On the Statistical Determination of Optimal Camera Configurations in Large Scale Surveillance Networks

Junbin Liu[1], Clinton Fookes[1], Tim Wark[2], and Sridha Sridharan[1]

[1] Image & Video Research Laboratory, Queensland University of Technology,
2 George Street, Brisbane, QLD 4000, Australia
junbin.liu@gmail.com, {c.fookes,s.sridhara}@qut.edu.au
[2] CSIRO ICT Centre,
1 Technology Court, Pullenvale, QLD 4069, Australia
tim.wark@csiro.au

Abstract. The selection of optimal camera configurations (camera locations, orientations etc.) for multi-camera networks remains an unsolved problem. Previous approaches largely focus on proposing various objective functions to achieve different tasks. Most of them, however, do not generalize well to large scale networks. To tackle this, we introduce a statistical formulation of the optimal selection of camera configurations as well as propose a Trans-Dimensional Simulated Annealing (TDSA) algorithm to effectively solve the problem. We compare our approach with a state-of-the-art method based on Binary Integer Programming (BIP) and show that our approach offers similar performance on small scale problems. However, we also demonstrate the capability of our approach in dealing with large scale problems and show that our approach produces better results than 2 alternative heuristics designed to deal with the scalability issue of BIP.

Keywords: Camera placement, optimization, resersible jump Markov chain Monte Carlo, simulated annealing.

1 Introduction

Networks of cameras have been widely used in the area of intelligent video surveillance (IVS). Based on user defined policies, IVS systems can automatically identify potential risks by detecting, localizing, tracking and recognizing targets and/or events of interest [1]. For these fixed camera networks, it is of vital importance that the optimal camera configuration (i.e. optimal location, orientation etc.) is determined before cameras are installed, as the cost of rewiring can be expensive and the optimal configuration may provide saving on the total number of cameras used to achieve the same level of utility.

The camera placement problem is far from trivial. Even in its simplest setting where an optimal configuration is sought to achieve a pre-defined coverage while minimizing the number of cameras needed, the problem may be infeasible if simple enumeration and search techniques are employed (e.g. [2, 3]) due to its NP-hard nature. Thus these methods can hardly be extended to large scale problems.

To address this issue, we first introduce a generalized statistical framework for the problem of selecting the optimal camera configuration, considering a number of user constraints and unknown number of cameras. Second, we propose a Trans-Dimensional Simulated Annealing (TDSA) algorithm based on Monte Carlo sampling to effectively estimate the optimal number of cameras as well the optimal parameters for these cameras. We show that our method offers very similar solutions to the optimal ones produced by Binary Integer Programming (BIP) in small scale problems. For larger scale problems where BIP is clearly infeasible, we demonstrate our approach offers notable improvements over two alternative heuristics proposed recently [3, 4]. It is worth noting that due to our problem formulation, sometimes user requirements are partially satisfied to 99.9%, but this limitation should not affect the use of the algorithm in practice.

2 Related Work

Current research into automatic camera placement techniques generally tackles two types of problems, depending on the objects to be monitored. The first type ([5–7] etc.) focuses on computing the camera placement for a set of cameras that monitor the same object. The goal is usually to reconstruct the object using multi-view inputs. The second type ([2, 8, 4, 9] etc.) focuses on the best strategy to spread the views of cameras so that an area is monitored. The approach described in this paper belongs to the latter category.

The automatic camera placement problem has been studied by various authors in a number of contexts with different constraints and requirements. The earliest related work was published by O'Rourke [10] who provided the formulation of the Art Gallery Problem and its solutions. The Art Gallery Problem is the assignment of guards to different positions in an art gallery in order to achieve the maximum visual coverage of the paintings. In essence, this is equivalent to finding the best locations for a set of infinite-depth omni-directional cameras to achieve the optimal coverage of the observation area.

Recently, Erdem and Sclaroff [2] formulated the general camera placement problem in an optimization framework. Given the set of all constraints τ required to achieve a specific task γ, the problem is to find the optimum placement for a set of cameras Π in the area V satisfying τ and minimizing a given cost function $G(\bullet)$. The authors proposed four instances of this general formulation and solved them using Binary Integer Programming (BIP) over a discrete problem space. This formulation was later adopted by Horster and Lienhart [3] but instead of restricting the areas to be polygon shaped, they utilise points to represent space.

Various user-requirements have been considered. Yao et al. [8] proposed to incorporate a hand-off success rate (the percentage of successful hand-offs) analysis in determining camera placement, preserving necessary uniform overlapped field of views (FoVs) between adjacent camera for an optimal balance between coverage and hand-off success rate. Bodor et al. [9] proposed an approach to the camera placement problem that tries to optimize the camera network's ability to observe a set of predefined tasks, such as human motion. The authors developed an analytical formulation of the observation problem, in terms of the statistics of the motion in the scene and the total resolution of the observed actions. The observability of frontal faces has been incorporated in the process of finding the optimal camera placement by Ram et al. [11]. Zhao and Cheung [4] described a visibility model that takes in a number of realistic inputs: arbitrary-shape 3D environments, 3D camera models, occupant traffic models, self-occlusion and mutual occlusion. The visibility model is used in evaluating the objective - maximization of the probability of tracking visual tags. Similarly, Mittal and Davis [12] considered occlusion in a probabilistic manner in the visibility calculation and used simulated annealing to compute the optimum camera parameters. However, it is not clear how the optimal number of cameras is determined, which we will address in this paper.

