In this paper, we propose a probability inspired binary Particle Swarm Optimization (PI-BPSO) algorithm to solve a homogeneous camera network placement problem. We model the problem under some more realistic assumptions: (1) deploy the cameras in the 3D space while the surveillance area is restricted to a 2D ground plane; (2) deploy the minimal number of cameras to get a maximum visual coverage under more constraints, such as field of view (FOV) of the cameras and the minimum resolution constraints. We can simultaneously optimize the number and the configuration of the cameras through the introduction of a regulation item in the cost function. The simulation results showed the effectiveness of the proposed PI-BPSO algorithm.

**Keywords:** camera network placement; coverage; particle swarm optimization (PSO); task constraints; camera constraints; scene constraints

---

### 1. Introduction

Camera networks are used in many novel applications, such as video surveillance [1], room sensing [2], smart video conferencing [3], *etc.* There are some challenge issues in the study of camera networks,

such as how to get an optimized camera network coverage, how to design a scalable network architecture, how to determine the trade-off between QoS requirements and energy costs [4], *etc.* Among the problems mentioned above, camera network coverage problem is a central issue, which has interested many researchers [5–9] and coverage rate is one of the most important performance metrics for camera network surveillance utility. Thus, determining the appropriate placement of the cameras to achieve the maximum amount of visibility becomes an important issue in designing camera network arrangements.

The camera network deployment problem can be defined as how to place the cameras in the appropriate places to maximize the coverage of the camera network under some constraints. The constraints can be categorized into three main types: task constraints, camera constraints and scene constraints. The task constraints include continuous tracking (enough overlap between cameras), people identification (image resolution and focus), complete coverage of the surveillance area (field of view of each camera) and so on. The camera constraints include the camera network type (a homogeneous or heterogeneous camera network), camera type (PTZ or static camera), camera intrinsic parameters (focus length, CCD size, *etc.*) and so on. The scene constraints include the surveillance area (2D or 3D, with or without holes, simple polygon or not), the positions where the camera network can be located (2D or 3D, in the walls, in the ceiling, at the same height or anywhere) and so on.

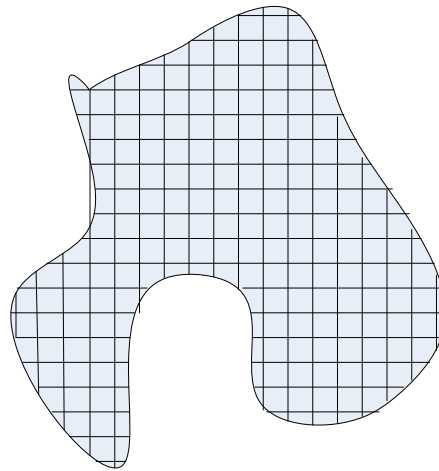
Since the camera network placement problem is a NP-hard combinatorial optimization problem [10], simple enumeration and search techniques will meet great difficulty in determining optimal placement configurations. There are many approximate techniques for solving optimal camera placement problems, such as the greedy based method, sampling based method *etc.* Zhao *et al.* have provided an excellent survey of the approximate techniques in [11]. Lately, some researchers have proposed some evolution-based optimization methods, such as PSO [12], BPSO-PI [13] and ABC [14].

There are some weaknesses in these results, mainly due to the overly simple assumptions used. From the perspective of the three constraints mentioned above, we give a brief explanation of the limitations of the earlier works. From the perspective of task and camera constraints, most of the works only consider the coverage of the area while the video resolution and focus are seldom considered; From the perspective of scene constraints, most of the scenes are modeled as a 2D case which is too simple to conduct the real camera network placement, or modeled as a 3D case which is too restrictive because in most of the cases we are only concerned with the surveillance plane area.

We give several examples. The surveillance area of [13] is modeled as a rectangle in the 3D cases while we know that in the real circumstances it is a trapezoid which is sensitive to the orientation of the cameras. The constraints in [14,15] only include the coverage rate (FOV is considered), while the resolution and focus are out of the scope of the articles.

In this paper, we consider the deployment of homogeneous camera network in the 3D space to surveil a 2D ground plane. For simplicity considerations, the surveillance plane is modeled as a rectangle area which is not essential to our work. We separate the surveillance plane area into  $n$  grids, as illustrated in Figure 1. We assume that the probability of choose each grid is the same  $1/n$  and the coverage ratio  $p$  can be determined by sampling as illustrated in the next section.

**Figure 1.** The surveillance plane area is divided into  $n$  grids no matter the shape of the area.



We take a more synthetic constraints set, including the surveillance video resolution, video focus, the camera field of view *etc.*, into consideration. Under the constraints, we propose a probability-inspired particle swarm optimization algorithm to get the optimized camera network placement configuration.

The main contributions of this paper can be summarized as follows:

- We consider a more realistic problem in that we deploy the cameras in a 3D space to surveil a plane area. Some of the previous works consider the problem in a 2D plane and the FOV of the camera is modeled as a sector which is too simple an assumption, while some works consider the problem in the 3D space and model the FOV of the camera as a cone which is too restrictive an assumption. We can get a more accurate result to solve the camera deployment problem in the 3D space to surveillance of a 2D plane and instruct the real life camera network placement;
- We take more constraints into consideration than others, including resolution, focus, FOV. Most of the previous works only consider the FOV of the camera to get a good coverage while the other constraints are important to get a good surveillance video;
- We propose a probability-inspired PSO algorithm to solve the camera network placement problem heuristically. In the algorithm, we introduce a regulated item in the fitness function to optimize the coverage ratio and the number of the cameras simultaneously. The experimental results show the effectiveness of the algorithm.

The rest of the paper is organized as follows: we review recent progress in camera network deployment in Section 2. In Section 3, we give the camera network placement problem from three perspectives. In Section 4, we propose a PI-BPSO algorithm and discuss the representation and fitness computation of particles in the algorithm. Simulation results are given in Section 5. We give the conclusions and discuss some future work in the last section.

## 2. Related Works

In Computational Geometry, there is a well-known problem called the Art Gallery problem (AGP) [16] and some of its variations which are very similar to the problem of camera placement in camera

networks. The aim of the camera network deployment is to provide full coverage of the surveillance area with the minimal cost, and the aim of the AGP is to monitor an art gallery with the least number of guards located at different locations in order to make sure that every point in the museum is seen by at least one guard. The difference between the two problems is mainly attributable to the assumption on the ability of the “guard” as the AGP assumes the guards have unlimited field of view, infinite depth of field, and have infinite precision and speed, while the cameras in the real world don’t have these abilities. Even if we make the unrealistic assumption of the ability of the guard, the problem is still proved to be NP-hard [17,18]. Though the AGP and its variants can’t give an exact answer to the camera network placement problem, they do give some insights into the problem, the visibility graph and the lower bound of the guard numbers. There are some good results in the AGP field which we will not introduce here in detail. We refer the reader to an excellent book on the topic [16].

The other interesting research problem related to the camera network placement problem is the wireless sensor network (WSN) placement problem [4,7,19–22]. To be strict, the camera is known as a visual sensor, but because the camera is a kind of directional sensor, this leads to some differences between the two disciplines. We refer the reader to the review [6] about the WSN placement problem.

Researches on the camera network placement problem can be divided into two disciplines, one is the coverage problem and the other is the optimization problem. The two problems can be related through the optimization framework issued by Zhao *et al.* [11], where they present that the camera network placement can be divided into two broad categories, the MIN and FIX problems. The MIN problem is to find the minimum cost cameras to satisfy the minimum coverage ratio, and the FIX problem is to find the maximum the coverage subject to a fixed number of cameras.

The visual coverage of the camera network describes what can be seen and what can’t in the surveillance area. It is so fundamental to many computer vision tasks that different works have suggested different visual coverage models according to the different surveillance tasks.

The standard coverage model is defined as the area of the surveillance area. It can be classified as two cases, from the perspective of the area and the camera. From the perspective of the area [13,23], the surveillance area is discrete into some grid and if the center point of a grid can be seen from a camera (under some limitation such as resolution and DOF *etc.*), then we say that the grid can be seen from the camera. From the perspective of the camera [24,25], the area that a cameras can monitor is determined by the camera’s FOV, DOF, resolution and the camera network placement problem is turned into a set cover problem. To get a continuous consistently labeled trajectory of the same object, Yao *et al.* [26] add handoff rate analysis to the standard coverage models. To improve the full coverage of events and objects/events recognition, Newell *et al.* [27] give an multi-perspective coverage (MPC) model where the coverage is calculated based on the  $\omega$ -perspective coverage (the number of perspectives that cover the event). Based on the MPC model, Yildiz *et al.* [8] give an angular coverage model where the coverage of an object is defined as the object can be seen from different perspectives that span 360°. We refer the reader to the excellent survey [5]. Most of the work model the area as a plane area and the camera is placed in the plane which is too simple to guide the real camera network placement.

As we have stated that the camera network placement problem is NP-hard, researchers have put forward various approximate optimization algorithms to solve the problem [11,13–15]. Chrysostomou *et al.* [14] propose a bee colony algorithm as the optimization engine to determine the

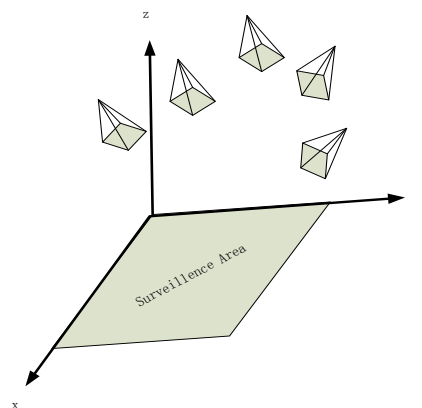
minimum possible cost (minimum number) of cameras to cover the given space under some camera placement constraints such as geometrical, optical, as well as reconstructive limitations and this delivers promising preliminary results. Morsly *et al.* [13] propose a Binary Particle Swarm Optimization Inspired Probability (BPSO-IP) algorithm as the optimization engine to ensure the accurate visual coverage of the monitoring space with a minimum number of cameras. The authors also give a detailed comparison between the BPSO-IP algorithm and other evolutionary-like algorithms such as BPSO, Simulated Annealing (SA), Tabu Search (TS) and genetic techniques based algorithms to solve the camera network placement problem. Lee *et al.* [15] give a generic algorithm as the optimization engine to solve the camera network placement problem which is modeled as a multi objective optimization problem. Zhao *et al.* [11] put forward a framework to compare the accuracy, efficiency and scalability of the greedy-based method, heuristics-based method, sampling-based method and LP and SDP relaxation-based method.

