Design of Multimedia Surveillance Systems

G. S. V. S. SIVARAM
Johns Hopkins University, USA
MOHAN S. KANKANHALLI
National University of Singapore
and
K. R. RAMAKRISHNAN
Indian Institute of Science

This article addresses the problem of how to select the optimal combination of sensors and how to determine their optimal placement in a surveillance region in order to meet the given performance requirements at a minimal cost for a multimedia surveillance system. We propose to solve this problem by obtaining a performance vector, with its elements representing the performances of subtasks, for a given input combination of sensors and their placement. Then we show that the optimal sensor selection problem can be converted into the form of Integer Linear Programming problem (ILP) by using a linear model for computing the optimal performance vector corresponding to a sensor combination. Optimal performance vector corresponding to a sensor combination refers to the performance vector corresponding to the optimal placement of a sensor combination. To demonstrate the utility of our technique, we design and build a surveillance system consisting of PTZ (Pan-Tilt-Zoom) cameras and active motion sensors for capturing faces. Finally, we show experimentally that optimal placement of sensors based on the design maximizes the system performance.

Categories and Subject Descriptors: I.6.4 [**Simulation and Modeling**]: Model Validation and Analysis

General Terms: Design, Security

Additional Key Words and Phrases: Performance vector, sensor selection and placement

ACM Reference Format:

Sivaram, G. S. V. S., Kankanhali, M. S., and Ramakrishnan, K. R. 2009. Design of multimedia surveillance systems. ACM Trans. Multimedia Comput. Commun. Appl. 5, 3, Article 23 (August 2009), 25 pages.
DOI = 10.1145/1556134.1556140 http://doi.acm.org/ 10.1145/1556134.1556140

1. INTRODUCTION

Most multimedia surveillance systems nowadays utilize multiple types of sensors which have different capabilities and which are of different costs for accomplishing a surveillance task. In general, a surveillance task constitutes of a set of subtasks. For example, if the surveillance task is to capture the face of a person who is shouting in a room then the associated subtasks are: (1) determine whether somebody is shouting or not based on the acoustics of the room; (2) localize the person; (3) capture the

Authors' addresses: G. S. V. S. Sivaram, Department of Electrical and Computer Engineering, Johns Hopkins University; email: sivaram@jhu.edu; M. S. Kankanhalli, National University of Singapore, Singapore; K. R. Ramakrishnan, Indian Institute of Science, Bangalore, India.

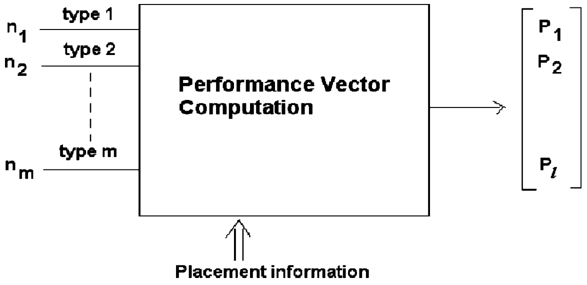


Fig. 1. Block diagram of the performance vector computation, n_i: number of ith type of sensors, and P_j: performance of the jth subtask.

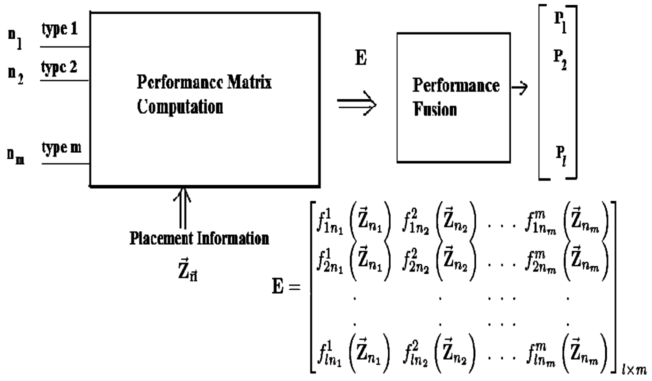

n_1
type 1
n_2
type 2
n_m
type m
Performance Matrix Computation
E
Performance Fusion
P_1
P_2
P_l
Placement Information
$\vec{Z}_{\vec{n}}$
$\mathbf{E} = \begin{bmatrix} f^1_{1n_1}(\vec{Z}_{n_1}) & f^2_{1n_2}(\vec{Z}_{n_2}) & \cdots & f^m_{1n_m}(\vec{Z}_{n_m}) \\ f^1_{2n_1}(\vec{Z}_{n_1}) & f^2_{2n_2}(\vec{Z}_{n_2}) & \cdots & f^m_{2n_m}(\vec{Z}_{n_m}) \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ f^1_{ln_1}(\vec{Z}_{n_1}) & f^2_{ln_2}(\vec{Z}_{n_2}) & \cdots & f^m_{ln_m}(\vec{Z}_{n_m}) \end{bmatrix}_{l\times m}$

Fig. 2. Cascade form, **E** : performance matrix.

image at the specified location. We analyzed the performance of each subtask in this work and it is clearly a function of the combination of sensors and their placement in a surveyed region. Note that the performance of the main surveillance task depends on the performances of the subtasks. However, we restrict our attention to ensuring a desired performance for each one of the subtasks at minimal cost in this article.

The following design problem has been addressed in this work. Given a set of m types of sensors that can be deployed in a surveyed region and also given a set of l subtasks along with the minimum required (desired) performance for each one of the subtasks, the problem is to find the combination of sensors along with their placement that minimizes the overall surveillance system cost. The motivation for considering this problem is that, to the best of our knowledge, there is no quantitative design technique for designing a surveillance system consisting of multiple types of sensors. Also, our interactions with the industry indicate that an ad hoc methodology is usually employed for designing such systems.

Imagine a model that takes the combination of sensors and their placement in a surveyed region as input and gives the performance vector as output, as shown in Figure 1. The elements of the performance vector represent the performances (quantified in the range [0, 1]) of subtasks. Our problem is to determine the optimal combination of sensors and their placement (input to the model) for a given desired performance vector (output of the model). To solve this problem, we simplify the model that determines the effect of each sensor type independently at a time and subsequently fuses all such effects to obtain the performance vector, as shown in Figure 2. Optimal performance vector corresponding to a sensor combination refers to the performance vector corresponding to the optimal placement of a sensor combination. Later we show that the optimal sensor selection problem can be converted into a form of the Integer Linear Programming problem (ILP) by using a linear model for computing this optimal performance vector. In fact, ILP is a common technique used in the field of sensor placement. Also we describe a special case of this wherein the optimal combination of sensors is obtained by solving each of the l inequalities independently at a time for its minimum integer argument. Note that we are not separating the tasks of finding the optimal set of sensors and their placement in this analysis. Instead, we are using a simple linear model for computing the optimal performance vector (corresponding to a sensor combination) in the formulation of the optimal sensor selection problem. This approach will ensure that we pick the optimal performance vector corresponding to the solution of the optimal sensor selection problem while designing a surveillance system.

To demonstrate how our technique can be used for designing a surveillance system, we consider a surveillance task of capturing the frontal face of an intruder in a surveyed region. This task has two subtasks: object (intruder) localization and image capture. Two types of sensors are considered for deployment, namely PTZ (Pan-Tilt-Zoom) infrared cameras and active motion sensors. Now the problem is to determine the optimal number of cameras and motion sensors along with their placement in a surveyed region such that a minimum specified performance is guaranteed for each subtask. Later in this article, we derive the effect of cameras and the 2D motion sensor grid on the performance of each subtask in order to obtain a solution based on the proposed technique. We also build a surveillance system using the optimal combination of sensors and their placement information obtained after following the previously mentioned design technique. Experimental results confirm that optimal sensor placement based on the design maximizes system performance.

A preliminary version of this work was published in VSSN 2006 wherein Sivaram et al. [2006] proposed a design methodology for optimal selection and placement of sensors in multimedia surveillance systems. Our idea is to derive a performance metric for the surveillance task as a function of the set of sensors and their placement and then to determine the optimal sensor combination based on the performance metric. However, deriving such a performance metric for any particular interaction strategy is not straightforward with an increasing number of sensor types. Therefore, in this article, we divide the surveillance task into subtasks and then consider the effect of each sensor type independently on each of the subtasks. To the best of our knowledge, this is the first time that such a design problem has been addressed for building a surveillance system consisting of multiple types of sensors and that performs multiple subtasks.

The remainder of this article is organized as follows. Section 2 presents the related work. In Section 3, we describe the proposed method for obtaining the optimal combination of sensors and their placement. In Section 4, we discuss a specific surveillance system design and then present the experimentation results in Section 5. Finally, in Section 6, we conclude the article with a discussion on future work.

2. RELATED WORK

Use of multisensors in different applications including surveillance is a rapidly evolving research area [Luo et al. 2002]. Luo et al. [2002] provide an overview of current sensor technologies and describe the paradigm of multisensor fusion and integration for various applications. Luo and Kay [1989] reviewed the sensor selection strategies, namely preselection during design and real-time selection in response to system conditions, for multisensor integration and fusion. However, their sensor selection strategy is based on the processing time and operating speed of the sensors, whereas the proposed sensor selection technique minimizes the overall system cost and guarantees the required performance for each of the subtasks.

Recently, efforts [Bodor et al. 2007] were made to tackle the problem of task-specific camera placement, in which the authors optimized camera placement to maximize observability of the set of actions performed in a defined area. Horster and Lienhart [2006] have proposed an algorithm to minimize the cost of a visual sensor array while ensuring a proper coverage. For tracking applications, Chen and Davis [2000] have developed a resolution metric for camera placement considering occlusions. Note that the aforesaid works deal with visual sensors (cameras) alone. Erdem and Sclaroff [2004] have described an approach for placement of multiple cameras in a polygonal space using "reasonable" assumptions for real-life cameras. Mittal and Davis [2004] have also described a method for determining the optimal number of cameras and their placement to monitor any given premises. However, both the aforementioned works do not consider direction of image captured as part of their suitability metrics. This is important, as it is often necessary to obtain images in one direction (e.g., frontal direction for face/object recognition, etc.) and not the other. Also, they consider only cameras while we address the problem for multiple types of sensors. And we demonstrate our technique by considering only two types of sensors: PTZ cameras and active motion sensors.

On the other hand, motion sensor is extensively used for space surveillance [Zierhut] and is a key component in burglar alarm systems. Howard et al. [2002] have proposed a method for localizing the members of a mobile robot team where a motion sensor is assumed to estimate changes in the pose of each robot. In other work, Wren et al. [2005] utilize a swarm of low-cost motion sensors for automatically calibrating PTZ cameras that are undertaking surveillance tasks. Their work, however, does not deal with finding the optimal number and position, etc., for the sensors.