Many of the existing methods formulated the problem as a combinatorial optimization [2–4] that belongs to a class of NP-hard problems. Heuristics methods have been developed with computational requirements proportional to small powers of N, i.e. in a relatively simple environment. However heuristics can be problem-specific as there is no guarantee that a heuristic will work for all instances. Alternatively, a divide and conquer approach can break-down the problem into a number of simpler ones. The problem with this approach is that the result is only optimal when the sub problems are disjoint, which is often not the case. To counter these problems, we formulate the camera placement as a maximum a-posteriori model selection and optimization in Section 4 and propose a viable solution using trans-dimensional simulated annealing in Section 5.

3 Problem Definition

In this work we are not restricting ourselves to a particular camera model. Each camera $c_i = \left[p_1, p_2, ..., p_{n_p}\right]^\top$ consists of n_p number of parameters which may include the camera location in x, y, z directions of the area to be monitored, orientation, tilting angle, lens type etc. The camera parameter space is denoted as $\mathcal{C} = \mathcal{R}^{n_p}$. Note that although c_i is a vector, we choose not to use boldface to avoid confusion in subsequent sections of the paper.

Although there is no particular restrictions on what parameter to include in the camera model, it is however, essential that given a particular network of cameras $\boldsymbol{\theta} = \left[c_1, c_2, ..., c_{n_c}\right]^\top$, a set of constraints $\mathbf{r} = \{r_j \mid r_j \in \{r_1, r_2, ..., r_{n_r}\}\}$ and a description of the area to be monitored ξ, there must exist some function $L\left(\mathbf{r} \mid \boldsymbol{\theta}\right)$ that determines how well the constraints $\mathbf{r}$ are jointly satisfied by $\boldsymbol{\theta}$. For example, if it is required to achieve a frontal face capture rate of 80%, but a particular camera set only achieves 60%, then it may be said the cameras have achieved $60/80 = 75\%$ of the requirement.

Given the camera parameter space $\mathcal{C}$, an input environment ξ, a set of user requirements $\mathbf{r}$, the camera placement problem can be defined as the selection of the camera configuration that meets $\mathbf{r}$ while minimizing the number of cameras used. We will show in Section 4 that this formulation can be easily applied to the problem where some utility is to be maximized while keeping a constant number of cameras.

4 Generalized Framework

To formulate the problem in a Bayesian modelling context, suppose that there is a countable collection of candidate models $\mathcal{M}_k$ indexed by a model indicator $k \in \mathcal{K}$. Each model $\mathcal{M}_k$ has an $\|\mathcal{M}_k\| = n_k$ dimensional vector of parameters $\boldsymbol{\theta}_k$. Note that for the sake of simplicity we use the model indicator to represent the model. For example, the dimension of the model is $\|k\| = n_k$. In the context of camera placement, the model of a camera network configuration is reflected by the number of cameras in the configuration and *each camera is considered as a random variable* over the space $\mathcal{C}$.

The camera placement problem is defined by the joint posterior,

$$\phi_{opt} = \underset{\phi_k \in \mathcal{X}}{\arg\max} \left\{ p\left(k, \boldsymbol{\theta}_k \mid \mathbf{r}\right) \right\}, \tag{1}$$

where $\phi = (k, \boldsymbol{\theta}_k)$ denotes a camera configuration which contains a model indicator k as well as camera parameters of the model $\boldsymbol{\theta}_k$. The joint state space is thus $\mathcal{X} = \bigcup_{k \in \mathcal{K}} \left(\{k\} \times \mathcal{C}^{n_k}\right)$. This formulation can be considered as: given there is an observation that the list of constraints and requirements have been satisfied, the problem is to find the optimal model k and the optimal parameters $\boldsymbol{\theta}_k$ that are most likely to have led to this observation. For example, if the requirement is covering the maximum amount of the floor with a given number of cameras. Eqn. (1) can then be interpreted as finding the most likely camera configuration that has caused maximum coverage to be observed (satisfied).

Expanding Eqn. (1) using Bayes theorem yields,

$$\phi_{opt} = \underset{\phi_k \in \mathcal{X}}{\arg\max} \left\{ L\left(\mathbf{r} \mid k, \boldsymbol{\theta}_k\right) p\left(\boldsymbol{\theta}_k \mid k\right) p\left(k\right) \right\}. \tag{2}$$

The first term $L\left(\mathbf{r} \mid k, \boldsymbol{\theta}_k\right)$ is the probability of satisfying the requirements and constraints $\mathbf{r}$ by the given set of camera parameters $\boldsymbol{\theta}_k$ and is therefore called the likelihood. The second term $p\left(\boldsymbol{\theta}_k \mid k\right)$ is termed the parameter prior since it defines the prior probability of the set of camera parameters. The prior term allows the user to set preferences on the parameters of the cameras. For example, wall locations can be preferred through assigning a high prior to cameras that are located on walls and omni-directional cameras may be unwanted by associating them with a low prior. The last term $p\left(k\right)$ is the model prior which captures user preference on the models. In the most settings where all cameras are treated equally, this term can be dropped. This is the case for all the experiments presented in Section 6.