### 3. Problem Definition and System Model

We put forward an optimal camera network placement problem to satisfy the need of different surveillance tasks on a specific surveillance area. The problem can be modeled as a multi objective optimization problem that must satisfy multiple constraints. In this work, we are interested in the static camera network placement problem, where the objective is to determine the number of cameras, their positions and poses for an rectangle surveillance area, given the intrinsic parameters of the cameras (such as focal length, the diameter of the lens's aperture  $a$ , the minimum dimension of a camera pixel  $c$ , the resolution  $R$ , *etc.*) and a set of task-specific constraints (such as the resolution constraints, the focus constraints and visibility constraints *etc.*). Commonly, this camera network placement problem takes place off-line to support the task-specific requirements of on-line computer vision surveillance systems. But occasionally we shall adjust the layout of the camera network on-line to support the different surveillance tasks.

In the real world circumstances, we often layout some linked cameras to surveillance a square which is the main subject of this article. We establish a world frame as that: we model the surveillance square as the  $XOY$  plane and the upward direction as the  $Z$  axis which is shown in Figure 2.

**Figure 2.** The coordinate frame we establish to model the surveillance scene where the ground plane (the surveillance area) is modeled as the  $XOY$  plane and the anti-gravity direction is modeled as the  $Z$  direction.



### 3.1. Camera Modeling

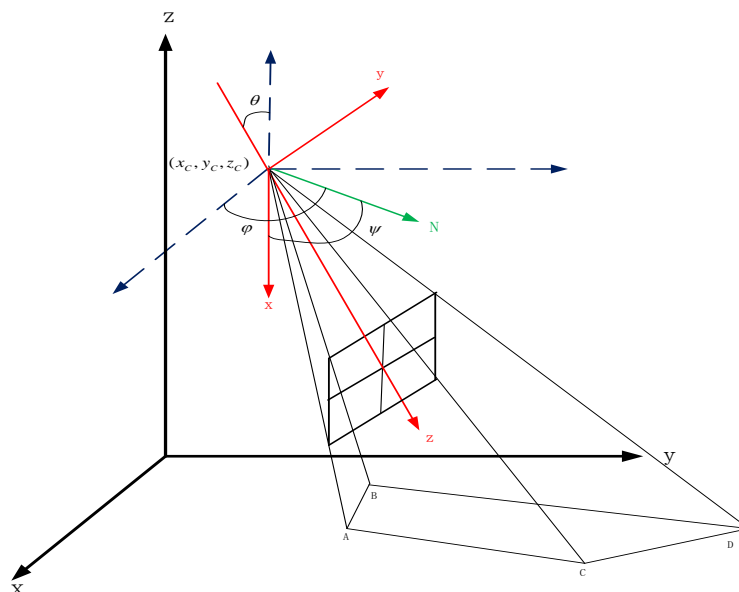
We assume that the various cameras used in the layout share the same intrinsic parameters, such as the resolution of the camera is  $R(R_h, R_v)$ , the horizontal and vertical dimensions of the size of the Coupled Charge Detector (CCD) element is  $s(h, w)$ , the focal length of the camera is  $f$ , etc. In our work we are mainly interested in the video resolution constraints, which is very common in surveillance tasks such that we can identify some persons in the surveillance video, and the focus constraints which are necessary to get a clear surveillance video.

#### 3.1.1. Modeling a Camera’s FOV (Field of View) in 3D Space

We use a pyramid to represent the camera’s FOV, which is shown in Figure 3. We are interested in the homogeneous camera network placement problem in this article. We set  $K$  as the intrinsic parameter matrix of the cameras in the network. The camera’s position  $C(x_c, y_c, z_c)$  and rotation  $R(\phi, \theta, \psi)$  describe the camera’s extrinsic parameters where  $R(\phi, \theta, \psi)$  is the Euler angles representation of the rotation matrix  $R$  [28] and  $(\phi, \theta, \psi)$  represents the camera’s yaw, pitch and roll angles respectively (Figure 3) and we have:

$$R = \begin{bmatrix} \sin \psi & -\cos \psi & 0 \\ \cos \psi & \sin \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} -\sin \phi & \cos \phi & 0 \\ -\cos \phi & -\sin \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

**Figure 3.** The FOV of the camera in 3D space and the camera’s yaw, pitch, roll angles which is shown as  $(\phi, \theta, \psi)$ . The vector  $N$  is the intersect of the  $XOY_{camera}$  plane and the  $XOY_{world\_t(C)}$  (the coordinates that transform the world coordinate to the point  $C$ ) plane.



For the surveillance application, we desire that the angle between the object and the direction of the camera is less than a constant angle  $\theta_0$  (which is critical for feature point extraction and the other applications), such as  $60^\circ$ . We assume that the object in the surveillance video is vertical to the ground

(which is almost true because Pisa tower is seldom), then we get a constraints on the  $(\phi, \theta, \psi)$  that  $\theta \leq \theta_0$ . As Figure 3 shows, the surveillance area of the camera is  $\overline{ABCD}$ , and the coordinates of the 4 vertexes can be determined by:

$$\begin{aligned} \begin{bmatrix} -\frac{h}{2} & \frac{h}{2} & \frac{h}{2} & -\frac{h}{2} \\ -\frac{w}{2} & -\frac{w}{2} & \frac{w}{2} & -\frac{w}{2} \\ 1 & 1 & 1 & 1 \end{bmatrix} &= K \begin{bmatrix} R & -\overline{C} \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \\ &= K \begin{bmatrix} R_1 & R_2 & -\overline{C} \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \\ &= P \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{aligned} \quad (2)$$

where  $h, w$  is the CCD height and width,  $K$  is the intrinsic matrix of the camera and  $(x_i, y_i) \ 1 \leq i \leq 4$  is the coordinates of the vertexes of  $\overline{ABCD}$ . We can transfer the above equation to a more simple one:

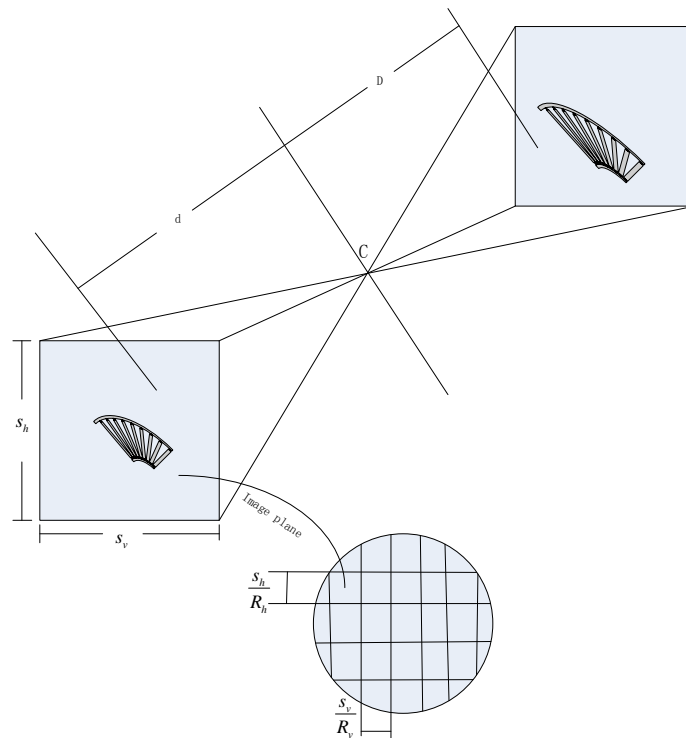
$$\begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ 1 & 1 & 1 & 1 \end{bmatrix} = P^\dagger \begin{bmatrix} -\frac{h}{2} & \frac{h}{2} & \frac{h}{2} & -\frac{h}{2} \\ -\frac{w}{2} & -\frac{w}{2} & \frac{w}{2} & -\frac{w}{2} \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (3)$$

If we know that a camera's configuration  $(R \ -\overline{C})$ , we can get the coordinates of the four vertexes of the quadrangular surveillance area in the plane. Then we can determine whether the point  $P$  in the plane can be covered by the configuration use  $P \in \overline{ABCD}$ .