Vision sensor planning has been studied extensively in the field of robotics. There are several surveys available in the literature [Scott et al. 2003; Tarabanis et al. 1995] regarding this same issue. Chen and Li [2004] propose automatic sensor placement by a genetic algorithm for model-based robot vision. In the context of wireless sensor networks, Pahalawatta et al. [2004] propose to solve the problem of optimal sensor selection by maximizing the information utility gained from a set of sensors subject to a constraint on the average energy consumption in the network. A sensor placement algorithm for optimizing the coverage area has been reported in Dhillon and Chakrabarty [2003]. They assume that the probability of detecting a target by using a sensor varies exponentially with the distance between target and sensor. However, the aforementioned works have not addressed the selection and placement of multiple types of sensors.

On the whole we realize that while optimal sensor selection and placement has generated reasonable research interest, the currently available methods fail to handle multiple types of sensors. To the best of our knowledge, this article is the first to address this issue.

3. PROPOSED TECHNIQUE

In general, a surveillance system performs multiple numbers of tasks in a surveyed region. In order to accomplish a particular task, we may be required to perform some smaller subtasks in series or in parallel. Usually, different sensor types are capable of accomplishing a particular subtask with varying performance and cost. Also sometimes a particular sensor type can accomplish more than one subtask. Given this scenario, it is important to find the optimal combination of sensors along with their placement to minimize the overall system cost. We address this problem in this section.

Notation.
Number of sensor types: m
Cost of one quantity of j*th type sensor :* c_j
Cost vector: $\vec{\mathbf{C}} = \begin{bmatrix} c_1 & c_2 \cdots c_m \end{bmatrix}^{\mathrm{T}}$

Number of jth *type sensors :* n_j
Sensor combination: $\vec{\mathbf{n}} = [n_1\, n_2 \cdots n_m]^{\mathrm{T}}$
Sensor placement: $\vec{\mathbf{Z}}_{\vec{\mathbf{n}}} = [\vec{\mathbf{Z}}_{n_1} \vec{\mathbf{Z}}_{n_2} \cdots \vec{\mathbf{Z}}_{n_m}]^{\mathrm{T}}$
Number of subtasks: l
Required performance for the ith subtask : $r_i \ \in [0,\ 1]$
Required performance vector: $\vec{\mathbf{b}} = [r_1\, r_2 \cdots r_l]^{\mathrm{T}}$

Problem Definition. Find the combination of sensors $\vec{\boldsymbol{n}}$ along with their placement $\vec{\boldsymbol{Z}}_{\vec{\boldsymbol{n}}}$ in a given surveyed region in order to minimize the overall cost $\vec{\boldsymbol{C}}.\vec{\boldsymbol{n}}$ such that the following constraints are satisfied.

(*1*) *Performance constraints :*
performance of the ith subtask $\geq r_i,\ \forall\, i = 1,\ 2, \ldots,\ l$

(*2*) *Integer constraints :*
n_j *must be a nonnegative integer*, $\forall\, j = 1,\ 2, \ldots,\ m.$

To solve this problem, we need to identify whether a particular combination of sensors along with their placement is feasible. The idea is to obtain a computation model that takes a sensor combination $\vec{\mathbf{n}}$ along with its placement information as input and computes the performance vector $\vec{\mathbf{P}} = [P_1 P_2 \ldots P_l]^{\mathrm{T}}$ as shown in Figure 1. We denote the sensor placement information (corresponding to $\vec{\mathbf{n}}$) using the sensor placement vector $\vec{\mathbf{Z}}_{\vec{\mathbf{n}}}$ and the components of it can be written as $\vec{\mathbf{Z}}_{\vec{\mathbf{n}}} = [\vec{\mathbf{Z}}_{n_1}\ \vec{\mathbf{Z}}_{n_2} \ldots \vec{\mathbf{Z}}_{n_m}]^{\mathrm{T}}$. (This is where $\vec{\mathbf{Z}}_{n_j}$ is a row vector, and represents the placement information of n_j number of j type sensors in a surveyed region. If the jth type of sensor can be placed at a point (for example, camera) then we can think of elements of $\vec{\mathbf{Z}}_{n_j}$ as some collection of n_j points in 3D space). The ith component of the performance vector, namely P_i, indicates the performance (quantified in the range $[0,\ 1]$) with which a surveillance system built using the input information ($\vec{\mathbf{n}}$ and $\vec{\mathbf{Z}}_{\vec{\mathbf{n}}}$) accomplishes the ith subtask. Furthermore, for a given input combination $\vec{\mathbf{n}}$, P_i varies as some function of $\vec{\mathbf{Z}}_{\vec{\mathbf{n}}}$. The following equation represents the functional dependence of P_i on $\vec{\mathbf{Z}}_{\vec{\mathbf{n}}}$ for a given $\vec{\mathbf{n}}$.

$$P_i \ = \ F_{i\vec{\mathbf{n}}}(\vec{\mathbf{Z}}_{\vec{\mathbf{n}}}) \ \in [0,\ 1],\ \forall\, i \ = \ 1,\ 2, \ldots, l. \tag{1}$$

Hence the performance constraints become

$$\vec{\mathbf{P}} \ \geq \ \vec{\mathbf{b}} \ \Rightarrow \ F_{i\vec{\mathbf{n}}}\left(\vec{\mathbf{Z}}_{\vec{\mathbf{n}}}\right) \ \geq r_i,\ \forall\, i \ = \ 1,\ 2, \ldots, l.$$

The preceding inequalities are mathematically not tractable, as the functional dependence ($F_{i\vec{\mathbf{n}}}(.)$ for all $\vec{\mathbf{n}}$) is not apparent. Therefore, we simplify the model for computing the performance vector as described next.

Since we know what each sensor type is capable of doing, we determine the effect of each sensor type independently and then combine all such effects to obtain the performance vector. Hence we divide the performance vector computation block in Figure 1 into two blocks, namely the performance matrix computation block and performance fusion block, in cascade form as shown in Figure 2. The idea is to first determine the performance matrix (let us denote it with $\mathbf{E}$), whose columns represent the performance vectors obtained by considering each sensor type independently at a time, and then fuse all of its columns to obtain the performance vector for the given $\vec{\mathbf{n}}$ and $\vec{\mathbf{Z}}_{\vec{\mathbf{n}}}$. The performance matrix computation block in Figure 2 takes $\vec{\mathbf{n}}$ and $\vec{\mathbf{Z}}_{\vec{\mathbf{n}}}$ as input and gives performance matrix $\mathbf{E}$ as output. The dimension of the matrix $\mathbf{E}$ is $l \times m$ as its jth column represents the performance vector obtained due to n_j the number of j type sensors placed at location $\vec{\mathbf{Z}}_{n_j}$ ($\forall\, j \ = \ 1,\ 2, \ldots, m$). Therefore, $\mathbf{E}_{i,j}$, (i, j)th element of the matrix $\mathbf{E}$, represents the performance with which n_j number of j type sensors placed at location $\vec{\mathbf{Z}}_{n_j}$ accomplish the ith subtask. We quantified and normalized $\mathbf{E}_{i,j}$ in our analysis and its range is $[0,\ 1]$, where 0 indicates “not capable of accomplishing the subtask” and 1 indicates “capable of accomplishing the subtask perfectly.” Furthermore, $\mathbf{E}_{i,j}$ varies as some function of $\vec{\mathbf{Z}}_{n_j}$ for a given n_j. Thus,

$$\mathbf{E}_{i,j} = f^{j}_{in_j}\left(\vec{\mathbf{Z}}_{n_j}\right) \in [0,\ 1], \tag{2}$$

where:

(1) j indicates the sensor type.
(2) n_j indicates the total number of sensors (of the previous type).
(3) i indicates the subtask number.

Therefore, the performance matrix can be written as

$$\mathbf{E} = \begin{bmatrix} f^{1}_{1n_1}\left(\vec{\mathbf{Z}}_{n_1}\right) & f^{2}_{1n_2}\left(\vec{\mathbf{Z}}_{n_2}\right) & \cdots & f^{m}_{1n_m}\left(\vec{\mathbf{Z}}_{n_m}\right) \\ f^{1}_{2n_1}\left(\vec{\mathbf{Z}}_{n_1}\right) & f^{2}_{2n_2}\left(\vec{\mathbf{Z}}_{n_2}\right) & \cdots & f^{m}_{2n_m}\left(\vec{\mathbf{Z}}_{n_m}\right) \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ f^{1}_{ln_1}\left(\vec{\mathbf{Z}}_{n_1}\right) & f^{2}_{ln_2}\left(\vec{\mathbf{Z}}_{n_2}\right) & \cdots & f^{m}_{ln_m}\left(\vec{\mathbf{Z}}_{n_m}\right) \end{bmatrix}_{l\times m}. \tag{3}$$

The ith row elements of the matrix $\mathbf{E}$ denote the performances with which different sensor types accomplish the ith subtask. But the final performance with which the surveillance system accomplishes the ith subtask depends not only on the ith row elements of $\mathbf{E}$ but also on the following:

(1) the interaction strategy among the sensor types; and
(2) the percentage of overlap in the information that different sensor types sense for accomplishing the ith subtask.

Thus, the performance fusion model (refer to Figure 2) should take into consideration all the preceding factors while fusing the ith row elements of $\mathbf{E}$ to obtain the overall performance of the ith subtask. Let the function $G_i(.)$ represent the fusion model for the ith subtask. Therefore P_i can be written as

$$P_i = G_i\left(f^{1}_{in_1}\left(\vec{\mathbf{Z}}_{n_1}\right),\ f^{2}_{in_2}\left(\vec{\mathbf{Z}}_{n_2}\right), \ldots, f^{m}_{in_m}\left(\vec{\mathbf{Z}}_{n_m}\right)\right), \quad \forall\, i = 1,\ 2, \ldots, l. \tag{4}$$

Note from Eq. (4) that for a given sensor combination $\vec{\mathbf{n}}$, performance vector elements vary as some function of $\vec{\mathbf{Z}}_{\vec{\mathbf{n}}}$. This fact is also obvious from Eq. (1). The following section discusses how to determine the optimal sensor placement $\vec{\mathbf{Z}}^{*}_{\vec{\mathbf{n}}}$ for a given sensor combination $\vec{\mathbf{n}}$. This information can be used to compute the optimal performance vector corresponding to a sensor combination $\vec{\mathbf{n}}$, and Section 3.2 describes the linear model for computing this optimal performance vector.