The formulation (Eqn. (2)) can be converted to a penalized model selection problem for further investigation. An equivalent form of Eqn. (2) can be obtained as,

$$\phi_{opt} = \underset{k, \boldsymbol{\theta}_k}{\arg\min} \left\{ -\log\{L\left(\mathbf{r} \mid k, \boldsymbol{\theta}_k\right) p\left(\boldsymbol{\theta}_k \mid k\right)\} + \log\{1/p\left(k\right)\} \right\}. \tag{3}$$

If letting $p(k) = \exp\{-n_k\}$, we obtain the well known AIC [13] that is often used in model selection problems,

$$\phi_{opt} = \underset{k, \boldsymbol{\theta}_k}{\arg\min} \left\{ -\log\{L\left(\mathbf{r} \mid k, \boldsymbol{\theta}_k\right) p\left(\boldsymbol{\theta}_k \mid k\right)\} + n_k \right\}. \tag{4}$$

The above formulation is designed for the problem when the constraints are to be met to a particular level while minimizing the number of cameras required. If however, the problem is to maximize a utility with a fixed number of cameras, the model indicator can simply be dropped out of the formulation to keep a constant model dimension.

In some cases one may be interested to minimize the total cost of a camera network when there are more than one type of cameras with different prices available. In these cases, the model prior can be assumed uniform and the penalty term (n_k in Eqn. (4)) can be replaced with a function on the total price of the particular configuration.

5 Trans-Dimensional Simulated Annealing

We propose to use the Trans-Dimensional Simulated Annealing (TDSA) [14, 15] to solve the stochastic problem posed in Section 4. Simulated Annealing (SA) is a class of algorithms capable of locating good near-optima of objective functions in large search spaces. The term simulated annealing derives from the interesting observation that as a heated material slowly cools down, its molecules will line up in a rigid pattern corresponding to a state of minimum energy provided that the cooling process is sufficiently slow. SA algorithms mimic this process and have been proven to converge [16]. Trans-dimensional simulated annealing is a class of algorithms that extend the traditional simulated annealing by allowing moves that not only change the parameters of the model but alsol move between plausible models. Therefore, TDSA algorithms are able to locate the models and parameters that minimize objectives such as AIC [13], BIC [17] and MDL [18]. In this section, we will detail the design of such an algorithm which can be used effectively for large scale camera placement problems.

To start the derivation process, we first introduce a slight modification to AIC in Eqn. (4) that allows the control of the severity of the penalty placed on model order,

$$J(\phi) = -\log\{L(\mathbf{r} \mid k, \boldsymbol{\theta}_k)\, p(\boldsymbol{\theta}_k \mid k)\} + \gamma n_k. \tag{5}$$

Remark. The value of $J(\phi)$ in Eqn. (5) is determined by the sum of a log term and a penalty term γn_k. If there are two configurations who have equal log terms, then the penalty term is effective in selecting the configuration with the least number of cameras. However, the use of the penalty term may also result in the cases where the configuration that satisfies the requirement 100% are not selected because of the existence of a very close alternative configuration whose penalty term is small enough that leads to a smaller overall $J(\phi)$. If the γ used is small then the alternative configuration will be extremely close, for example 99.99% as opposed to 100% user requirement satisfaction as shown in the experiments in Section 6.

Given the objective function $J(\phi)$ that we wish to minimize over the joint space $\bigcup_{k\in\mathcal{K}} (\{k\} \times \mathcal{C}^{n_k})$, the corresponding Boltzmann distribution [19] can be defined as,

$$\begin{aligned} b_T(\phi) &\propto \exp\{-J(\phi)/T_i\} \\ &= (L(\mathbf{r} \mid k, \boldsymbol{\theta}_k)\, p(\boldsymbol{\theta}_k \mid k))^{1/T_i} \exp\{\gamma n_k/T_i\}, \end{aligned} \tag{6}$$

where T_i is a decreasing cooling schedule with $\lim_{i\to\infty} Ti = 0$. There exists many valid types of cooling schedules such as linear, logarithmic and geometric. In particular, the logarithmic schedule has been proven to always lead to convergence but at a very slow rate. For real camera placement problems, near-optimal solutions are often acceptable. Therefore, we have chosen to use the geometric schedule with initial value of T_0, as based on a few pilot runs of the experiment, this schedule leads to a balanced speed and accuracy of convergence.

$$T_i = \rho T_{i-1}, \tag{7}$$

where $0 < \rho < 1$ is the cooling coefficient.