### 3.1.2. Modeling the Resolution Constraints in 3D Space

In this section we describe the relationship between the surveillance video resolution constraints and the camera's position. For a specific surveillance camera, the required resolution provides an upper bound on the distance between the camera and the surveillance area. Figure 4 illustrates the image process of an object  $S$  lying at distance  $D$  from the lens center, where the distance between the image and the lens center is  $d$ . There is a relationship between  $d$ ,  $D$  and the lens focal length  $f$  by the Gaussian lens equation:

$$\frac{1}{d} + \frac{1}{D} = \frac{1}{f} \quad (4)$$

**Figure 4.** Camera and spatial resolution for the camera.

The surveillance video resolution of object  $S$ , described as the pixels per unit length, depends on the direction in which it is measured. It is easy to see that the maximum resolution occurs along the rows or columns of pixels and the minimum occurs along the diagonal of each pixel. In consideration of simplicity, we assume that the desired resolution constraints is  $\alpha$  pixels per unit length corresponding to the diagonal of each pixel, then:

$$\frac{1}{D_{Max}} = \frac{\alpha \sqrt{\left(\frac{h}{R_h}\right)^2 + \left(\frac{w}{R_v}\right)^2}}{d} \quad (5)$$

The distance between the position of the camera lens's center and the surveillance area  $D$  must satisfy:

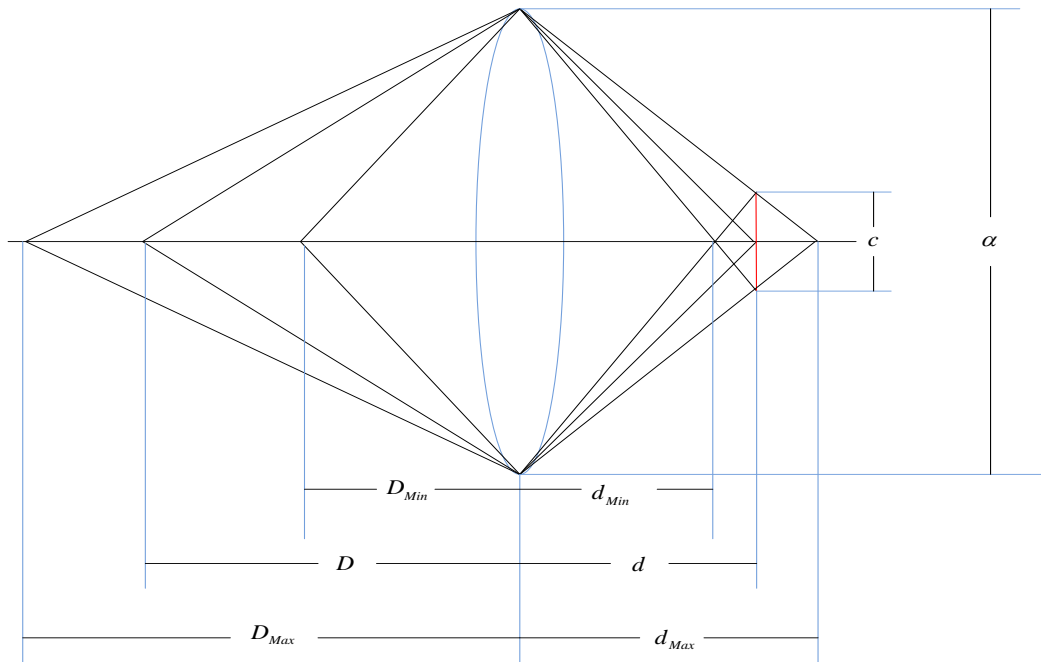
$$D \leq D_{max} \quad (6)$$

### 3.1.3. Modeling the Focus Constraints in 3D Space

In this section, we determine the constraints on the camera's viewpoints when we request that all the points of the surveillance area must be sharp (in focus) in some surveillance camera's FOV. For any camera, there is only one plane on which the camera can precisely focus, and the point object in any other plane is imaged as a disk (known as the blur spot) rather than a point. When the diameter of blur spot is sufficiently small, the image disk is indistinguishable from a point. The diameter of the blur spot is known as the acceptable circle of confusion, or simply as the circle of confusion (CoC). We can easily induce that there is a region of acceptable sharpness between two planes on either side of the plane of focus which is illustrated in Figure 5. The region is known as depth of field (DOF).



Figure 5. DOF for the camera.



Here we assume the *CoC* is the minimum dimension of a camera pixel  $c$ , The focus distance is  $D$ , the lens's focal length  $f$ , the diameter of the lens's aperture  $\alpha$ , relative aperture (f-number) of lens  $N$ . We can determine the maximum distance  $D_{Max}$ , and the minimum distance  $D_{Min}$  as follows [29]:

$$D_{Max} = \frac{Df^2}{f^2 - Nc(D - f)} \tag{7}$$

$$D_{Min} = \frac{Df^2}{f^2 + Nc(D - f)} \tag{8}$$

$$N = \frac{f}{\alpha} \tag{9}$$

If we want the video to be sharp, then:

$$D_{Min} \leq D \leq D_{Max} \tag{10}$$

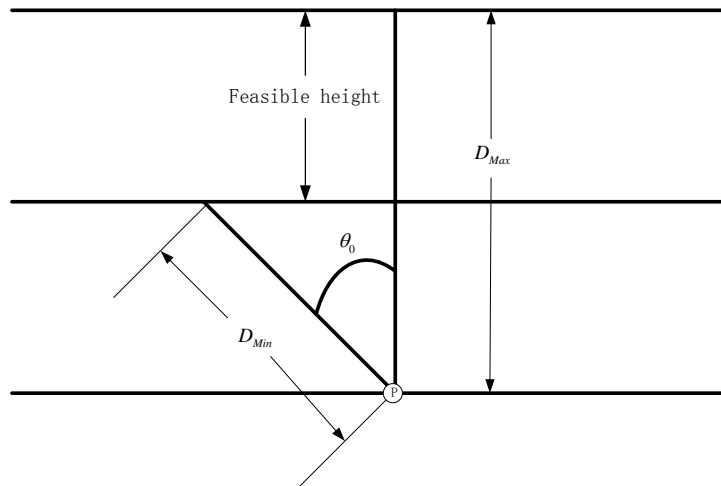
Then we have that the height of the camera must satisfy (illustrated in Figure 6):

$$D_{Min} \cos\theta_0 \leq H \leq D_{Max} \tag{11}$$

### 3.2. Space Modeling

In theory, cameras can be located anywhere in the space since the camera position variables  $x_C, y_C, z_C$  and the pose variables  $\varphi, \theta$  and  $\psi$  are all continuous variables. In practice, we ordinarily restrict the selection of the cameras' location and pose in a discrete parameter space which is determined by the spatial sampling in the continuous parameter space. So we can transform the continuous optimization problem to a discrete optimization problem. We should state that as the spatial sampling frequencies  $f_x, f_y, f_z, f_\varphi, f_\theta$  and  $f_\psi \rightarrow \infty$ , the approximated discrete solution converges to the continuous-case solution.

**Figure 6.** The feasible height of the cameras. We can see that the point  $P$  will be not sharp if the height of the camera's height  $H$  don't satisfy  $D_{Min} \cos\theta_0 \leq H \leq D_{Max}$ . That is to say that  $D_{Min} \cos\theta_0 \leq H \leq D_{Max}$  is a necessary condition for the point  $P$  in sharp.  $\theta_0$  is the biggest feasible angle between the direction of the camera and the object which is discussed above.



In the real situation, there are some other different constraints on the allowed deployable area such as the constraints that the cameras must be located on the walls of the indoor surveillance environment or the constraints that the cameras must be located at some restricted height for the consideration of their management, but in this article we only restrict the position of the cameras according to the constraints of surveillance video resolution, FOV (field of view) and DOF (depth of field), which are described in the previous section.

### 3.3. Camera Network Placement Problem Modeling

We can model the camera network placement as an optimization problem which is defined as maximization of some utility function given some cameras and task constraints. From the definition of the model, we can analyze the problem from four perspectives: the utility function (or cost function), the task constraints, the space constraints and the camera's parameters. Let  $T$  be the given task and let  $C_T$  be the set of all constraints imposed by the task  $T$  as: quality of service constraints like resolution constraints and focus constraints, square coverage constraints, camera network overlap ration constraints *etc.*

Let the vector  $C(C^I, C^E)$  represents the camera's intrinsic and extrinsic (position and pose) parameters. The problem is to find where to layout the set of cameras in the specific area  $A$  to minimize the cost function under the set of constraints  $C_T$ :

$$\arg \min_C G(C) \text{ where } C \in A \cap C_T \quad (12)$$

where  $C = (C_i, i = 1, 2, \dots, N)$  is the cameras we want to place in the surveillance area  $A$ ,  $G(C)$  is the cost function.

In this work we are mainly interested in the 2D plane surveillance by a camera network located in a 3D environment. In this special surveillance scene, we want to place the minimum number of cameras

in the optimal locations (position and pose) to maximize the visual coverage under the surveillance task constraints. Based on the assumptions we make above, we give the following specific model to solve this visual sensor placement.

Suppose that we have  $N$  cameras  $C = (C_i, i = 1, 2, \dots, N_C)$  in the network, and the positions and poses that we can place our cameras are  $S = (S_i, i = 1, 2, \dots, N_S)$  (which are determined by the space modeling we discuss in the section above), and the surveillance area  $A$  is modeled as a surveillance space  $T = (T_i, i = 1, 2, \dots, N_T)$ . The surveillance space has different forms according to the different tasks. In the situation we are interested in this article, we reduced the area to a rectangular grid and choose the middle point of each grid  $(T_i, i = 1, 2, \dots, N_T)$  as the representation of the surveillance area.

We define a binary variable  $\{b_{i,j} : i = 1, \dots, N_S, j = 1, \dots, N_C\}$  to represent the camera network configuration as follows:

$b_{i,j} = 1$  if a camera  $C_j$  is placed in a location  $S_i$

$b_{i,j} = 0$  otherwise

We define a binary variable  $\{x_{i,j} : i = 1, \dots, N_T, j = 1, \dots, N_C\}$  to represent the surveillance utility of the camera network.