3.1 Optimal Sensor Placement

Imagine a *l-dimensional* space $\mathbf{R}^{l}$ with its coordinate axes representing the performances of the subtasks. Hence we can represent any performance vector as a point in the *l-dimensional* space $\mathbf{R}^{l}$. From Eq. (1), we note that each element of the performance vector varies as some function of $\vec{\mathbf{Z}}_{\vec{\mathbf{n}}}$ for a given sensor combination $\vec{\mathbf{n}}$. So, for a given $\vec{\mathbf{n}}$, a set of performance vectors can be obtained by varying $\vec{\mathbf{Z}}_{\vec{\mathbf{n}}}$ over

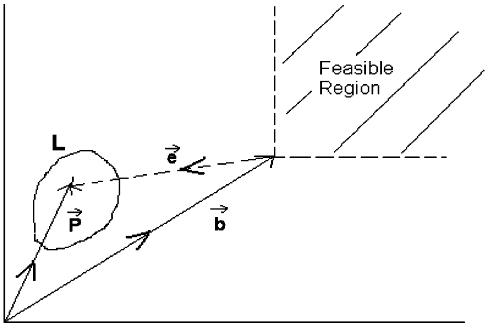


Fig. 3. Optimality condition for any given sensor combination $\vec{\mathbf{n}}$.

all its possible values. Imagine this set of performance vectors forming a region $\mathbf{L}$ in the *l-dimensional* space as shown in Figure 3. We know that any vector in the feasible region must satisfy the constraints $P_i \geq r_i, \forall i = 1, 2, \ldots, l$. We can think of the inequality $P_i \geq r_i$ as a region to the right of the $(l-1)$-dimensional hyperplane $P_i = r_i$ in the *l-dimensional* space. Thus a feasible region is a *l-dimensional* hypercube, obtained by taking the intersection of l spaces $P_i \geq r_i, \forall i = 1, 2, \ldots, l$. Out of all vectors in the feasible region, the vector $\vec{\mathbf{b}}$ is nearest (Euclidean distance) to the origin (see Figure 3).

For any vector $\vec{\mathbf{P}} \in \mathbf{L}$, we can find the corresponding (unique) vector $\vec{\mathbf{e}}$ such that (refer to Figure 3)

$$\vec{\mathbf{b}} + \vec{\mathbf{e}} = \vec{\mathbf{P}} \Rightarrow \vec{\mathbf{e}} = \vec{\mathbf{P}} - \vec{\mathbf{b}}. \tag{5}$$

For a given sensor combination $\vec{\mathbf{n}}$, we determine the optimal sensor placement $\vec{\mathbf{Z}}^*_{\vec{\mathbf{n}}}$ and the corresponding optimal performance vector $\vec{\mathbf{P}}^*$ by maximizing the dot product $\vec{\mathbf{e}}.\vec{\mathbf{b}}$ over the region $\mathbf{L}$. In other words, we define $\vec{\mathbf{Z}}^*_{\vec{\mathbf{n}}}$ as the placement that maximizes the component of $\vec{\mathbf{e}}$ in the direction of $\vec{\mathbf{b}}$. Furthermore, it is sometimes possible that the optimal placement $\vec{\mathbf{Z}}^*_{\vec{\mathbf{n}}}$ is not unique according to our definition, as any vector which is orthogonal to $\vec{\mathbf{b}}$ can be added to $\vec{\mathbf{e}}$ without changing the dot product value. However, in this case, we have to choose the placement that minimizes the Euclidean distance between the performance vector and its component in the direction of $\vec{\mathbf{b}}$, and that maximizes the dot product $\vec{\mathbf{e}}.\vec{\mathbf{b}}$. From Eq. (5),

$$\vec{\mathbf{e}}.\vec{\mathbf{b}} \Rightarrow (\vec{\mathbf{P}} - \vec{\mathbf{b}}).\vec{\mathbf{b}}.$$

Since $\vec{\mathbf{b}}.\vec{\mathbf{b}}$ is a constant, we can find $\vec{\mathbf{Z}}^*_{\vec{\mathbf{n}}}$ for a given $\vec{\mathbf{n}}$ by maximizing $\vec{\mathbf{P}}.\vec{\mathbf{b}}$ with respect to $\vec{\mathbf{P}}$ over the region $\mathbf{L}$. In other words, for a given $\vec{\mathbf{n}}$, we can find the optimal sensor placement $\vec{\mathbf{Z}}^*_{\vec{\mathbf{n}}}$ by maximizing $\vec{\mathbf{P}}.\vec{\mathbf{b}}$ with respect to $\vec{\mathbf{Z}}_{\vec{\mathbf{n}}}$. The dot product $\vec{\mathbf{P}}.\vec{\mathbf{b}}$ can be expanded using Eq. (4) as follows.

$$\vec{\mathbf{P}}.\vec{\mathbf{b}} = \sum_{i=1}^{l} r_i \times G_i \left(f^1_{in_1}\left(\vec{\mathbf{Z}}_{n_1}\right),\ f^2_{in_2}\left(\vec{\mathbf{Z}}_{n_2}\right), \ldots, f^m_{in_m}\left(\vec{\mathbf{Z}}_{n_m}\right)\right) \tag{6}$$

We can simplify this expression further by using the linear fusion model. The linear fusion model for the ith subtask is given by

$$\begin{aligned} P_i &= G_i \left(f^1_{in_1}\left(\vec{\mathbf{Z}}_{n_1}\right),\ f^2_{in_2}\left(\vec{\mathbf{Z}}_{n_2}\right), \ldots, f^m_{in_m}\left(\vec{\mathbf{Z}}_{n_m}\right)\right) \\ &= \sum_{j=1}^{m} f^j_{in_j}\left(\vec{\mathbf{Z}}_{n_j}\right), \forall i = 1, 2, \ldots, l. \end{aligned} \tag{7}$$

The aforesaid fusion model computes the performance of each subtask by adding the individual performances of different sensor types. This model is applicable only when different sensor types sense complementary information about each subtask. Otherwise, the value computed according to this model represents the upper bound on the performance of each subtask. However, we use this model in Eq. (6) to make the problem mathematically tractable and also to gain some insights. Thus we have

$$\begin{aligned}\vec{\mathbf{P}}.\vec{\mathbf{b}} &= \sum_{i=1}^{l} r_i \times \sum_{j=1}^{m} f_{in_j}^{j}\left(\vec{\mathbf{Z}}_{n_j}\right)\\ &= \sum_{i=1}^{l}\sum_{j=1}^{m} r_i \times f_{in_j}^{j}\left(\vec{\mathbf{Z}}_{n_j}\right).\end{aligned}$$

We can interchange the order of summation as the number of elements involved in the summation is finite. Hence, we have the following.

$$\begin{aligned}\vec{\mathbf{P}}.\vec{\mathbf{b}} &= \sum_{j=1}^{m}\sum_{i=1}^{l} r_i \times f_{in_j}^{j}\left(\vec{\mathbf{Z}}_{n_j}\right)\\ &= \sum_{j=1}^{m}\left(\sum_{i=1}^{l} r_i \times f_{in_j}^{j}\left(\vec{\mathbf{Z}}_{n_j}\right)\right)\\ &= \sum_{j=1}^{m} \mathbf{T}_j\left(\vec{\mathbf{Z}}_{n_j}\right) \qquad (8)\\ &\left(where\ \mathbf{T}_j\left(\vec{\mathbf{Z}}_{n_j}\right) = \sum_{i=1}^{l} r_i \times f_{in_j}^{j}\left(\vec{\mathbf{Z}}_{n_j}\right)\right)\end{aligned}$$

Using Eq. (2), we can write $\mathbf{T}_j(\vec{\mathbf{Z}}_{n_j})$ as

$$\mathbf{T}_j\left(\vec{\mathbf{Z}}_{n_j}\right) = \sum_{i=1}^{l} r_i \times \mathbf{E}_{i,j} = \vec{\mathbf{b}}.\vec{\mathbf{E}}_j, \qquad (9)$$

where $\vec{\mathbf{E}}_j$ represents the jth column of the performance matrix $\mathbf{E}$.

Note from the previous equation that $\mathbf{T}_j(\vec{\mathbf{Z}}_{n_j})$ represents the weighted combination of the jth column elements of the performance matrix $\mathbf{E}$. So if we determine the performance matrix $\mathbf{E}$ for a given $\vec{\mathbf{n}}$ and $\vec{\mathbf{Z}}_{\vec{\mathbf{n}}}$ then it is possible to determine the value of $\mathbf{T}_j(\vec{\mathbf{Z}}_{n_j})$ $(\forall\ j = 1, 2, \ldots, m)$.

In order to determine the performance matrix $\mathbf{E}$, we need to model the effect of each sensor type on the performance of each subtask. Depending on the the sensor type and the subtask at hand, we may get either a closed-form expression or some algorithmic procedure as a result of modeling. Therefore such a model takes n_j and $\vec{\mathbf{Z}}_{n_j}$ as input and gives $f_{in_j}^{j}(\vec{\mathbf{Z}}_{n_j})$ as output.

Furthermore, for a given $\vec{\mathbf{n}}$, maximizing $\vec{\mathbf{P}}.\vec{\mathbf{b}}$ with respect to $\vec{\mathbf{Z}}_{\vec{\mathbf{n}}}$ is same as maximizing the summation $\sum_{j=1}^{m} \mathbf{T}_j(\vec{\mathbf{Z}}_{n_j})$ with respect to $\vec{\mathbf{Z}}_{\vec{\mathbf{n}}}$ (using Eq. (8)). Note that no two functions $\mathbf{T}_i\,(.)$ and $\mathbf{T}_j\,(.)$, for different values of i and j, have common independent variables. Hence for a given $\vec{\mathbf{n}}$, we can maximize the summation $\sum_{j=1}^{m} \mathbf{T}_j(\vec{\mathbf{Z}}_{n_j})$ with respect to $\vec{\mathbf{Z}}_{\vec{\mathbf{n}}}$ by maximizing the individual functions $\mathbf{T}_j(\vec{\mathbf{Z}}_{n_j})$ with respect to $\vec{\mathbf{Z}}_{n_j}$ $(\forall\ j = 1, 2, \ldots, m)$ and then accumulating the maximum values. We summarize the procedure for obtaining the optimal sensor placement $\vec{\mathbf{Z}}_{\vec{\mathbf{n}}}^{*}$ for a given sensor combination $\vec{\mathbf{n}}$ next.

(1) Determine the optimal value of $\vec{\mathbf{Z}}_{n_j}$ (let us denote it as $\vec{\mathbf{Z}}^*_{n_j}$) which maximizes the function $\mathbf{T}_j(\vec{\mathbf{Z}}_{n_j})$ for a given n_j ($\forall\ j = 1, 2, \ldots, m$).
(2) Optimal sensor placement is given by

$$\vec{\mathbf{Z}}^*_{\vec{\mathbf{n}}} = \left[\vec{\mathbf{Z}}^*_{n_1}\ \vec{\mathbf{Z}}^*_{n_2}\ \ .\ .\ .\ \ \vec{\mathbf{Z}}^*_{n_m}\right]^{\mathrm{T}}.$$

(Note that we haven't discussed the problem of how to determine the optimal value $\vec{\mathbf{Z}}^*_{n_j}$ of a function $\mathbf{T}_j(\vec{\mathbf{Z}}_{n_j})$, as it depends on the form of the function. It could be a brute-force search in the entire placement space in the worst case. Also in some specific cases [Leskovec et al. 2007], it is possible to devise efficient algorithms for determining the optimal placement of sensors.)