The TDSA algorithm proposed involves simulating a non-homogeneous Markov chain whose stationary distribution at temperature T_i is proportional to $b_T(\phi)$. For each temperature, given that the Markov chain is at some current state ϕ, the algorithm first selects a move type m from a set of predefined moves, which are birth m_b, death m_d and update m_u with prior probability $p_m \in \{p_b, p_d, p_u\}$. It then generates a new candidate camera configuration ϕ' by sampling an auxiliary variable $\mathbf{u}$ from a known density $g_m(\mathbf{u})$. $\mathbf{u}$ is subsequently combined with the current state ϕ through some deterministic function h: $\phi' = h(\phi, \mathbf{u})$ to produce the proposed candidate. The move can either change the dimensionality of the state (i.e. jump to a different model) or the chain can remain at the current model (i.e. dimension remains unchanged). Last, ϕ' is accepted with probability $\alpha(\phi, \phi')$ [20], where

$$\alpha(\phi, \phi') = \min\left\{1, \frac{b_T(\phi')p'_m g'_m(u')}{b_T(\phi)p_m g_m(u)} \left|\frac{\partial(\phi', u')}{\partial(\phi, u)}\right|\right\}, \tag{8}$$

Here p'_m is the probability of choosing the reverse move of m and g'_m is the associated known proposal density of the reverse move. For the proposed algorithm, birth, death and update moves have been selected. The birth and death constitute a pair of reversible moves that allow the state of the chain to grow from k to $k+1$ and decrease from k to $k-1$. The reverse move of the update move is itself. Although other moves, typically merge and split, can be defined, the ones selected have been tested and found to produce satisfactory results. Each move is described below.

5.1 Birth and Death Moves

Birth and death moves are a pair of reversible moves that facilitate model dimension changes. The birth move we propose is rather simple: for a given state ϕ, a new camera is created randomly and added to ϕ to form ϕ'. Similarly the death move is achieved by randomly removing a camera from the existing camera configuration. The acceptance ratio of the moves are,

$$\alpha_{birth} = \min\left\{1, \frac{b_T(\phi')p_{m_d}n_{max}}{b_T(\phi)p_{m_b}(n_k+1)}\right\}, \tag{9}$$

$$\alpha_{death} = \min\left\{1, \frac{b_T(\phi')p_{m_b}n_k}{b_T(\phi)p_{m_d}n_{max}}\right\}, \tag{10}$$

where n_{max} is a user defined maximum allowable dimension of the models. The Jacobian term of Eqn. (8) for both birth and death moves can be shown to be 1.

5.2 Update Moves

The update move is an important move that allows the estimation of better camera configuration while preserving the dimension of the state. It starts by first randomly selecting an existing camera from the current state of the configuration, i.e. select c_i from $\boldsymbol{\theta}_k$ and then update this camera with random walk. For example, if a camera is described by its location on a planar map and its orientation, the random walk may be constructed as a consecutive Gaussian perturbations of the x location, the y location and the orientation, with means being their current values and some pre-defined standard deviations. Since the random walk is symmetrical and the move does not involve dimension changes, the acceptance probability is reduced to,

$$\alpha_{update} = \min\left\{1, \frac{b_T(\phi')}{b_T(\phi)}\right\}. \quad (11)$$

5.3 Summary

The steps of the proposed trans-dimensional simulated annealing algorithm are summarized in Alg. 1. It can be seen that there is exactly one evaluation of the objective function $J(\phi)$ in each iteration of the Markov chain and all the rest of the computation takes constant time. Therefore the speed of the proposed algorithm is almost proportional to the speed of evaluation of J, which is case dependent.

6 Evaluation

The generalized framework and the proposed TDSA algorithm are evaluated in a number of ways. We start by describing the experimental methodology and then compare our approach with Erdem and Sclaroff's method [2], the GREEDY algorithm by Zhao et al. [4] and the dual sampling proposed by Horster and Lienhart [3] in a number of setups. Note that Erdem and Sclaroff's methods is based on BIP which produces optimal results but only for small scale problems. The later two methods are customized heuristics designed to deal with this shortcoming of BIP but at the expenses of optimality. Last, we demonstrate the flexibility of the proposed approach in dealing with a more complex scenario where, apart from achieving 100% coverage, a number of pre-labelled critical areas are also required to be covered by at least two cameras.

6.1 Experimental Methodology

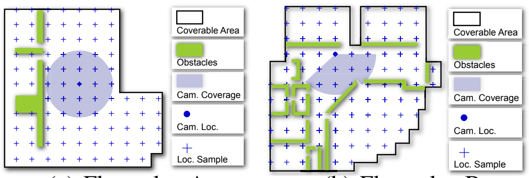
Since our focus is on establishing a generalized statistical framework for selecting the optimal camera configuration and proposing an effective algorithm to deal with the problem of scalability of BIP, we have chosen a simple setup to evaluate our algorithm. Given a 2D map of an area and the camera specifications, the goal is to compute the optimal configuration that uses the least number of cameras and achieve: (1) a desired total coverage of 100% (Section 6.2), or (2) a desired total coverage of 100% and 100% coverage of the pre-labelled critical regions by two or more cameras (Section 6.3).