$x_{i,j} = 1$  if a camera  $C_j$  can see the object  $T_i$

$x_{i,j} = 0$  otherwise

Based on the notations we define above, we define the utility function  $G(C)$  as:

$$G(C) = \mu(C) + \lambda f(N_C) \quad (13)$$

where  $\mu(C)$  is the surveillance utility function of the camera network which is increase as the number of the cameras increase. In our experiment, we set  $\mu(C) = p$  where  $p$  is the ratio of the surveillance area:

$$p = \frac{\sum_{i=1}^{N_T} \max_j(x_{i,j})}{N_T} \quad (14)$$

When the number of grids in the surveillance area is too large, then we can apply a sample method to determine the coverage. When we choose  $m$  grids to check if they are covered by the camera network and  $k$  grids is covered then we can see that the coverage rate is:

$$p = \frac{k}{m} \quad (15)$$

$f(N_C)$  is a regulation item which decreases as the number of the cameras increases, so it can be a punitive item to the number of the cameras. In our experiment, we set  $f(N_C) = \frac{1}{N_C}$ .

Based on the augment above, we turn the camera network placement problem to an optimization problem as follows:

$$\arg \max_C \left( \frac{\sum_{j=1}^{N_T} \max_i(x_{ij})}{N_T} + \lambda \frac{1}{N_C} \right) \quad (16)$$

Satisfies:

$$\sum_{j=1}^{N_c} b_{i,j} \leq 1 \text{ one location only has one camera}$$

$$\sum_{i=1}^{N_s} b_{i,j} \leq 1 \text{ a camera can only be placed in one state (position and pose)}$$

$$\frac{\sum_{j=1}^{N_T} \max_i(x_{ij})}{N_T} \geq p_0 \text{ the surveillance ration must exceed } p_0.$$

#### 4. Optimization Method

Evolutionary computation technique, motivated by the evolution of Nature, is a powerful tool to approximately solve many NP-hard problems. There are many kind of evolution computing methods, such as genetic algorithm (GA) [30], differential evolution (DE) [31], artificial bee colony (ABC) [32,33]. The differences between different evolution algorithms mainly result from the different observations on Nature and the methods used to get new solutions from the old hypothesis. Among the various evolutionary computing methods, Particle Swarm Optimization (PSO) [34] inspired by the manner that a flock of birds or fishes exhibit a coordinated collective behavior during the travel, is an optimization tool used to deal with various optimization problems [13,35].

The PSO algorithm consists of a population of agents called particles, each of which is a potential solution to the optimization problem. The particle  $i$ , has a memory to record the current (time  $t$ ) position (solution)  $x_i^t$ , the previous personal best position  $pb_i^t$  and a velocity  $v_i^t$  at which speed the particle fly to the next position. We assume that the global best position of the group of particles is  $gb^t$  until time  $t$ . The position and velocity of each particle in standard PSO algorithm are updated as the followed equations:

$$x_i^{t+1} = x_i^t + v_i^t \quad (17)$$

$$v_i^t = \underbrace{\omega v_i^{t-1}}_{inertia} + \underbrace{\varphi_1 U_1^t (pb_i^t - x_i^t)}_{personal} + \underbrace{\varphi_2 U_2^t (gb^t - x_i^t)}_{society} \quad (18)$$

where  $\omega$  is the inertia weight,  $\varphi_1$  and  $\varphi_2$  are the weights of the personal influence and the society influence,  $U_1^t$  and  $U_2^t$  are two random numbers uniformly distributed in the interval  $[0,1]$ . There are many variants of the standard PSO algorithm, and the main differences between them lies in the velocity update strategy used.

Despite the different form of the various PSO algorithms that can be used, the main task during the solving process is to determine the representation of the particles and the fitness of each particle. That is to say, to solve the camera network placement with the PSO approach, we should build a mapping between the camera network deployment solution to the particle state and calculate the fitness of each particle's state. Because we represent the solution space in a discrete space, we must have a mechanism to transform the new position of the particle to a legal place. Like any optimization method, the initialization is very important for the convergence speed and solution quality. We will illustrate the important factors above in the next sections.

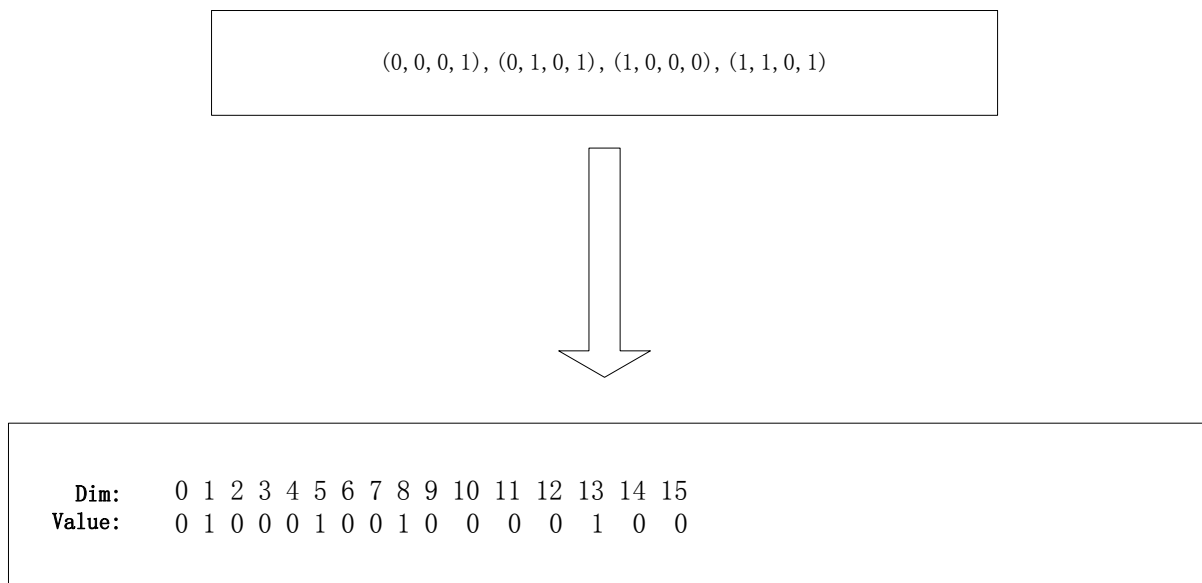
4.1. Representation

In this section, we describe the state representation of the camera network placement problem. Each of the PSO particles is represented as a  $f_x \times f_y \times f_z \times f_\phi \times f_\theta \times f_\psi$  dimensional array, where each column is a binary value representing the position and the orientation of the camera to be deployed. From the definition, we know that the number of 1 in the array represent the number of the cameras deployed in the network. The PSO population of the particles can be set to a const, such as 10,000.

Next we show how to represent a camera network placement problem as an example. Suppose that we deploy four cameras to surveil a rectangular area and the cameras can be placed in a  $\overbrace{2 \times 2 \times 1}^{\text{Position}} \times \overbrace{2 \times 2 \times 1}^{\text{Orientation}}$  grid in the 3D space where  $f_z = 1$  represents that we set the cameras in the same height above the surveillance plane in the example and  $f_\psi = 1$  represents that we neglect the influence of the rotation of the cameras across the axis  $Z$  in the example. We can simplify the particle state space to a  $\overbrace{2 \times 2}^{\text{Position}} \times \overbrace{2 \times 2}^{\text{Orientation}}$  space.

Figure 7 shows a mapping between one camera network placement instance to a particle’s state in the PSO state space where each tuple represent the corresponding position and pose of a deployed camera on the 3D grid. For example, the camera state (0,0,0,1) can be transferred to the position  $0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 1$  of 1 in the particle space, and the other camera state can also be transformed to the position of 1 in the particle state.

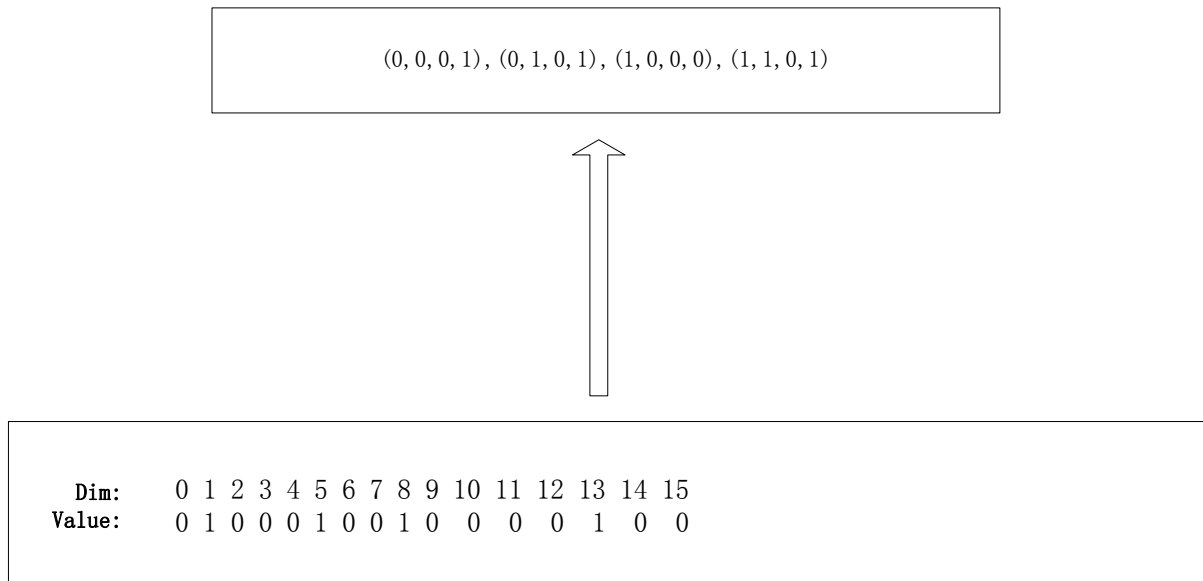
**Figure 7.** A particle state representation for a camera network placement example where we can see that the number of 1 in the particle state is 4.