In the next section, we describe a linear model for the computation of the optimal performance vector $\vec{\mathbf{P}}^*$ corresponding to the sensor combination $\vec{\mathbf{n}}$.

3.2 Linear Model for the Optimal Performance Vector

Let us represent the performance matrix corresponding to the optimal sensor placement $\vec{\mathbf{Z}}^*_{\vec{\mathbf{n}}}$ as $\mathbf{E}^*$. Thus from Eq. (3) we have

$$\mathbf{E}^* = \begin{bmatrix} f^1_{1n_1}\left(\vec{\mathbf{Z}}^*_{n_1}\right) & f^2_{1n_2}\left(\vec{\mathbf{Z}}^*_{n_2}\right) & .\ .\ . & f^m_{1n_m}\left(\vec{\mathbf{Z}}^*_{n_m}\right) \\ f^1_{2n_1}\left(\vec{\mathbf{Z}}^*_{n_1}\right) & f^2_{2n_2}\left(\vec{\mathbf{Z}}^*_{n_2}\right) & .\ .\ . & f^m_{2n_m}\left(\vec{\mathbf{Z}}^*_{n_m}\right) \\ . & . & .\ .\ . & . \\ . & . & .\ .\ . & . \\ f^1_{ln_1}\left(\vec{\mathbf{Z}}^*_{n_1}\right) & f^2_{ln_2}\left(\vec{\mathbf{Z}}^*_{n_2}\right) & .\ .\ . & f^m_{ln_m}\left(\vec{\mathbf{Z}}^*_{n_m}\right) \end{bmatrix}.$$

The preceding matrix $\mathbf{E}^*$ is fixed for a given sensor combination $\vec{\mathbf{n}}$. Thus we note that $\mathbf{E}^*$ varies as some function of the sensor combination $\vec{\mathbf{n}}$. Also, the jth column elements of the matrix $\mathbf{E}^*$ vary only with respect to n_j (number of j type sensors), $\forall\ j = 1, 2, \ldots, m$. We rewrite all the entries of the matrix $\mathbf{E}^*$ to explicitly show the previously mentioned dependencies given next, namely

$$\mathbf{E}^*_{i,j} = f^j_{in_j}\left(\vec{\mathbf{Z}}^*_{n_j}\right) = f_{ij}\left(n_j\right).$$

Therefore,

$$\mathbf{E}^* = \begin{bmatrix} f_{11}(n_1) & f_{12}(n_2) & .\ .\ . & f_{1m}(n_m) \\ f_{21}(n_1) & f_{22}(n_2) & .\ .\ . & f_{2m}(n_m) \\ & & & \\ . & . & .\ .\ . & . \\ . & . & .\ .\ . & . \\ f_{l1}(n_1) & f_{l2}(n_2) & .\ .\ . & f_{lm}(n_m) \end{bmatrix}_{l\times m}. \tag{10}$$

The optimal performance vector $\vec{\mathbf{P}}^*$ corresponding to the sensor combination $\vec{\mathbf{n}}$ can be obtained by substituting $\vec{\mathbf{Z}}^*_{\vec{\mathbf{n}}}$ in Eq. (7) (linear fusion model).

$$P^*_i = \sum_{j=1}^{m} f^j_{in_j}\left(\vec{\mathbf{Z}}^*_{n_j}\right) = \sum_{j=1}^{m} f_{ij}\left(n_j\right)$$

$$\forall\ i = 1, 2, \ldots, l$$

$$\left(\textit{where } P^*_i \textit{ is the ith component of } \vec{\mathbf{P}}^*\right)$$

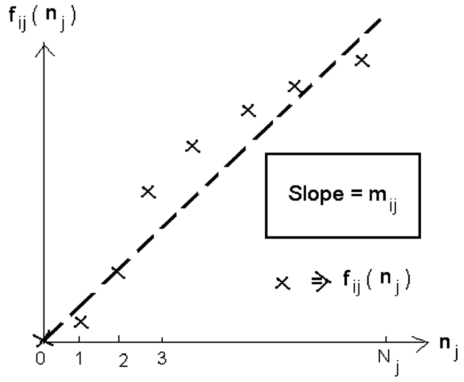


Fig. 4. Linear approximation of the data.

By using the matrix notation,

$$\vec{\mathbf{P}}^{*} = \mathbf{E}^{*} \times \begin{bmatrix} 1 \\ 1 \\ . \\ . \\ . \\ 1 \end{bmatrix}_{m \times 1} . \tag{11}$$

For any given input sensor combination $\vec{\mathbf{n}}$, we can determine the optimal performance vector $\vec{\mathbf{P}}^{*}$ using Eq. (11). Additionally, the transformation from $\vec{\mathbf{n}}$ to $\vec{\mathbf{P}}^{*}$ is nonlinear because of the nonlinear dependence of the matrix $\mathbf{E}^{*}$ on $\vec{\mathbf{n}}$ (from Eq. (10)). Therefore we can make the transformation from $\vec{\mathbf{n}}$ to $\vec{\mathbf{P}}^{*}$ linear by approximating the dependence of $\mathbf{E}^{*}$ on $\vec{\mathbf{n}}$ using a linear model.

The ith row and jth column entry $f_{ij}\left(n_j\right)$ of the matrix $\mathbf{E}^{*}$ is a function of the discrete integer (positive) variable n_j as represented in Figure 4. Let the maximum number of j type sensors that we can deploy be $N_j, \forall\, j = 1, 2, \ldots, m$. We approximate this function $f_{ij}\left(n_j\right)$ with a straight line, passing though origin and having slope m_{ij}, as shown in Figure 4. This is because the performance must be zero when n_j equals to zero. We determined the slope m_{ij} using the least squares technique and its value is

$$m_{ij} = \frac{\sum_{k=1}^{N_j} k \times f_{ij}(k)}{\sum_{k=1}^{N_j} k \times k}, \ \forall\, i = 1, 2, \ldots, l, \quad \forall\, j = 1, 2, \ldots, m. \tag{12}$$

Therefore, $f_{ij}\left(n_j\right) \approx m_{ij} \times n_j$. Note that in order to compute m_{ij} for all $i \in 1, 2, \ldots, l$ and a particular j, we must know the values of $\vec{\mathbf{Z}}^{*}_{n_j}$ for all $n_j \in 1, 2, \ldots, N_j$.

By substituting the preceding approximate value of $f_{ij}\left(n_j\right)$ in Eqs. (10) and (11), we have

$$\vec{\mathbf{P}}^{*} = \begin{bmatrix} m_{11} \times n_1 & m_{12} \times n_2 & . \ . \ . & m_{1m} \times n_m \\ m_{21} \times n_1 & m_{22} \times n_2 & . \ . \ . & m_{2m} \times n_m \\ . & . & . \ . \ . & . \\ . & . & . \ . \ . & . \\ m_{l1} \times n_1 & m_{l2} \times n_2 & . \ . \ . & m_{lm} \times n_m \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ . \\ . \\ . \\ 1 \end{bmatrix}_{m \times 1} .$$

The preceding equation can be written as

$$\vec{\mathbf{P}}^* = \begin{bmatrix} m_{11} & m_{12} & . & . & . & m_{1m} \\ m_{21} & m_{22} & . & . & . & m_{2m} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ m_{l1} & m_{l2} & . & . & . & m_{lm} \end{bmatrix} \times \begin{bmatrix} n_1 \\ n_2 \\ . \\ . \\ . \\ n_m \end{bmatrix}.$$

Let us define the matrix **A** as

$$\mathbf{A} = \begin{bmatrix} m_{11} & m_{12} & . & . & . & m_{1m} \\ m_{21} & m_{22} & . & . & . & m_{2m} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ m_{l1} & m_{l2} & . & . & . & m_{lm} \end{bmatrix}.$$

Therefore the optimal performance vector corresponding to the sensor combination $\vec{\mathbf{n}}$ is given by

$$\vec{\mathbf{P}}^* = \mathbf{A} \times \vec{\mathbf{n}}. \tag{13}$$

3.3 Integer Programming Problem

While building a surveillance system, we have to choose the optimal combination of sensors and their placement in order to minimize the cost. But we do not know a priori such information. In this section, we formulate the optimal sensor selection problem by incorporating the linear model (Eq. (13)) for computing the optimal performance vector to ensure the optimal placement for the sensors. Therefore, the performance constraints can be written as follows from Eq. (13).

$$\mathbf{A} \times \vec{\mathbf{n}} \geq \vec{\mathbf{b}}$$

Hence the problem considered in the beginning of Section 3 can be stated as follows.

Problem definition. Minimize the cost $\vec{\boldsymbol{C}}.\vec{\boldsymbol{n}}$ *with respect to* $\vec{\boldsymbol{n}}$ *such that* $\boldsymbol{A} \times \vec{\boldsymbol{n}} \geq \vec{\boldsymbol{b}}$ *and* $0 \leq n_i \leq N_i$ *and* $n_i \in$ *integer*, $\forall i = 1, 2, \ldots, m$.

The previous problem is in the form of an Integer Linear Programming problem (ILP) (or) Integer Programming (IP) problem [Schrijver 1986]. An optimal solution for this problem can be found using the following steps. (1) Solve the LP relaxation of the problem (2) Find the feasible solution around the solution found in step1 which minimizes the cost. One of the heuristic approaches to handle larger instances of the problem is to divide the surveillance region into independent (nonoverlapping or with some small overlap) regions and then independently handle each region. But the solution obtained with this approach may not be optimal.

The following algorithm summarizes the steps involved in finding the optimal combination of sensors and their placement in a surveyed region.

3.3.1 *Advantages of the Proposed Technique.* In this section, we show the advantages of our technique in terms of computation and modeling requirements by comparing it with a simple brute-force approach.

—Modeling requirements:
In case of brute-force approach, the designer has to model the effect of any particular sensor combination on the performance of each subtask. By contrast, in the proposed technique, the designer has to model the effect of each sensor type on the performance of each subtask.

—Computation:

Algorithm.

step1: Find the effect of jth type of sensor on the ith subtask ($\forall i = 1, 2, \ldots, l$ *and* $\forall j = 1, 2, \ldots, m$). This model should take n_j and $\vec{\mathbf{Z}}_{n_j}$ as input and give $f^{j}_{in_j}(\vec{\mathbf{Z}}_{n_j})$ as output.

step2: Determine the optimal placement information ($\vec{\mathbf{Z}}^{*}_{n_j}$) of the n_j number of j type sensors by maximizing the measure $\mathbf{T}_j(\vec{\mathbf{Z}}_{n_j})$ (see Eq. (9)), $\forall j = 1, 2, \ldots, m$ and by varying n_j from 1 to N_j. (Hence the total number of optimization problems in this step is $\sum_{j=1}^{m} N_j$.)

step3: Determine the elements of the matrix $\mathbf{A}$ using Eq. (12).

step4: Find the optimal sensor combination by solving the integer programming problem stated in the Section 3.3 and then obtain the corresponding optimal placement from step2.