Floor Plan. Two floor plans were used in the experiments which are shown in Fig. 1. The first floor plan (638×616pixel^2) is adopted from [2] and the second floor plan (804×733pixel^2) is a modification from the real floor plan of a typical university building. Both of the floor plans consist of only polygonal areas where camera coverage is possible, and holes (cavities) where camera viewing frustums are blocked completely or partially.

```
Function $\{\phi_{opt}, s_{max}\} \leftarrow$ TDSA $(T_0, T_e, p_{m_b}, p_{m_d}, \phi_0, l, \gamma, \rho)$
Input:
$T_0, T_e$ – The initial temperature and the end temperature.
$p_b, p_d, p_u$ – The probability of choosing the birth, death and update move respectively.
$\phi^0 = (k^0, \boldsymbol{\theta}_k^0)$ – The initial state of the Markov chain, where
$k^0$ – The initial model order.
$\boldsymbol{\theta}_k^0$ – The initial camera configurations.
$l, \gamma, \rho$ – The chain length, model penalty parameter and cooling coefficient.
Output:
$\phi_{opt}$ – The optimal model and the optimal camera parameters.
$s_{max}$ – The optimal value of the objective function Eqn. 5.
begin
    $T \leftarrow T_0$,$\phi \leftarrow \phi_0$,$s_{max} \leftarrow -\infty$.
    $s_c \leftarrow$ J $(\phi, \gamma)$ /*Eqn 5*/.
    while $T \geq T_e$ do
        for $j \in \{1, 2, 3, ..., l\}$ do
            $\beta \leftarrow$ RAND $(0, 1)$.
            if $\beta \leq p_b$ then
                $\phi' \leftarrow$ BIRTH $(\phi)$ /*Birth Move*/.
                $s_p \leftarrow$ J$(\phi', \gamma)$ /*Eqn 5*/.
                $\alpha \leftarrow$ ALPHA$_B(\phi', s_p)$ /*Eqn. 9*/.
            else if $\beta \leq p_b + p_d$ then
                $\phi' \leftarrow$ DEATH $(\phi)$ /*Death Move*/.
                $s_p \leftarrow$ J$(\phi', \gamma)$ /*Eqn 5*/.
                $\alpha \leftarrow$ ALPHA$_D(\phi', s_p)$ /*Eqn. 10*/.
            else
                $\phi' \leftarrow$ UPDATE $(\phi)$ /*Update Move*/.
                $s_p \leftarrow$ J$(\phi', \gamma)$ /*Eqn 5*/.
                $\alpha \leftarrow$ ALPHA$_U(\phi', s_p)$ /*Eqn. 11*/.
            end
            $\mu \leftarrow$ RAND $(0, 1)$.
            if $\mu < \alpha$ then
                $\phi \leftarrow \phi'$, $s_c \leftarrow s_p$.
                if $s_c > s_{max}$ then
                    $s_{max} \leftarrow s_c$, $\phi_{opt} \leftarrow \phi$.
                end
            end
        end
        $T \leftarrow \rho T$.
    end
end
```

Algorithm 1. trans-dimensional simulated annealing for determining the optimal camera configuration

Camera Model. Each camera is described by 4 parameters, $c = [x, y, o, t]$, which are x and y locations, orientation o and type t. The camera type parameter specifies the field of view and the depth (maximum distance visible in pixels) of the camera. For omnidirectional cameras, orientation is irrelevant and the field of view is always $360°$. The



(a) Floor plan A (b) Floor plan B

Fig. 1. Floor plans used in the experiments

coverage area of an omni-directional camera and that of a PTZ or perspective camera is depicted in Fig. 1(a) and Fig. 1(b) respectively.

Since the input environment map is naturally available in the unit of pixel, therefore most variables in our experiments were converted to pixels. In particular, these include camera locations and coverage area size. If given the real parameters of a camera,this is not difficult to achieve. For example, the OmniVision OV9655 has a pixel size of 3.18×10^{-6}m and focal length of $4.85 \times 10^{-}3$m. Thus, the focal length in pixels is $4.85 \times 10^{-3}/(3.18 \times 10^{-6}) = 1525$pixels. Assuming an OV9655 camera is mounted horizontally on a wall at human height, the distance at which the width of the image of a human head of 20×10^{-2}m will be exactly 50pixels is = $20 \times 10^{-2}/(50 \times 1525) = 6.1$m. If the floor is 26×25m^2 and the size of the image of the floor plan is 638×616pixel^2, then each meter in real world corresponds to $638/26 = 616/25 = 24.5$pixels on the image of the floor plan. 6.1m $\times$ 24.5pixel/m $= 149.45$pixels. Therefore the maximum depth of the camera has to be at most 150pixels to ensure human heads are at least 50pixels wide on the images captured by the camera.

Sampling. Ideally, the camera parameters are continuous (except the camera type). A camera can be positioned anywhere in an environment and posed at any angle. However, due to the use of optimization methods as opposed to closed form solutions, the parameters were *sampled* to reduce computation. The 2D environment maps were divided into grids to allow easier computation of cameras' coverage regions. The locations where cameras can exist were restricted to a number of location samples, which are represented as crosses in Fig. 1(a) and Fig. 1(b). Similarly, the orientations of the cameras (except omni-directional cameras) were also sampled. The sampling of parameters allows the construction of the set of *candidate cameras* (also sometimes referred to as the sampled set), from which a optimal subset is to be selected to satisfy the user constraint. All approaches were implemented on an Intel Core 2 Quad 2.5GHz PC with 4GB memory running Matlab 2010b (64bit). For the BIP optimization routine required in [2], we used the *binprog* function available in the optimization toolbox.