4.2. Fitness Assignment

The fitness of each particle is calculated according to the cost function  $G(C)$  of the camera network placement problem. At first we should transfer the particle’s state to a camera network placement solution which is a inverse problem of the particle representation which is shown in Figure 8.

**Figure 8.** A camera network placement strategy for a particle state which is a inverse process of the particle representation.



Then we can get the fitness of the camera network configuration.

#### 4.3. Flowchart of the Proposed PI-BPSO Algorithms

The flows of most PSO algorithms are very similar, that is a four step iterations of “initialization, evaluate, update, stop”. The main difference between them is the strategy of new particles generation and the update strategy. In this work, we propose a probability induced binary particle swarm optimization algorithm (PI-BPSO) which is a extension of the algorithm propose in [13]. In PI-BPSO, the bit value of each particle’s state is determined by the “or” value of its current value (current state  $x_{ij}$ ) and the probability of being “1” (velocity  $v_{ij}$ ) which is updated according to the information sharing mechanism of PSO. The main steps of PI-BPSO can be described as follows (see Figure 9).

**Step 1: Initialization.** Setting the size of the population of the PSO and create the population of particles which is encoded as the position and orientation of the cameras in the network.

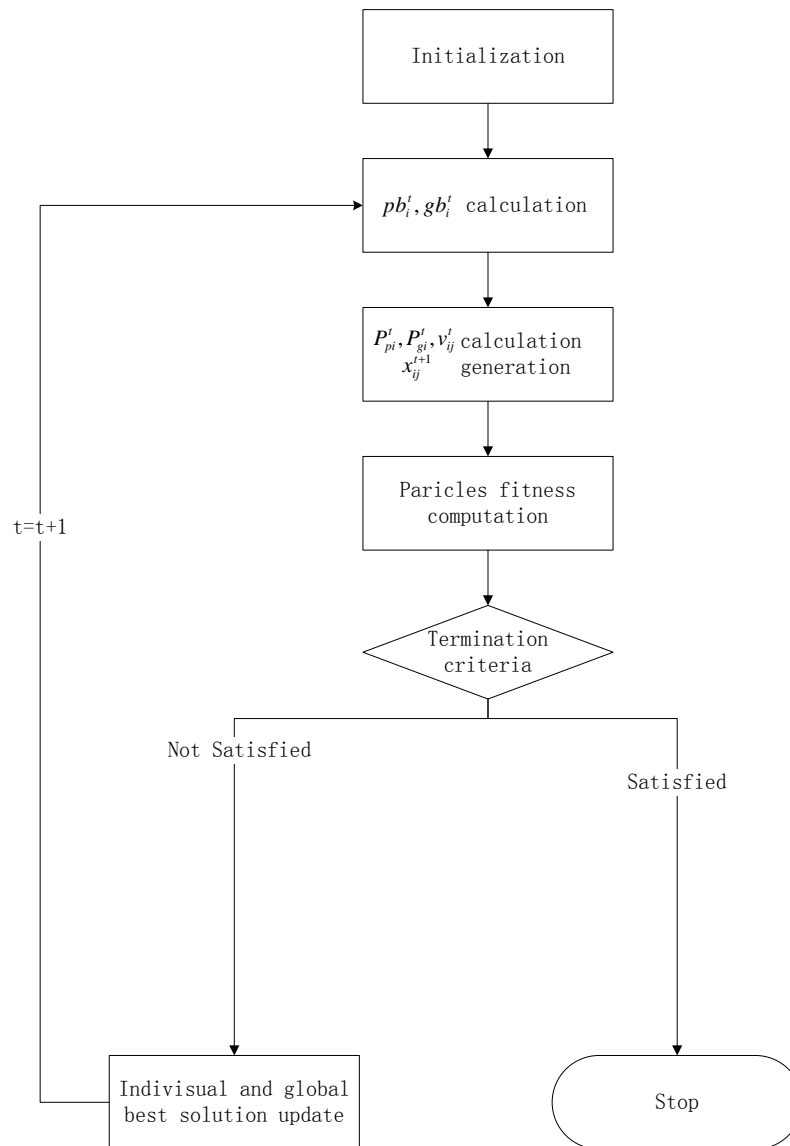
**Step 2: Calculation.** Calculate each particle  $i$ ’s best state  $pb_i^t$  and the global best state  $gb^t$  of particles until time  $t$ :

$$pb_i^t = \arg \min_t G(C_i) \quad (19)$$

$$gb^t = \arg \max_i pb_i^t \quad (20)$$

**Step 3: New particle generation.** Generate the new position of the particle  $i$  according to the probability  $P_{pi}^t$ ,  $P_{gi}^t$  step by step.

**Figure 9.** PI-BPSO flowchart. We should point out that we should calculate both the personal best and global best of the particles  $P_{pi}^t$ ,  $P_{gi}^t$  to compute the velocity of the particle.



First we calculate the probability  $P_{pi}^t$ ,  $P_{gi}^t$  as follows:

$$P_{pi}^t = \frac{1}{1 + (G(C_{pi}^t) - G(C_i^t))} \quad (21)$$

$$P_{gi}^t = \frac{1}{1 + (G(C_g^t) - G(C_i^t))} \quad (22)$$

where  $G(C_{pi}^t)$  is the utility when the particle  $i$  in its best position until time  $t$ ,  $G(C_i^t)$  is the utility when the particle  $i$  in time  $t$ ,  $G(C_g^t)$  is the maximum utility of all the particles until time  $t$ .

Secondly, we generate two uniformly distributed random numbers  $r_{ijp}^t, r_{ijg}^t$  and compare them with the probability  $P_{pi}^t$ ,  $P_{gi}^t$  to determine the personal velocity and society velocity as follows:

$$v_{ij}^{pt} = \begin{cases} 0 \dots r_{ijp}^t < P_{gi}^t \\ 1 \dots otherwise \end{cases} \quad (23)$$

$$v_{ij}^{gt} = \begin{cases} 0 \dots r_{ijg}^t < P_{gi}^t \\ 1 \dots otherwise \end{cases} \quad (24)$$

Thirdly, we determine the velocity at time  $t$  :

$$a = \varphi_1 v_{ij}^{pt} + \varphi_2 v_{ij}^{gt} \text{ where } \varphi_1 + \varphi_2 = 1 \quad (25)$$

$$v_{ij}^t = \begin{cases} 0 \dots r_{ijv}^t \leq a \\ 1 \dots otherwise \end{cases} \quad (26)$$

Fourthly, we determine the new state of the particle  $i$  :  $x_{ij}^{t+1} = x_{ij}^t \oplus v_{ij}^t$ , where  $\oplus$  is a logical operator we define as follows: If  $v_{ij}^t = 1$ ,  $x_{ij}^{t+1} = x_{ij}^t$  else  $x_{ij}^{t+1} = \text{not}(x_{ij}^t)$  which is different from [13] where the operator is “OR”. In the work of [13], when one position is assigned to be “TRUE”, then it will almost always be assigned a camera there, we think this is unreasonable in some sense. We should explain one case clearly. If  $x_{ij} = 1$  for all  $j$ , the fitness of the configuration will not be the best particle candidate (because the regulation item in the utility function) and the velocity will have great probability to be zero. Considering the update strategy in our algorithm, some position of the particle will be set to “0” with great probability. So the optimization process will not halt.

**Step 4: Particle fitness computation.** Compute the fitness of the particles according to its new positions  $x_i^t$ .

**Step 5: Decide whether to update or stop according to the terminate criteria.** Determine whether the terminate criteria is satisfied (the upper limit of the iteration times or the accuracy precision). If the terminate criteria are satisfied, then output the global best particle, else update the new personal best solution of each particle and the global best particle and goto the next iteration.

In the algorithm aspect, the main difference between our algorithm and [13] is that we distinguish the person influence and the society influence which are very important for the success of the PSO algorithms while [13] only gives a update strategy which ignores the two influence factors.

There is another key difference between the two methods: PI-BPSO uses a regulation item in the cost function to determine the appropriate number of cameras while BPSO-PI [13] doesn't consider the issue in the article. The benefit of involving the number of cameras in the cost function is evident that is when the benefit of increasing of the number of cameras can't cover the cost of the increase of the number of the cameras, we should stop increasing the number of cameras. We think that this intuitive idea will help reach get the camera network placement solution more easily.

## 5. Simulation Results

Most of the constrained optimization problems solved by PSO [13,36] apply some kind of feasibility preserving strategy to handle the constraints, which means that we only preserve the particles that satisfy all the constraints mentioned above. Through the strategy we transfer the constrained optimization problem to an unconstrained ones.



For the problems in this article, we apply the following strategy:

- (1) During the initialization stage, all particles are started with feasible solutions;
- (2) During the updating stage, only the feasible particles are kept in their memories.

To evaluate the effectiveness and reliability of the proposed algorithm, we adopted the following experiment configuration:

- (1) The surveillance area is a 50 m  $\times$  50 m square;
- (2) The camera intrinsic parameter is  $K = \begin{bmatrix} 4.8 & 0 & 0 \\ 0 & 4.8 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ ;
- (3) The minimum resolution requirement is  $\alpha = 1$  pixel/cm (for face recognition, a face is imaged as a 20  $\times$  20 square);
- (4) The CCD size is 1/4" (3.2 mm  $\times$  2.4 mm) and the resolution of the camera is 1,024  $\times$  768;
- (5) The Euler angle of the camera is restricted in the following range:  $\frac{2\pi}{3} \leq \theta \leq \frac{3\pi}{4}$ .