Brute-force approach:

(1) Total number of optimal placement problems : $\prod_{j=1}^{m} N_j$ (the same as the total number of possible sensor combinations).
(2) Dimension of the search space for any particular problem: cardinality of $\vec{\mathbf{Z}}_{\vec{\mathbf{n}}}$.
(3) The designer can pick straightaway the optimal combination and placement of sensors from the solutions of the preceding optimal placement problems.

Proposed technique :

(1) Total number of optimal placement problems: $\sum_{j=1}^{m} N_j$.
(2) Dimension of the search space for any particular problem: cardinality of $\vec{\mathbf{Z}}_{n_i}$. (n_i number of ith type of sensors)
(3) Solve the previous ILP problem to obtain the optimal combination of sensors, and the corresponding placement can be obtained from the solutions of the optimal placement problems.

3.3.2 *Special Case.* Suppose that there are l sensor types and l subtasks and no two sensor types can accomplish the same subtask. In this case, by properly numbering the sensor types, the matrix $\mathbf{E}^{*}$ (Eq. (10)) becomes diagonal. Thus the optimal performance vector can be written as follows from Eq. (11).

$$\vec{\mathbf{P}}^{*} = \begin{bmatrix} f_{11}(n_1) \\ f_{22}(n_2) \\ \cdot \\ \cdot \\ \cdot \\ f_{ll}(n_l) \end{bmatrix}$$

Hence the performance constraints become $f_{ii}(n_i) \geq r_i, \forall i = 1, 2, \ldots, l$. Therefore, the optimal combination of sensors can be found by solving each one of the inequalities independently for its minimum integer argument. Specifically, the ith component of the optimal sensor combination is given by

$$\underset{min}{arg}\ f_{ii}(n_i) \geq r_i. \tag{14}$$

4. SURVEILLANCE SYSTEM DESIGN

In this section, we consider the design of a specific surveillance system consisting of two types of sensors which are PTZ (Pan-Tilt-Zoom) infrared cameras and active motion sensors ($m = 2$). The surveillance

Table I. Elements of the Performance Matrix

	cameras	motion sensors
image capture	$f^1_{1n_1}\left(\vec{\mathbf{Z}}_{n_1}\right)$	0
localization	0	$f^2_{2n_2}\left(\vec{\mathbf{Z}}_{n_2}\right)$

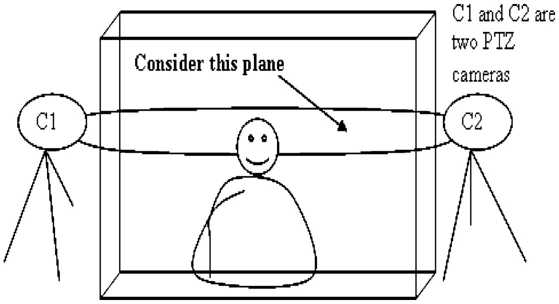


Fig. 5. A typical surveillance setup: C1 and C2 represent PTZ cameras.

task is to capture the frontal face of an intruder in a surveyed region. This task has two subtasks: intruder localization and image capture ($l = 2$). We know that the localization subtask can be performed by both types of sensors while the image capture subtask can be performed by the camera alone. This can be simplified to the special case ($l = 2$) described in Section 3.3.2 by neglecting the effect of the cameras on the object localization subtask. The performance matrix becomes diagonal with this simplification as there is one different sensor type for each different subtask. Thus we need to determine the elements of the performance matrix as indicated in Table I.

4.1 Modeling the Effect of Cameras on the Image Capture Subtask

In this section, we determine $f^1_{1n_1}(\vec{\mathbf{Z}}_{n_1})$, that is, the performance with which n_1 number of PTZ infrared cameras arranged in a particular configuration ($\vec{\mathbf{Z}}_{n_1}$) accomplish the image capture subtask. The average probability of capturing the frontal part of a face at a particular time instant [Sivaram et al. 2006] could be used as a measure of performance ($f^1_{1n_1}(\vec{\mathbf{Z}}_{n_1})$) as we are interested in capturing the frontal face. Initially, we derive the preceding performance measure for a convex surveyed region and then extend the same to the nonconvex region.

We make the following assumptions while deriving the performance measure for n_1 cameras.

(1) The face can be captured if its centroid lies within the conical FOV (Field Of View) of cameras.

(2) If half or more than half of the frontal part of the face is captured by any one of the cameras, then due to the symmetry, the frontal face can be obtained.

The first assumption allows us to do the analysis on a plane by considering face centroid positions, while the second assumption allows us to reconstruct the frontal face for all poses (0 degree = frontal face in camera) between −90 and +90 degrees with respect to a camera axis and further to compute probability of capturing a frontal face at a particular point due to multiple cameras.

Consider a plane which is parallel to the floor and at the same vertical height as that of the cameras, as shown in Figure 5. The top view of this plane is as shown in Figure 6. Though the actual centroid of a face may not lie on the considered plane due to the variability in pose, etc., most of the time the Field Of View (FOV) of the cameras is enough to capture the face. In this case, the cameras capture a slightly

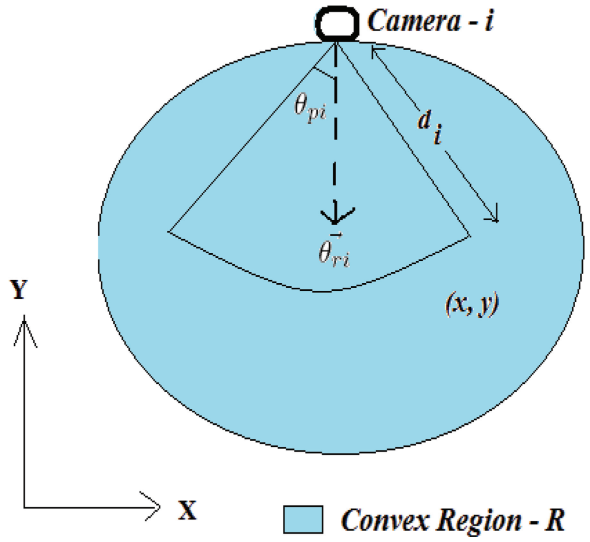


Fig. 6. Top view of the convex surveyed region (2D plane) and parameters of the ith camera.

distorted face due to the angle of projection of an object onto the camera plane. We neglect this effect and assume that the centroid of a face lies on the considered plane for analysis. Also, in practice the FOV gets affected by changes in the zoom parameter of the camera, but we neglect this consideration as of now for ease of modeling.

Consider a convex surveyed region. Let us denote its top view, a convex 2D region, as R (shaded region in Figure 6). We now derive an expression for the probability of capturing a frontal face if its centroid is at a location, say $(x, y) \in R$, as in Figure 6. This analysis will not impose any restriction on the orientation of the face. We represent the orientation of a face using a random variable which is distributed uniformly in the range $[0, 2\pi)$. This is intuitively satisfying because any orientation angle for the face is equally likely. The idea is to find a set of orientation angles of a face having centroid at $(x, y) \in R$ for which the frontal face can be captured by at least one of the n_1 cameras and then determine the probability of realizing this set. By assumption 2, if we capture half or more than half of the frontal face it implies that, due to symmetry, the total frontal face can be obtained.

The parameters associated with the ith camera ($1 \le i \le n_1$) are as follows.

—Location : (x_i, y_i) (on the boundary only)

—Zoom : d_i

—Reference direction :$\theta_{ri} = arg(\vec{\theta_{ri}})$, $0 \le \theta_{ri} < 2\pi$, where $arg(\vec{\theta_{ri}})$ is the angle that is measured counterclockwise from the positive x-axis to the vector $\vec{\theta_{ri}}$.

—Maximum pan angle : θ_{pi} *radians*, (> 0)

(hence $\vec{\mathbf{Z}}_{n_1} = \left[x_1\ \ y_1\ \ \theta_{r1}\ \ .\ \ .\ \ x_{n_1}\ \ y_{n_1}\ \ \theta_{rn_1}\right]^{\mathrm{T}}$).

The zoom parameter indicates the maximum distance that the ith camera can focus, and the maximum pan angle indicates the maximum pan allowed in either the positive or negative direction about the reference direction, as shown in Figure 6. In this analysis, it is assumed that the parameter maximum pan angle includes the effect of field of view of the camera, namely $\theta_{pi} = \theta_{pi,orig} + (FOV\ in\ radians)/2$, where $\theta_{pi,orig}$ is the actual maximum pan angle of the camera.

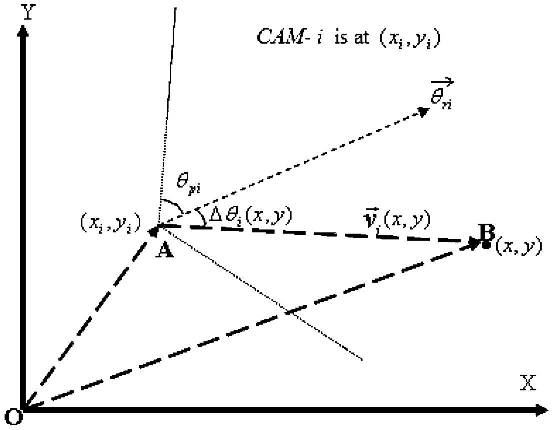


Fig. 7. Directions of various vectors. A: position of the ith camera; B: face centroid position (x, y); O: origin of the 2D plane; θ_{pi} & $\vec{\theta_{ri}}$: maximum pan angle and reference direction of the ith camera; $\vec{V_i}(x, y)$: vector from the ith camera to the point (x, y), and $\Delta\theta_i(x, y)$: angle between $\vec{V_i}(x, y)$ and $\vec{\theta_{ri}}$.

We define the *characteristic function* $I_i(x, y)$ for the ith camera for all points $(x, y) \in R$ as

$$I_i(x, y) = \begin{cases} 1, & \text{if } i\text{th camera can focus on } (x, y) \\ 0, & \text{otherwise} \end{cases}$$

and it can be written as $I_i(x, y) = I_{i1}(x, y) \times I_{i2}(x, y)$, where $I_{i1}(x, y) = U(d_i^2 - [(x - x_i)^2 + (y - y_i)^2])$ and $I_{i2}(x, y) = U(\theta_{pi} - \Delta\theta_i(x, y))$, $U(.)$ is the unit step function. $\Delta\theta_i(x, y)$ is the angle difference between the reference direction vector $(\vec{\theta_{ri}})$ of the ith camera and the vector $\vec{V_i}(x, y)$ (from ith camera to the point (x, y)) as shown in Figure 7.