6.2 Performance Comparison

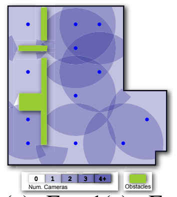
We first conducted 4 experiments using floor plan A and B to evaluate the performance of the proposed approach against the alternative approaches: Erdem06 [2], Zhao09 [4] and Horster09 [3] under different setups. All the experiments conducted have the same objective: minimizing the number of cameras used to achieve a 100% coverage of the two floor plans. To compute the likelihood (as required by Eqn. 5) of a particular camera configuration in satisfying the required constraints, we first employ the visibility computation routine described in [2] to compute the coverage areas of all the cameras in the configuration, which is denoted as COV (ϕ). Then the likelihood can be computed by $L(\mathbf{r} \mid \phi) \propto \exp\{-(\text{COV}(\phi) - r_{cov})^2/2\sigma^2\}$, where r_{cov} is the desired coverage percentage, which in this case is 100% and σ is set to 0.1.

The separation between each camera location samples were 90, 60 and 30 pixels in both x and y directions for the 4 experiments respectively. Two types of cameras were employed: omni-direction cameras with $360°$ field of view (FoV) and cameras with $120°$ FoV. Both types of cameras have depth of 150pixels and the orientation of the $120°$ FoV camera is sampled every $20°$. The 4 experimental setups are summarized in Table. 1. For TDSA, we also used the following parameters: $T_0 = 10$, $T_e = 10^{-4}$, $\rho = 0.99$, $\gamma = 10^{-5}$, $n_{max} = 200$, $p_b = p_d = 0.2$. The results are plotted in Fig. 3(a) and Fig. 3(b) and the computed placements strategies of Exp. 1 and Exp. 3 are shown in Fig. 2.

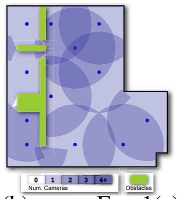
Table 1. Experiments conducted for comparing different approaches

	Floor plan	Cam. loc. sep.	Cam. FoV	Ori. sep.	Depth	Num. candidate
Exp. 1(a)	A	90pixel	360°	NA	150°	28
Exp. 1(b)	B	90pixel	360°	NA	150°	28
Exp. 2(a)	A	60pixel	360°	NA	1500°	79
Exp. 2(b)	B	60pixel	360°	NA	1500°	64
Exp. 3(a)	A	30pixel	360°	NA	1500°	304
Exp. 3(b)	B	30pixel	360°	NA	1500°	251
Exp. 4(a)	A	30pixel	120°	20°	150°	5472
Exp. 4(b)	B	30pixel	120°	20°	150°	4518

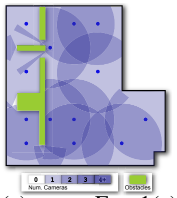
The Matlab implementation of BIP (used in Erdem06) uses a branch and bound algorithm which in the worst case scenario will visit all the possible combinations. In Exp.1 and Exp.2, the BIP algorithm took less than 10 seconds to complete but for Exp.3 and Exp.4, where the search spaces were approx. 2^{300} and 2^{5000}, it was unable to find a solution in feasible time (48 hours in our case). This severely limits the applicability of Erdem and Sclaroff's approach to larger scale problems. Our approach, on the other hand, takes a few hours for each experiment despite the rapid growth of search space size from Exp.1 to Exp.4 (Exp.4(b) took 5.5 hours on a 2.5GHz PC running matlab). It can be seen by inspecting Fig. 2(a) and 2(b), Fig. 2(e) and 2(f) that both Erdem06 and TDSA produced almost identical results, meaning optimal solutions have been found by TDSA. Comparing the proposed approach with the other 2 heuristics in Fig. 2, it is evident that the results of Horster09 and Zhao09 are much more cluttered, especially in cases when the search spaces are relatively large (comparing Fig. 2(i) with 2(j), Fig. 2(k) with 2(l), 2(m), 2(n)). The same conclusion can be drawn by inspecting the two summary plots in Fig. 4(a) and Fig. 3(b), where the number of cameras used to cover each floor plan computed by all 4 methods are plotted. Overall, the strategies computed using our TDSA method require 18.6% and 21.2% less cameras than Horster09 and Zhao09 respectively.