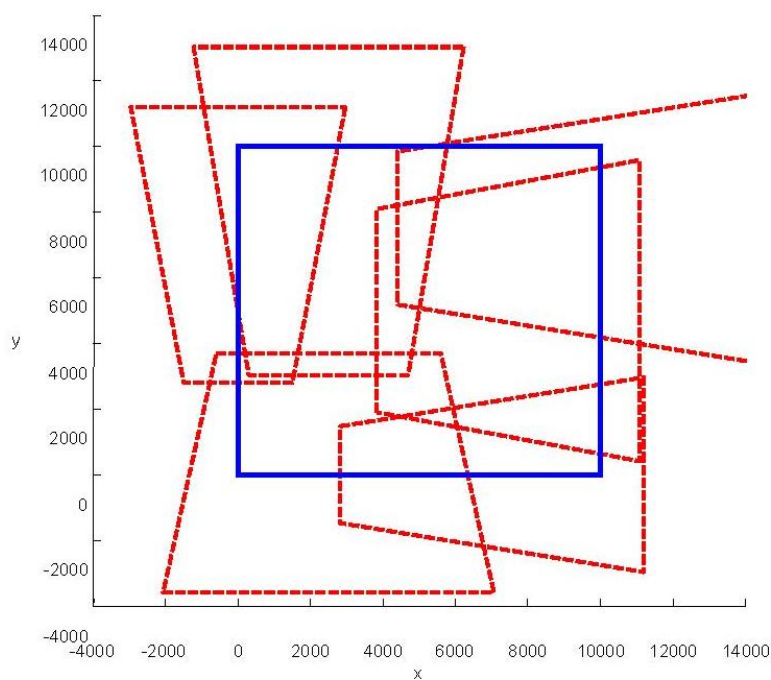
We do two kinds of experiments. One kind of experiment is to determine the influence of the different sampling frequencies on the solution of the camera network deployment problem; the second kind of the experiment is to determine the influence of the  $\lambda$  parameters on the cost functions. We set the population size at 20 and the maximum number of iterations in the experiments is 10,000.

We present some results obtained by the algorithms (Figures 10–12). In the figures, blue bold line represents the area that needs to be surveilled and the red dash lines represent the FOV of the cameras. In Figures 11 and 12, we choose the same surveillance area and choose different sample frequencies for the pose parameters. In Figure 11, the parameters is ( $f_x = f_y = f_\phi = f_\psi = 4$ ,  $f_z = f_\theta = 2$ ), the fitness of the surveillance is 0.9785, the number of the surveillance cameras is 24; while in Figure 12, the parameters is ( $f_x = f_y = f_z = f_\phi = f_\psi = f_\theta = 4$ ), the fitness of the surveillance grows to 0.9908 while the number of the surveillance cameras is decreased to 16. From the experiments, we know that we can get more accurate results when we use a bigger sample space. In Figure 11, the sample space is 4,096 which is too big for classical Binary Integer Linear programming optimizations.

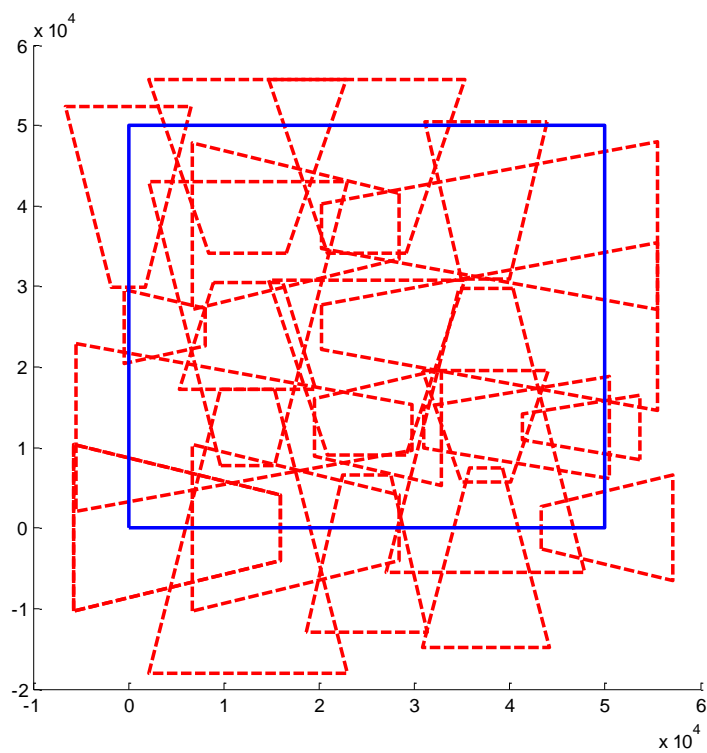
In the utility function, we set a regularization item which is a penalty for the camera number. We do an experiment to determine the effect of the  $\lambda$  parameter. In the next experiment we choose the task parameter as follows: surveillance area is 50 m  $\times$  50 m and the sample frequency is  $f_x = f_y = f_z = f_\phi = f_\psi = f_\theta = 4$ .

In Figure 13, the regularization parameter  $\lambda = 2$ . In Figure 14, the regularization parameter  $\lambda = 0.1$ . In Figure 15, the regularization parameter  $\lambda = 0.01$ . We can see that the regularization parameter takes effects to control the number of the cameras.

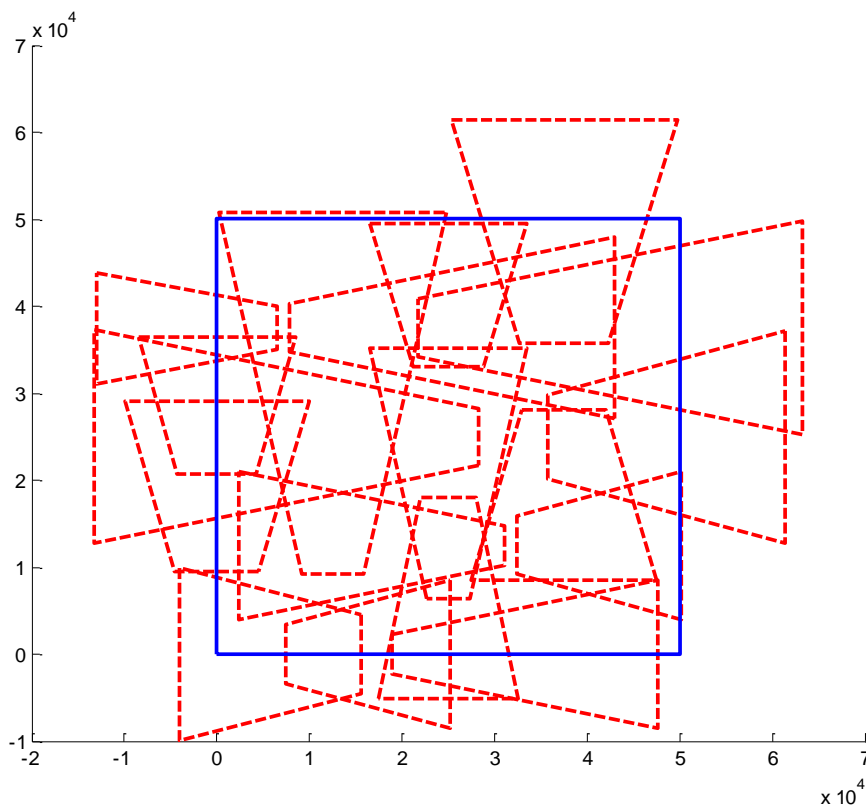
**Figure 10.** Optimal placement of cameras (six cameras). The surveillance area is  $10\text{ m} \times 10\text{ m}$ ,  $f_x = 4$ ,  $f_y = 4$ ,  $f_z = 2$ ,  $f_\varphi = 4$ ,  $f_\theta = 2$ ,  $f_\psi = 4$ ,  $\lambda = 1$ .



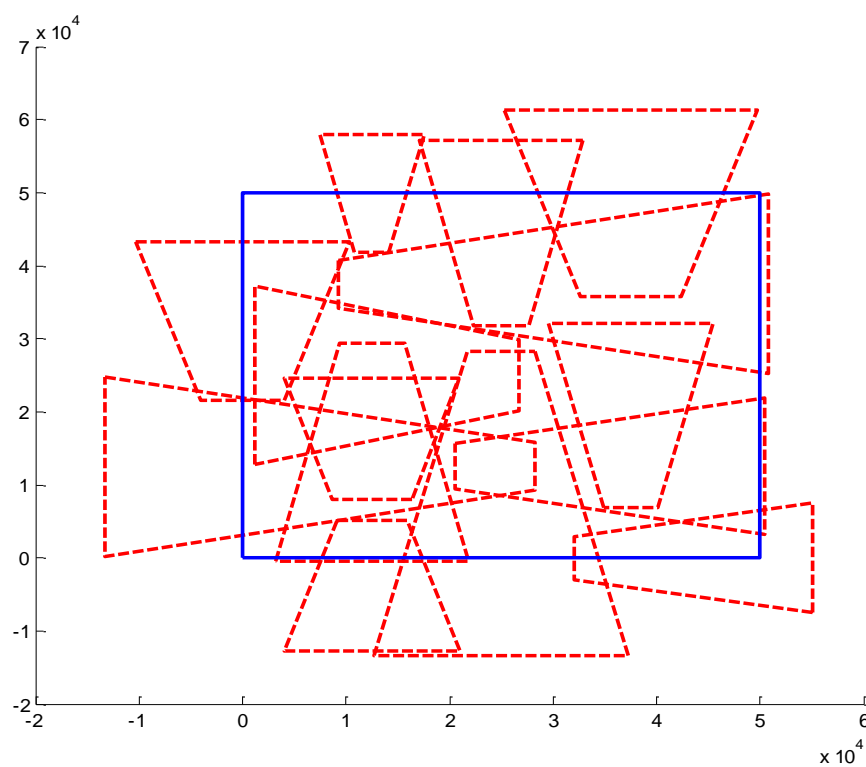
**Figure 11.** Optimal placement of cameras (24 cameras). The surveillance area is  $50\text{ m} \times 50\text{ m}$ ,  $f_x = 4$ ,  $f_y = 4$ ,  $f_z = 2$ ,  $f_\varphi = 4$ ,  $f_\theta = 2$ ,  $f_\psi = 4$ ,  $\lambda = 1$ .