The characteristic function $I_i(x, y)$ essentially describes whether the object's image can be captured by camera i at point (x, y) or not. The function $I_{i1}(x, y)$ indicates the distance constraint imposed by the zoom of the camera and $I_{i2}(x, y)$ indicates the pan angle constraint. The vector from ith camera to the object centroid at (x, y) is represented using $\vec{V_i}(x, y)$ and can be found from the triangle OAB (refer to Figure 7).

$$(x_i, y_i) + \vec{V_i}(x, y) = (x, y) \Rightarrow \vec{V_i}(x, y) = (x - x_i, y - y_i).$$

Let us define $\theta_i(x, y) = arg(\vec{V_i}(x, y))$, $0 \leq \theta_i(x, y) < 2\pi$, as indicated in Figure 8 for some (x, y). As stated earlier, the orientation of a face is represented using a random variable θ which is distributed uniformly in the range $[0, 2\pi)$. According to assumption 2, the ith camera can capture the frontal face having centroid at (x, y) whenever the orientation angle of a face $\theta \in S_i(x, y)$. Figure 8 shows a specific case. $S_i(x, y)$ is expressed as

$$S_i(x, y) = \{\theta_i : \theta_i(x, y) + \pi/2 \leq \theta_i < \theta_i(x, y) + 3\pi/2\} \bmod 2\pi,$$

which represents the set of all orientation angles for a face having centroid at (x, y) for which ith camera can capture the frontal face. If the object is such that the frontal part of it can be obtained from any of its captured images (independent of its orientation) then the analysis becomes simple and we have merely to maximize the coverage area. This is not true for objects like human and animal faces, as shown in Figure 8. Therefore, we need to do the following analysis. Let us define $P_i(x, y) = Prob\ \{\theta \in S_i(x, y)\}$. Hence the probability of capturing the frontal face having centroid at (x, y) using the ith camera is given

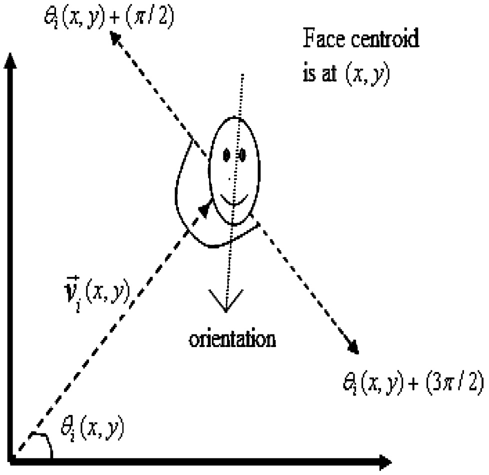


Fig. 8. Set of orientations for which ith camera can capture the frontal face, $\vec{V}_i(x, y)$: vector from the ith camera to the face centroid at (x, y).

by $P_i(x, y) \times I_i(x, y)$. Let $P^{n_1}(x, y)$ denote the probability of capturing a frontal face having centroid at (x, y) and with n_1 number of cameras arranged in any fixed configuration.

4.1.1 *Single Camera Case* ($n_1 = 1$). Recall that $I_1(x, y)$ indicates whether camera 1 can focus on (x, y) or not. Hence, in this case, $P^1(x, y) = I_1(x, y) \times P_1(x, y)$.

4.1.2 *Dual Camera Case* ($n_1 = 2$). We know that $P(X \bigcup Y) = P(X) + P(Y) - P(X \bigcap Y)$, where X and Y are any two events.

Case 1. When both the cameras are able to focus on (x, y)

$$P^2(x, y) = Prob\left\{\theta \in \left[S_1(x, y) \bigcup S_2(x, y)\right]\right\}$$

$$= Prob\left\{\theta \in S_1(x, y)\right\} + Prob\left\{\theta \in S_2(x, y)\right\}$$

$$-Prob\left\{\theta \in \left[S_1(x, y) \bigcap S_2(x, y)\right]\right\}$$

$$= P_1(x, y) + P_2(x, y) - P_{12}(x, y)$$

$$\left(P_{12}(x, y) = Prob\left\{\theta \in \left[S_1(x, y) \bigcap S_2(x, y)\right]\right\}\right)$$

where $P_{12}(x, y)$ denotes the probability of capturing the frontal face having centroid at (x, y) by both the cameras.

Case 2. When only one of the cameras is able to focus on (x, y)

$$P^2(x, y) = Prob\left\{\theta \in S_i(x, y)\right\} = P_i(x, y)$$

where only camera i can focus on (x, y), $i = 1$ or 2.

Case 3. Either of the cameras can't focus on (x, y)

$$P^2(x, y) = 0.$$

All three previous cases can be compactly written as

$$P^2(x, y) = I_1(x, y) \times P_1(x, y) + I_2(x, y) \times P_2(x, y)$$

$$-I_1(x, y) \times I_2(x, y) \times P_{12}(x, y).$$

Since the random variable θ is uniformly distributed in the range $[0, 2\pi)$, the preceding expression reduces to

$$\begin{aligned} P^2(x, y) = (1/2) \times [I_1(x, y) + I_2(x, y)] \\ -I_1(x, y) \times I_2(x, y) \times P_{12}(x, y). \end{aligned} \tag{15}$$

The point (x, y) can be anywhere on the plane and belongs to the convex set R and the characteristic function of a particular camera describes whether that camera can focus on this point. The average probability of capturing the frontal face at any particular time instant (performance measure) can be found if we know the probability density function $f(x, y)$ for a face centroid position over the convex region R. Thus, the average probability in this case represents the performance measure $f^1_{1n_1}(\vec{\mathbf{Z}}_{n_1})$ for $n_1 = 2$, as discussed earlier.

$$f^1_{12}\left(\vec{\mathbf{Z}}_2\right) = \iint_R P^2(x, y)\ f(x, y)\, dx\, dy$$

Let the area of the convex region R be A_R and further assume that the position (x, y) is a random variable with a uniform density (in this case, $f(x, y) = \frac{1}{A_R}$). Uniform density for the position means the face can be found with an equal probability in any region of fixed total area.

$$f^1_{12}\left(\vec{\mathbf{Z}}_2\right) = \frac{1}{A_R} \iint_R P^2(x, y)\, dx\, dy$$

Substituting for $P^2(x, y)$ from Eq. (15), we have

$$= \frac{1}{A_R} \iint_R (1/2) \times [I_1(x, y) + I_2(x, y)]\, dx\, dy$$

$$-\frac{1}{A_R} \iint_R I_1(x, y) \times I_2(x, y) \times P_{12}(x, y)\, dx\, dy$$

$$\begin{aligned} f^1_{12}\left(\vec{\mathbf{Z}}_2\right) = \frac{0.5}{A_R} \left\{ Volume\ under I_1(x, y) + Volume\ under I_2(x, y) \right\} \\ -\frac{1}{A_R} \iint_A P_{12}(x, y)\, dx\, dy \end{aligned} \tag{16}$$

where A : area where both the cameras can focus (i.e., set of all (x, y) under Case 1).

4.1.3 *More Than Two Cameras.* In this section we extend the performance metric to the n_1 camera case. As mentioned earlier, $P^{n_1}(x, y)$ denotes the probability of capturing the frontal face having centroid at (x, y) and with n_1 number of PTZ cameras in a fixed layout ($\vec{\mathbf{Z}}_{cam}$). If (x, y) is such that all cameras are able to focus on this point then expression for $P^{n_1}(x, y)$ is given by

$$P^{n_1}(x, y) = Prob \left\{ \theta \in \left[S_1(x, y) \bigcup S_2(x, y) \ldots \bigcup S_{n_1}(x, y)] \right] \right\}.$$

Since we know how to deal with two cameras, initially we start with two cameras. After determining the effect of first two cameras, we add one more camera to find its effect. Note that the order in which we add cameras to the existing configuration has no effect on the final performance metric, as the union operator is associative. This process of adding a new camera to the existing system is repeated until we include all the cameras. The algorithmic approach is described next.

Once we know $\{P^{n_1}(x, y),\ \forall(x, y) \in R\}$, the average probability $f^1_{1n_1}(\vec{\mathbf{Z}}_{n_1})$ can be found by integrating and averaging over the entire convex region, as discussed in Section 4.1.2. The optimal camera placement $\vec{\mathbf{Z}}^*_{n_1}$ is obtained by maximizing $f^1_{1n_1}(\vec{\mathbf{Z}}_{n_1})$ (brute-force search) with respect to the camera placement $\vec{\mathbf{Z}}_{n_1}$. Let us denote the optimal performance of n_1 number of cameras as $f_{11}(n_1)$. Therefore $f_{11}(n_1) = f^1_{1n_1}(\vec{\mathbf{Z}}^*_{n_1})$.

4.1.4 *Extension to Nonconvex Region.* To account for the nonconvexity of the region, we use the modified characteristic function $I^c_i(x, y)$ instead of $I_i(x, y)$ in the algorithm of Section 4.1.3. The modified characteristic function for the ith camera is given by $I^c_i(x, y) = I_i(x, y) \times C_i(x, y)$, where $C_i(x, y)$ indicates the visibility function of the ith camera and is defined for all points $(x, y) \in R$ as

$$C_i(x, y) = \begin{cases} 1, & \text{if } \vec{V}_i(x, y) \text{ does not cross boundaries of } R, \\ 0, & \text{otherwise} \end{cases}$$

where $\vec{V}_i(x, y)$ is a vector from the ith camera to the point (x, y) as shown in Figure 7.

Algorithm. To determine $P^{n_1}(x, y)$

```
Inputs:
    Sets: $S_i(x, y), i = 1, 2, ...n_1$
    Probabilities: $P_i(x, y), i = 1, 2, ...n_1$
    Charcteristic functions: $I_i(x, y), i = 1, 2, ...n_1$
Initialize:
    $A \leftarrow S_1(x, y)$ and $B \leftarrow S_2(x, y)$
    $p_1 \leftarrow P_1(x, y)$ and $p_2 \leftarrow P_2(x, y)$
    $i_1 \leftarrow I_1(x, y)$ and $i_2 \leftarrow I_2(x, y)$
    for $j = 3$ to $n_1$
        Compute:
            $p = i_1 \times p_1 + i_2 \times p_2 - i_1 \times i_2 \times p_{12}$
            where $p_{12} = Prob\{\theta \in A \bigcup B\}$
        Update sets:
            if $i_1 = 1$ and $i_2 = 1$
            then $A \leftarrow A \bigcup B$
            if $i_1 = 0$ and $i_2 = 1$
            then $A \leftarrow B$
            if $i_1 = 0$ and $i_2 = 0$
            then $A \leftarrow \phi$
        Update probabilities:
            $p_1 \leftarrow p$ and $p_2 \leftarrow P_j(x, y)$
        Update characteristic functions:
            $i_1 \leftarrow \max(i_1, i_2)$ and $i_2 \leftarrow I_j(x, y)$
    end for
    $P^{n_1}(x, y) = i_1 \times p_1 + i_2 \times p_2 - i_1 \times i_2 \times p_{12}$
Output $P^{n_1}(x, y)$
```

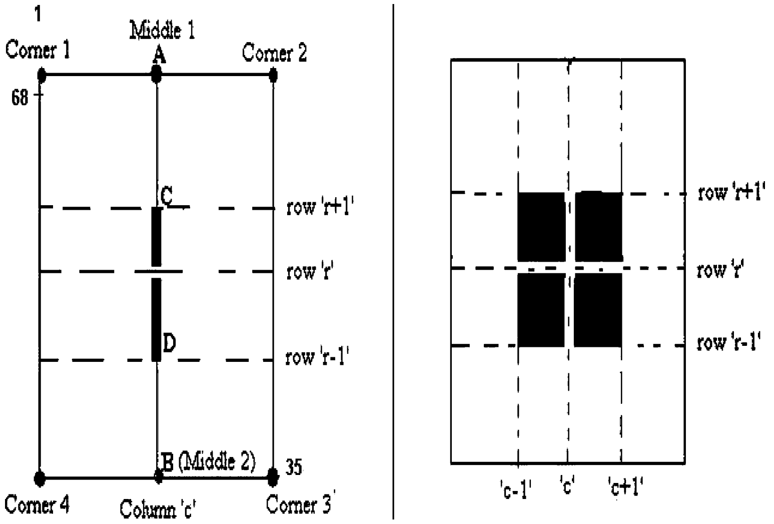


Fig. 9. Uncertainty introduced by a 2D motion sensor grid in performing a localization subtask.