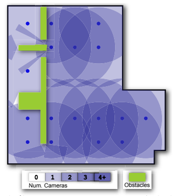
(a) Exp.1(a) Erdem06



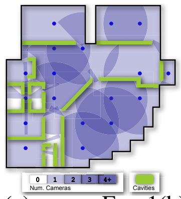
(b) Exp.1(a) TDSA



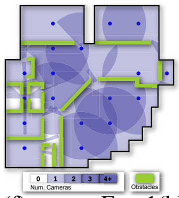
(c) Exp.1(a) Horster09



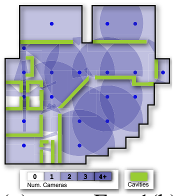
(d) Exp.1(a) Zhao09



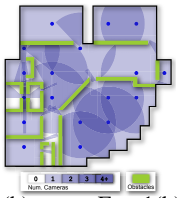
(e) Exp.1(b) Erdem06



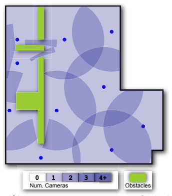
(f) Exp.1(b) TDSA



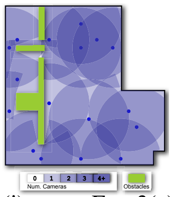
(g) Exp.1(b) Horster09



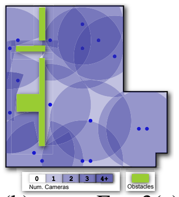
(h) Exp.1(b) Zhao09



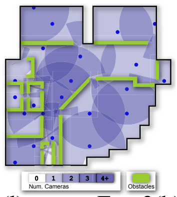
(i) Exp.3(a) TDSA



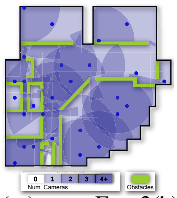
(j) Exp.3(a) Horster09



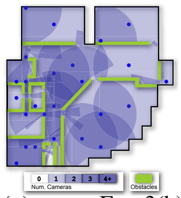
(k) Exp.3(a) Zhao09



(l) Exp.3(b) TDSA



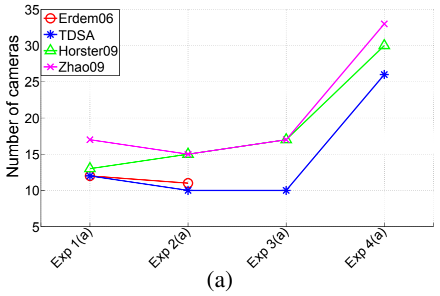
(m) Exp.3(b) Horster09



(n) Exp.3(b) Zhao09

Fig. 2. Plot of the camera configurations computed by all 4 methods for Exp. 1 and 3

In addition, our approach is more flexible as we penalize the increases in the number of cameras that do not bring much gain to the coverage. By comparing the results of Erdem06 and TDSA in Fig. 3(a) and Fig. 3(a), it appears that our TDSA outperforms the optimal BIP method (Erdem06) in Exp.2. This is caused by the fact that our approach omitted a few pixels in the corners, achieving coverage of 99.99% as opposed to 100%. This behavior is primarily due to the penalty of the extra camera which outweighed the 0.01% coverage increase in the objective function (Eqn. 5) and thus was not included in the optimal set.



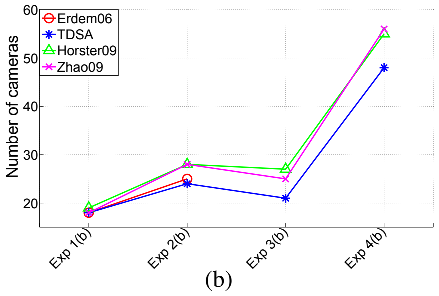


Fig. 3. Number of cameras computed to cover floor plan A (a) and floor plan B (b). Note that Erdem06 was not able to produce results for Exp.3 and Exp.4 since the large search spaces could not be handled by BIP.

6.3 Example of a Different User Objective

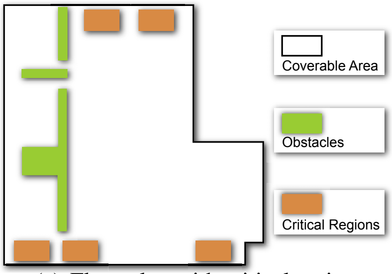
The objective used in the previous experiments (Exp.1-4) is only a particular case of the various objectives that can be dealt with by our proposed formulation of the problem. In this section, we show the result of computing optimal camera configuration for a different objective and provide an easy way to define likelihood functions. In real deployment of a camera network for surveillance purposes, covering the total area to a pre-defined level is often not the only requirement. It is also sometimes a necessity that some critical areas such as entrances to prohibited areas are to be monitored by more than one cameras in order to provide redundancy in capturing frontal faces. Shown in Fig. 4(a) is a modified version of floor plan A (Fig. 1(a)). The difference is the addition of a number of critical regions which are to be covered by at least two cameras. The total area, as usual, is required to be covered 100%. The likelihood function required in Eqn. 5 can be written as the product of the two likelihoods that correspond to the two separate objectives,

$$L(\mathbf{r} \mid \phi) \propto \exp\left\{-\left(\mathrm{COV}_{total}(\phi) - r_{total}\right)^2 / 2\sigma_{total}^2\right\} \times \exp\left\{-\left(\mathrm{COV}_{crit}(\phi) - r_{crit}\right)^2 / 2\sigma_{crit}^2\right\}, \quad (12)$$

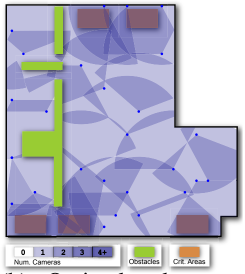
where COV_{total} and COV_{total} are the percentage of the total area and the percentage of critical regions that are covered by the required number of cameras. r_{total} and r_{crit} are the desired coverage percentage which in this case is 100%. σ_{total} and σ_{crit} is set to 0.1.