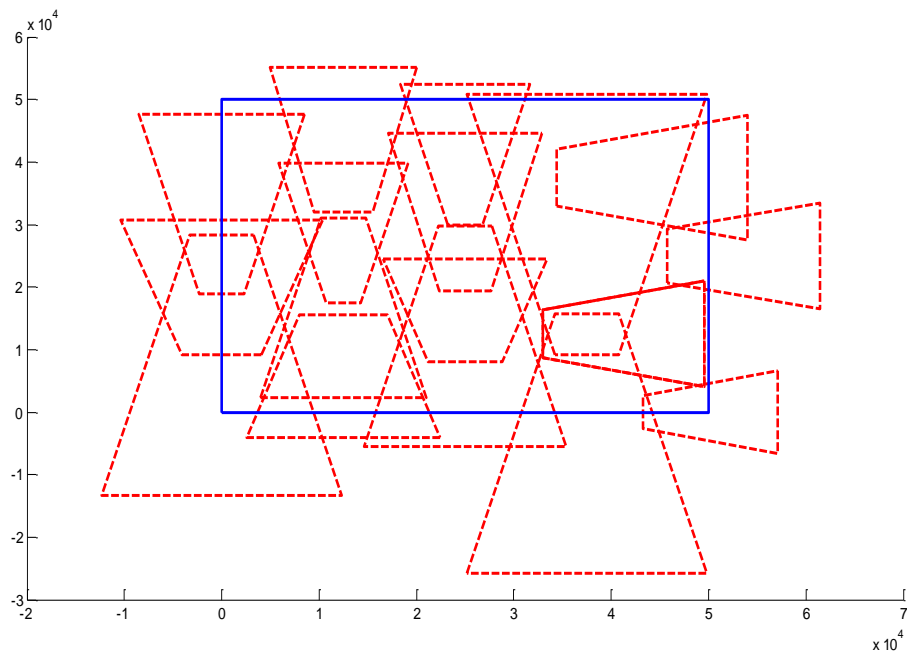
**Figure 12.** Optimal placement of cameras (16 cameras). The surveillance area is  $50\text{ m} \times 50\text{ m}$ ,  $f_x = 4, f_y = 4, f_z = 4, f_\varphi = 4, f_\theta = 4, f_\psi = 4, \lambda = 1$ .



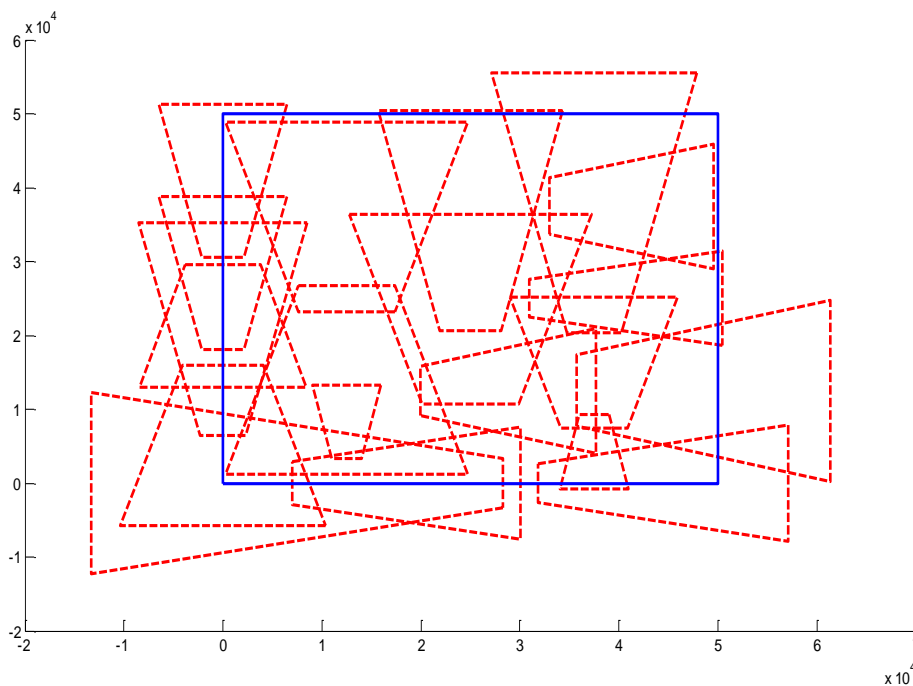
**Figure 13.** Optimal placement of cameras (14 cameras). The surveillance area is  $50\text{ m} \times 50\text{ m}$ ,  $f_x = 4, f_y = 4, f_z = 4, f_\varphi = 4, f_\theta = 4, f_\psi = 4, \lambda = 2$ .



**Figure 14.** Optimal placement of cameras (18 cameras). The surveillance area is  $50 \text{ m} \times 50 \text{ m}$ ,  $f_x = 4, f_y = 4, f_z = 4, f_\varphi = 4, f_\theta = 4, f_\psi = 4, \lambda = 0.1$ .

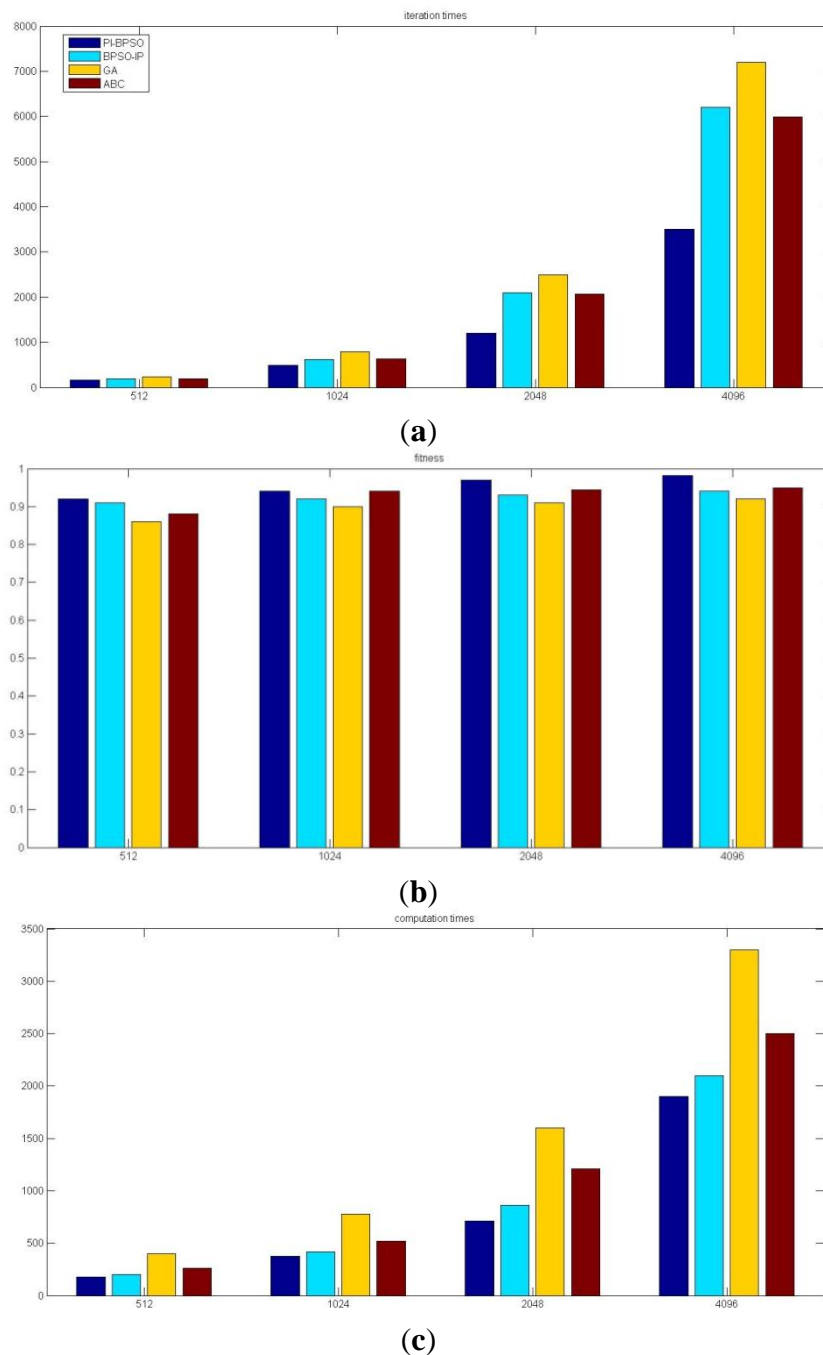


**Figure 15.** Optimal placement of cameras (20 cameras). The surveillance area is  $50 \text{ m} \times 50 \text{ m}$ ,  $f_x = 4, f_y = 4, f_z = 4, f_\varphi = 4, f_\theta = 4, f_\psi = 4, \lambda = 0.01$ .