4.2 Modeling the Effect of Motion Sensors on the Localization Subtask

One kind of active motion sensor consists of a transmitter and a receiver pair and has two states, namely where the beam is continuous and the beam is discontinuous. In this section, we determine the performance $f^2_{2n_2}(\vec{\mathbf{Z}}_{n_2})$ with which n_2 number of motion sensors arranged in the form of a 2D grid ($\vec{\mathbf{Z}}_{n_2}$) accomplish the localization subtask.

4.2.1 *Localization by Motion Sensor Grid.* Consider a motion sensor grid formed by arranging the motion sensors into rows and columns as shown in Figure 9. Let us call the motion sensors along the row as "row motion sensors" and along the column as "column motion sensors". The grid points correspond to the locations where beams from the row motion sensors and column motion sensors meet. In localizing an object (or intruder) using the sensor grid, we assume that the object cannot cross the beam of any motion sensor in less than Δt seconds, where Δt is the polling time for the sensor grid. Using the past localization information and the current sensor grid status, the new localization information can be found. But in most of the cases, there is some uncertainty associated with the localization information obtained from the sensor grid. We explain different cases next.

Case 1. At time instant t, the row motion sensor with number r and the column motion sensor with number c are discontinuous. In this case there is no uncertainty in localizing an object and it is exactly there on the grid/intersection point (r, c). This is because (see Figure 9, right) the object can be anywhere obstructing the row motion sensor r and column motion sensor c, and this is possible only if the object is at the intersection point (r, c).

Case 2. At time instant t, only one of the motion sensors is discontinuous (row or column). Let the current discontinuous motion sensor be a column motion sensor with number c (refer Figure 9, left). Let the previous latest discontinuous row motion sensor number be r. As per the assumption in Section 4.2.1, the object cannot cross either row $r - 1$ or $r + 1$ and yet obstruct column motion sensor c. Because of the assumption, the uncertainty in localizing an object in this case is reduced from the line segment AB to the line segment CD (thick) excluding the intersection/grid point, as shown in Figure 9 (left).

Case 3. At time t, no motion sensor is discontinuous. Let the latest previous discontinuous row and column motion sensors be r and c, respectively. In this case an object cannot cross the row motion sensors $r-1$ and $r+1$ and similarly the column motion sensors $c-1$ and $c+1$. Hence the uncertainty region in this case is the dark region, as shown in Figure 9, right (note that the row motion sensor r and column motion sensor c are continuous).

So we can think of the uncertainty region associated with each grid point as a rectangle formed by the adjacent motion sensors' (both rows and columns) beam. The shape of the uncertainty region depends on the placement of the motion sensors. The performance of the localization subtask increases as the maximum distance between the grid point and the points in the corresponding uncertainty region decreases. Let us represent this maximum distance corresponding to the ith grid point as D_i, $\forall i = 1,\ 2, \ldots, N_g$ (where N_g represents the total number of grid points). Note that the performance of the localization subtask increases as the average of the maximum distances associated with the grid points decreases. Hence we can write the performance measure as

$$f_{2n_2}^{2}\left(\vec{\mathbf{Z}}_{n_2}\right) = 1 - \frac{\frac{1}{N_g} \times \sum_{k=1}^{N_g} D_k}{D_{max}} \tag{17}$$

(where D_{max} represents the maximum distance between any two points in the surveyed region)

4.2.2 *Optimal Motion Sensor Grid.* The optimal placement of the motion sensor grid is obtained by maximizing the performance measure $f_{2n_2}^{2}(\vec{\mathbf{Z}}_{n_2})$ in Eq. (17). Let us denote the corresponding optimal performance with $f_{22}\,(n_2)$. Therefore,

$$f_{22}\,(n_2) = f_{2n_2}^{2}\left(\vec{\mathbf{Z}}_{n_2}^{*}\right) = 1 - \frac{\min\left(\frac{1}{N_g} \times \sum_{k=1}^{N_g} D_k\right)}{D_{max}}. \tag{18}$$

5. RESULTS

We present in this section the simulation results describing the optimal selection of sensors and their placement. We also show the experimental results for tracking and capturing the face of an intruder.

5.1 Optimal Combination of Sensors and Their Placement

The design problem considered in the beginning of Section 4 can be reduced to the form of the special case described in Section 3.3.2 by neglecting the effect of the cameras on the intruder localization subtask. Hence the optimal performance becomes

$$\vec{\mathbf{P}}^{*} = \begin{bmatrix} f_{11}\,(n_1) \\ f_{22}\,(n_2) \end{bmatrix}.$$

We considered a rectangular surveillance region of size $6m \times 2.5m$ and the required performance vector has been chosen to be $\vec{\mathbf{b}} = \begin{bmatrix} 0.78 & 0.81 \end{bmatrix}^{\mathrm{T}}$.

5.1.1 *Optimal Selection and Placement of Cameras.* The parameters maximum pan angle (θ_{pi}) and zoom (d_i) of the PTZ camera are chosen to be 45 degrees and 5.5m, respectively. To reduce the dimension of the search space, the reference direction (suboptimal θ_{ri}) for each camera is chosen such that maximum volume is included under the corresponding characteristic function. Figure 11 shows the performance measure of the image capture subtask with two cameras ($f_{12}^{1}(\vec{\mathbf{Z}}_2)$) as a function of cameras' position along the perimeter (total 68 equally spaced points are considered along the perimeter and are numbered starting from corner 1 as shown in Figure 9, left). Note that this function is two-way symmetric, as camera positions can be swapped without changing the performance of the image capture

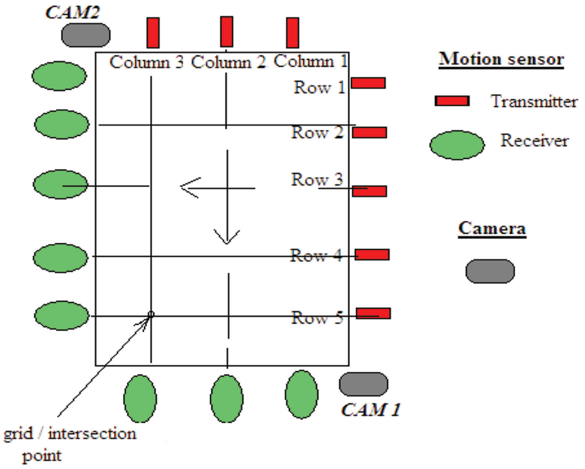


Fig. 10. Optimal placement of cameras and motion sensors in a rectangular surveyed region (convex).

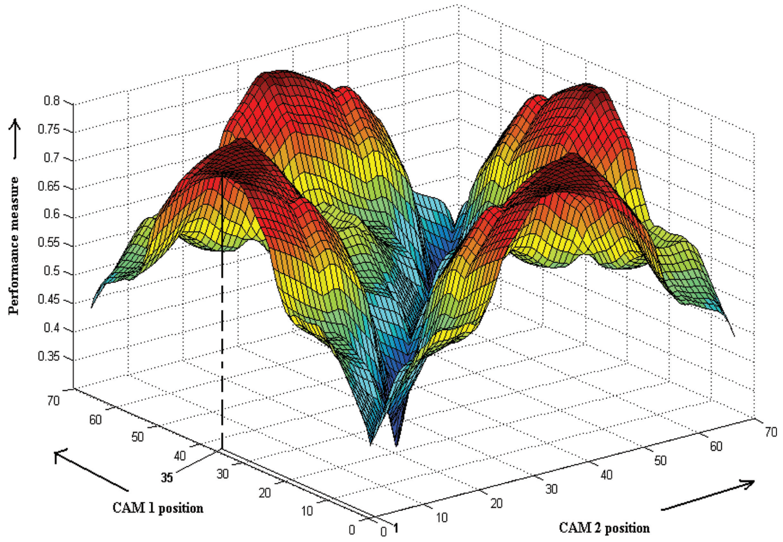


Fig. 11. Performance vs. camera placement for the rectangular surveyed region.

subtask and also due to the symmetric surveyed region. The performance measure is maximum when the cameras are placed in diagonally opposite corners and thus it corresponds to optimal performance. We found this optimal performance with two cameras ($f_{11}(2)$) to be 0.782. When both the cameras are placed at the same point then the performance measure is same as that of the single camera placed at this point. The performance measure values along the line $CAM1\ position - CAM2\ position = 0$ (in Figure 11) represent the single camera case. The maximum value ($f_{11}(1)$) along the line is 0.43.

From Eq. (14), the optimal number of cameras (n_1^*) is given by

$$n_1^* = \mathop{}^{arg}_{min} \left(f_{11}(n_1) \geq 0.78 \right).$$

Therefore, the optimal number of cameras n_1^* is two and their placement is in diagonally opposite corners, as shown in Figure 10.

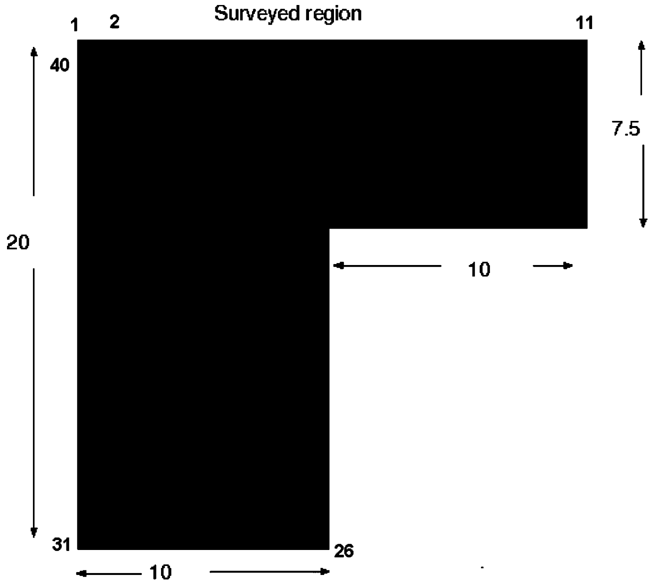


Fig. 12. Nonconvex surveyed region.