The parameters used in this experiment were the same as those used in Exp.4(a) which is listed in Table 1. The following parameters were used as input to the proposed TDSA algorithm to compute the optimal camera configurations: $T_0 = 10$, $T_e = 10^{-4}$, $\rho = 0.99$, $\gamma = 10^{-5}$, $n_{max} = 200$, $p_b = p_d = 0.2$.

The resultant placement strategy is shown in Fig. 4(b) and the total number of cameras used is 27. Since the total number of candidate cameras is 5472, we were not able to compute the true optimal strategy by using BIP. However, as can be seen from



(a) Floor plan with critical regions



(b) Optimal placement strategy

Fig. 4. Optimal placement strategy with critical regions covered by at least 2 cameras and other area covered by at least 1 camera

Fig. 4(b), the whole area is covered by at least one camera and those critical regions are fully covered by at least 2 cameras.

7 Conclusions

In this paper we presented an approach to the problem of determining the optimal camera configurations in multi-camera networks. This is an important and still unsolved problem and optimal solutions will be of significant benefit in the design of multi-camera surveillance networks. Our approach includes a generalized statistical formulation of the problem, taking into account a set of user constraints, the number of cameras and the parameters of the cameras. We developed a trans-dimensional simulated annealing algorithm to compute the optimal configuration. To evaluate the performance we have compared our approach with a state-of-the-art methods described in [2] and show that similar performance to the optimal BIP solution can be obtained. Comparing to the other two heuristics designed to cope with larger scale problems [3, 4] when BIP cannot handle, the configuration computed by our proposed approach requires 18% less cameras than [3] and 21% less cameras than [4]. In order to demonstrate the flexibility of our proposed approach, we considered a more realistic user objective: 100% coverage is required for the whole area and a number of critical regions are to be fully covered by at least two cameras. Our results show that the proposed approach can successfully solve this problem.

References

1. Fookes, C., Denman, S., Lakemond, R., Ryan, D., Sridharan, S., Piccardi, M.: Semi-supervised intelligent surveillance system for secure environments. In: Proceedings of the IEEE Industrial Electronics Symposium, pp. 2815–2820 (2010)
2. Erdem, U.M., Sclaroff, S.: Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements. Computer Vision and Image Understanding 103(3), 156–169 (2006)

3. Horster, E., Lienhart, R.: 5, Optimal Placement of Multiple Visual Sensors. In: Multi-Camera Networks: Concepts and Applications. Elsevier (2009)
4. Zhao, J., Cheung, S.C.S., Nguyen, T.: 6, Optimal Visual Sensor Network Configuration. In: Multi-Camera Networks: Concepts and Applications. Elsevier (2009)
5. Fleishman, S., Cohen-Or, D., Lischinski, D.: Automatic camera placement for image-based modeling. In: Proceedings of Seventh Pacific Conference on Computer Graphics and Applications, Fleishman 1999, pp. 12–20, 315 (1999)
6. Mordohai, P., Medioni, G.: Dense multiple view stereo with general camera placement using tensor voting. In: Proceedings of 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 3DPVT 2004, pp. 725–732 (2004)
7. Olague, G., Mohr, R.: Optimal camera placement to obtain accurate 3d point positions. In: Proceedings of Fourteenth International Conference on Pattern Recognition, vol. 1, pp. 8–10 (1998)
8. 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 Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 40(1), 101–115 (2010)
9. Bodor, R., Drenner, A., Schrater, P., Papanikolopoulos, N.: Optimal camera placement for automated surveillance tasks. Journal of Intelligent & Robotic Systems 50(3), 257–295 (2007)
10. O'Rourke, J.: Art gallery theorems and algorithms. Oxford University Press, Inc. (1987)
11. Ram, S., Ramakrishnan, K.R., Atrey, P.K., Singh, V.K., Kankanhalli, M.S.: A design methodology for selection and placement of sensors in multimedia surveillance systems. In: Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks, 1178801, pp. 121–130. ACM (2006)
12. Mittal, A., Davis, L.S.: A general method for sensor planning in multi-sensor systems: Extension to random occlusion. Int. J. Comput. Vision 76(1), 31–52 (2008)
13. Akaike, H.: A new look at the statistical model identification. IEEE Transactions on Automatic Control 19(6), 716–723 (1974)
14. Brooks, S.P., Friel, N., King, R.: Classical model selection via simulated annealing. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 65(2), 503–520 (2003)
15. Andrieu, C., Freitas, N.d., Doucet, A.: Reversible jump mcmc simulated annealing for neural networks. In: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, 719898, pp. 11–18. Morgan Kaufmann Publishers Inc (2000)
16. Granville, V., Krivanek, M., Rasson, J.-P.: Simulated annealing: A proof of convergence. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(6), 652–656 (1994)
17. Schwarz, G.: Estimating the dimension of a model. The Annals of Statistics 6(2), 461–464 (1985)
18. Rissanen, J.: Stochastic complexity. Journal of the Royal Statistical Society. Series B (Methodological) 49(3), 223–239 (1987)
19. Kirkpatrick, S.: Optimization by simulated annealing: Quantitative studies. Journal of Statistical Physics 34(5), 975–986 (1984)
20. Green, P.J.: Reversible jump markov chain monte carlo computation and bayesian model determination. Biometrika 82(4), 711–732 (1995)