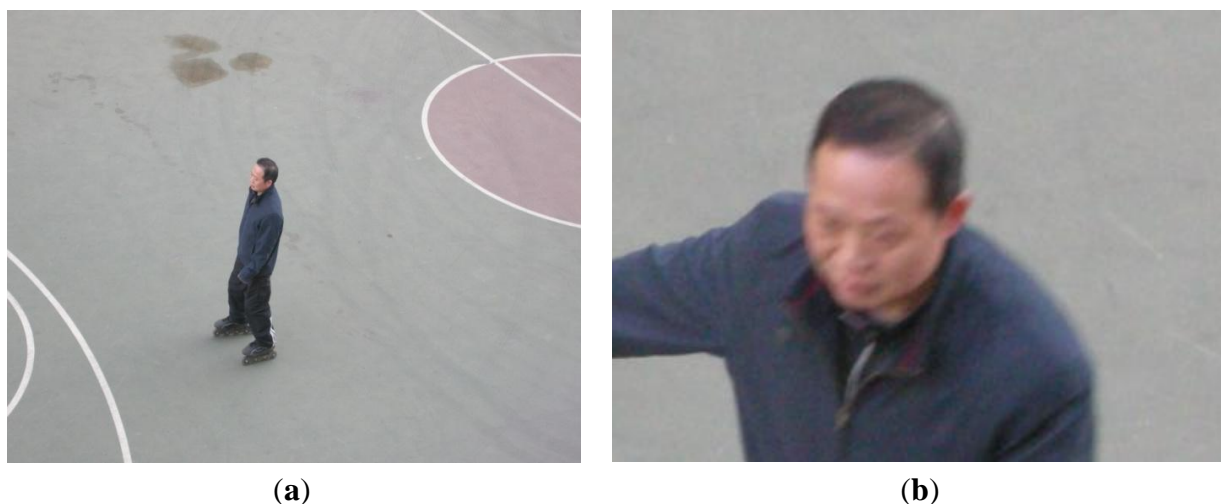
We do an experiment to compare PI-BPSO with BPSO-IP, GA and ABC from three perspectives: iteration times, fitness and computation time. We repeat the experiment 50 times and the value of the iteration times, fitness and computation time are the average values of the 50 tests. Because the codes of the other methods are all unknown, the experimental results may not be same as the original author stated. We show the result in Figure 16.

**Figure 16.** The comparison result of different optimization method. (a) The iteration times comparison result; (b) The fitness comparison result; (c) The computation time comparison result.



We do an experiment with real cameras in a hall building with the simulation result that we have obtained from the simulation. The surveillance area is a basketball court and a man is roller skating. We determine the position and pose of the surveillance cameras using the PI-BPSO algorithm. The experimental result shows the effect of the algorithm is that the video is in focus and the people in the video can be identified easily. We show the result in Figure 17.

**Figure 17.** Some images from the real surveillance video to validate the effectiveness of the algorithm. (a) The video is in focus; (b) The resolution constraints are satisfied and the person can be identified in the video.



## 6. Conclusions

In this paper, we discuss the automatic camera network placement problem which is solved by an evolution-like method, PI-BPSO. The different simulation results show the effectiveness of the proposed algorithm. The algorithm is guaranteed to get a global optimum with high probability due to two reasons:

- (1) The initialization and the update process are both determined randomly which assures that we will now land in the local minimum with high probability;
- (2) The introduction of the regulation item eases the optimization process and allows us to optimize the number of the cameras and the configuration of the cameras at the same time.

In the future, we will continue our study on the camera network placement problem from several directions, such as a more accurate initialization or integration of the PSO and other optimization methods to speed up the convergence.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants 61225008 and 61020106004, and by the Ministry of Education of China under Grant 20120002110033. This work is also supported by Tsinghua University Initiative Scientific Research Program.

## Conflicts of Interest

The authors declare no conflict of interest.

## References

1. Wang, X. Intelligent multi-camera video surveillance: A review. *Pattern Recognit. Lett.* **2013**, *34*, 3–19.

2. Chen, W.; Aghajan, H. User-centric environment discovery with camera networks in smart homes. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2011**, *41*, 375–383.
3. Gruenwedel, S.; Xie, X.; Philips, W.; Chen, C.-W.; Aghajan, H. A Best View Selection in Meetings through Attention Analysis Using a Multi-Camera Network. In Proceedings of the 2012 Sixth International Conference on Distributed Smart Cameras (ICDSC), Hong Kong, China, 30 October–2 November 2012.
4. Charfi, Y.; Wakamiya, N.; Murata, M. Challenging issues in visual sensor networks. *IEEE Trans. Wirel. Commun.* **2009**, *16*, 44–49.
5. Chen, A.M.X. Modeling coverage in camera networks: A survey. *Int. J. Comput. Vis.* **2012**, *101*, 205–226.
6. Wang, B. Coverage problems in sensor networks: A survey. *ACM Comput. Surv.* **2011**, *43*, 1–53.
7. Chien-Fu, C.; Kuo-Tang, T. Distributed barrier coverage in wireless visual sensor networks with B-QoM. *IEEE Sens. J.* **2012**, *12*, 1726–1735.
8. Yildiz, E.; Akkaya, K.; Sisikoglu, E.; Sir, M. Optimal camera placement for providing angular coverage in wireless video sensor networks. *IEEE Trans. Comput.* **2013**, *99*, 1, doi:10.1109/TC.2013.45.
9. Munishwar, V.P.; Abu-Ghazaleh, N.B. Scalable Target Coverage in Smart Camera Networks. In Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras, Atlanta, GA, USA, 31 August–4 September 2010; pp. 206–213.
10. Cole, R.; Sharir, M. Visibility problems for polyhedral terrain. *Symb. Comput.* **1989**, *7*, 11–30.
11. Zhao, J.; Haws, D.; Yoshida, R.; Cheung, S.S. Approximate Techniques in Solving Optimal Camera Placement Problems. In Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCVW), Barcelona, Spain, 6–13 November 2011.
12. Kulkarni, R.; Venayagamoorthy, G.K. Particle swarm optimization in wireless sensor networks: A brief survey. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2011**, *41*, 262–267.
13. Morsly, Y.; Aouf, N.; Djouadi, M.S.; Richardson, M. Particle swarm optimization inspired probability algorithm for optimal camera network placement. *IEEE Sens. J.* **2012**, *12*, 1402–1412.
14. Chrysostomou, D.; Gasteratos, A. Optimum Multi-Camera Arrangement Using a Bee Colony Algorithm. In Proceedings of the 2012 IEEE International Conference on Imaging Systems and Techniques (IST), Manchester, UK, 16–17 July 2012.
15. Lee, J.-Y.; Seok, J.-H.; Lee, J.-J. Multiobjective optimization approach for sensor arrangement in a complex indoor environment. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2012**, *42*, 174–186.
16. Urrutia, J. Art Gallery and Illumination Problems. In *Handbook of Computational Geometry*; Sack, J.R., Urrutia, J., Eds.; Elsevier Science: Amsterdam, Holland, 2000; pp. 973–1027.
17. Culberson, J.C.; Reckhow, R.A. Covering polygons is hard. *J. Algorithms* **1994**, *17*, 2–44.
18. Orourke, J.; Supowit, K.J. Some NP-hard polygon decomposition problems. *IEEE Trans. Inf. Theory* **1983**, *29*, 181–190.
19. Zhao, J.; Cheung, S. Optimal Visual Sensor Planning. In Proceedings of the IEEE International Symposium on Circuits and Systems, Taipei, Taiwan, 24–27 May 2009.
20. Soro, S.; Heinzelman, W. A survey of visual sensor networks. *Adv. Multimed.* **2009**, doi: 10.1155/2009/640386.

21. Indu, S.; Chaudhury, S.; Mittal, N.R.; Bhattacharyya, A. Optimal Sensor Placement for Surveillance of Large Spaces. In Proceedings of the Third ACM/IEEE International Conference on Distributed Smart Cameras, Como, Italy, 30 August–2 September 2009.
22. Puccinelli, D.; Haenggi, M. Wireless sensor networks: Applications and challenges of ubiquitous sensing. *IEEE Circuits Syst. Mag.* **2005**, *5*, 19–31.
23. Wang, C.; Qi, F.; Shi, G. Observation Quality Guaranteed Layout of Camera Networks via Sparse Representation. In Proceedings of the 2011 IEEE, Visual Communications and Image Processing (VCIP), Tainan, Taiwan, 6–9 November 2011.
24. Cowan, C.K.; Kovese, P.D. Automatic sensor placement from vision task requirements. *IEEE Trans. Pattern Anal. Mach. Intell.* **1988**, *10*, 407–416.
25. Erdem, U.M.; Sclaroff, S. Optimal Placement of Cameras in Floorplans to Satisfy Task Requirements and Cost Constraints. In Proceedings of the OMNIVIS Workshop, Prague, Czech Republic, 11–14 May 2004.
26. Yao, Y.; Chen, C.-H.; Abidi, B.; Page, D.; Koschan, A.; Abidi, M. Can You See Me Now? Sensor Positioning for Automated and Persistent Surveillance. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2010**, *40*, 101–115.
27. Newell, A.; Akkaya, K.; Yildiz, E. Providing Multi-Perspective Event Coverage in Wireless Multimedia Sensor Networks. In Proceedings of the 2010 IEEE 35th Conference on Local Computer Networks Conference (LCN), Denver, CO, USA, 10–14 October 2010.
28. Wolfram Mathworld. Available online: <http://mathworld.wolfram.com/EulerAngles.html> (accessed on 22 November 2013).
29. Conrad, J. Depth of Field in Depth. Available online: <http://www.largeformatphotography.info> (accessed on 22 November 2013).
30. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197.
31. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359.
32. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471.
33. Karaboga, D.; Basturk, B. On the performance of artificial bee colony (ABC) algorithm. *Appl. Soft Comput.* **2008**, *8*, 687–697.
34. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995.
35. Tzevanidis, K.; Argyros, A. Unsupervised learning of background modeling parameters in multicamera systems. *Comput. Vis. Image Underst.* **2011**, *115*, 105–116.
36. Hu, X.; Eberhart, R. Solving Constrained Nonlinear Optimization Problems with Particle Swarm Optimization. In Proceedings of the Sixth World Multi-conference on Systemics, Cybernetics and Informatics, Orlando, FL, USA, 14–18 July 2002.