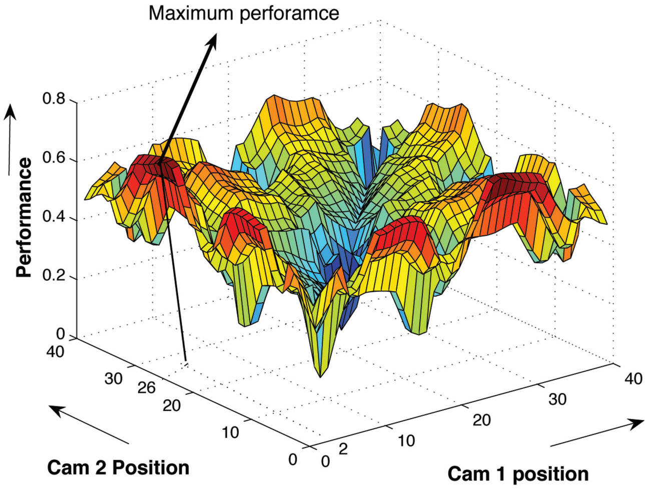
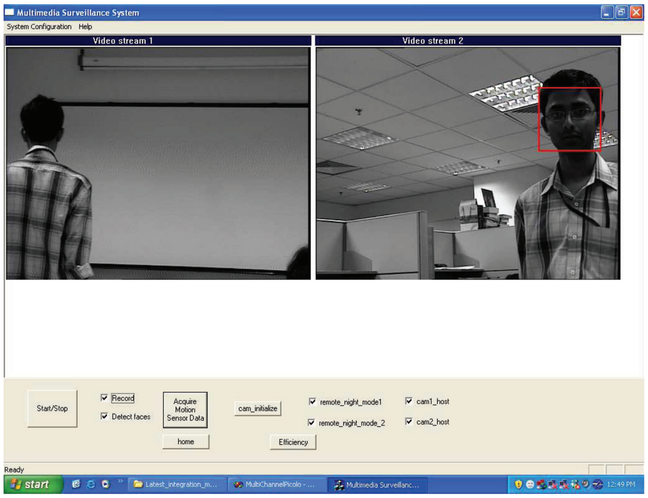


Fig. 13. Performance as a function of camera placement for the nonconvex surveyed region.

5.1.2 *Nonconvex Region.* Simulation results for the two camera placement problem in a nonconvex region are discussed in this section. The perimeter of a nonconvex region shown in Figure 12 is divided into 40 equal parts. The maximum pan angle (θ_{pi}) and zoom are chosen to be 47 degrees and 20m, respectively, for both the cameras. The reference direction for each camera is chosen such that the maximum volume is included under the corresponding characteristic function. Performance of the image capture subtask $f^1_{12}(\vec{\mathbf{Z}}_2)$ as a function of the cameras' position is shown in Figure 13. The combination of 2 and 26 gave the maximum performance of 0.72.

Table II. Optimal Localization Performance

n_2	optimal placement $n_{21} \times n_{22}$	$f_{mot}(n_2)$
1	1×0	0.0
2	2×0	0.637
3	3×0	0.699
4	4×0	0.733
5	5×0	0.754
6	4×2	0.775
7	5×2	0.799
8	5×3	0.818



☑ Record
☑ Detect faces
☑ remote_night_mode1
☑ remote_night_mode_2
☑ cam1_host
☑ cam2_host

Fig. 14. Testbed for our experiments.

5.1.3 *Optimal Selection and Placement of Motion Sensors.* We determined the optimal motion sensor grid for a given number of motion sensors by using Eq. (18). Table II lists the optimal performance of the localization subtask as a function of the number of motion sensors in a rectangular surveillance region of size 6m × 2.5m. The optimal placement $n_{21} \times n_{22}$ (refer to Table II) denotes n_{21} number of sensors along the length (6m) and n_{22} number along the width(2.5m) respectively (sensors are equally spaced). The optimal number of motion sensors (n_2^*) can be determined using the following (from Eq. (14)) equation.

$$n_2^* = \substack{arg \\ min} \left(f_{22}(n_2) \geq 0.81 \right).$$

Therefore, from Table II, the optimal number of motion sensors n_2^* is 8 and the corresponding optimal placement is 5 × 3 grid, as shown in Figure 10.

5.2 Tracking Results

In this section, we present face tracking results of the surveillance system built using the optimal combination of sensors found in Section 5.1. Figure 14 shows our testbed for the tracking experiment. To track and further capture the frontal face of an intruder, cameras parameters like pan, tilt, and zoom need to be adjusted based on the localization information obtained from motion sensors. Such an

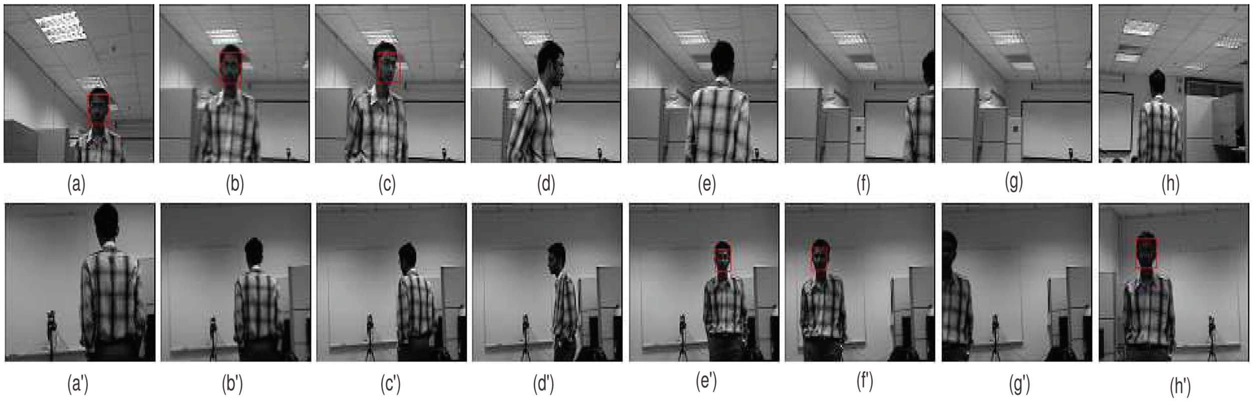


Fig. 15. Tracking results: (a)–(h) Camera 1 images; (a')–(h') Camera 2 images.

Table III. Effect of Camera Placement

Camera placement	Face capturing ratio (%)
Cam1-Middle1, Cam2-Middle2	42
Cam1-Middle1, Cam2-Corner3	51
Cam1-Corner1, Cam2-Corner3	69
Cam1-Corner1, Cam2-Corner4	38
Cam1-Corner1, Cam2-Corner1	27
Cam1-Middle1, Cam2-Middle1	17

interaction strategy between sensors allows the system to react and track an intruder efficiently. For example, consider Figure 15, where a few images captured by both the cameras of a surveillance system for a particular camera placement are shown (images in any column of Figure 15 (i.e., (a),(a'), etc.) are captured at a particular time instant). Since localization is done by the motion sensor grid, cameras are able to react and track an intruder even if no face is being detected in the captured frames. This can be observed from images (g), (g'), (h), and (h') of Figure 15. Surveillance systems consisting of only cameras cannot track in this case.

Table III summarizes the effect of camera placements on the "successful face capturing" ratio. We define the "successful face capturing" ratio of the number of frames captured with frontal facial data to the total number of frames captured for each camera. In our experiments, we considered a fixed motion trajectory that passes through all the grid points and obtained 100 frame images per camera for each camera placement. A total of 6 points were chosen (i.e.,Corner1–4 and Middle1–2) along the perimeter for the camera position, as shown in the left image of Figure 9. The experimental results show that a maximum accuracy of 69% is obtained when cameras are placed in diagonally opposite corners. Note that we found the same placement by maximizing the performance measure of the image capture subtask ($f^1_{12}(\vec{\mathbf{Z}}_2)$) in Section 5.1.1. Thus, we verified experimentally for the image capture subtask that optimal placement based on the performance measure maximizes the performance of the subtask.

6. CONCLUSIONS

In this article, we addressed the problem of how to determine the optimal combination of sensors along with their placement in order to meet the given performance requirements at minimal cost for multimedia surveillance systems. First, we approximated, optimal performance vector corresponding to a sensor combination, with the output of a linear model with sensor combination being an input.

Then we used this linear model in the formulation of the optimal sensor selection problem (to ensure performance corresponding to the optimal placement) and showed that it can be reduced to the form of the Integer Linear Programming problem (ILP). To demonstrate how our technique can be used for designing a surveillance system, we considered a face capturing system consisting of PTZ cameras and motion sensors. We also build a real-time surveillance system using the optimal combination of cameras and motion sensors and their placement information obtained after following the proposed design technique. Experimental results have confirmed the effectiveness of the proposed technique. Future work includes designing of surveillance systems that use different sensors like acoustic sensors, pressure sensors, infrared sensors, etc., and perform different subtasks.

REFERENCES

Bodor, R., Drenner, A., Schrater, P., and Papanikolopoulos, N. 2007. Optimal camera placement for automated surveillance tasks. *J. Intell. Robotic Syst. 50*, 3, 257–295.

Chen, S. and Li, Y. 2004. Automatic sensor placement for model-based robot vision. *IEEE Trans. Syst. Man Cybernetics 34*, 1, 393–408.

Chen, X. and Davis, J. 2000. Camera placement considering occlusion for robust motion capture. Tech. rep. CS-TR-2000-07. Stanford University, Department of Computer Science.

Dhillon, S. S. and Chakrabarty, K. 2003. Sensor placement for effective coverage and surveillance in distributed sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, 1609–1614.

Erdem, U. M. and Sclaroff, S. 2004. Optimal placement of cameras in floorplans to satisfy task requirements and cost constraints. In *Proceedings of the International Workshop on Omnidirectional Vision*, 30–41.

Horster, E. and Lienhart, R. 2006. Approximating optimal visual sensor placement. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, 1257–1260.

Howard, A., Matark, M., and Sukhatme, G. 2002. Localization for mobile robot teams using maximum likelihood estimation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 434–439.

Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., and Glance, N. 2007. Cost-effective outbreak detection in networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 420–429.

Luo, R. and Kay, M. 1989. Multisensor integration and fusion in intelligent systems. *IEEE Trans. Syst. Man Cybernetics 19*, 5, 901–931.

Luo, R., Yih, C., and Su, K. 2002. Multisensor fusion and integration: Approaches, applications, and future research directions. *IEEE Sensors J. 2*, 2, 107–119.

Mittal, A. and Davis, L. 2004. Visibility analysis and sensor planning in dynamic environments. In *Proceedings of the European Conference on Computer Vision, Part 1*, 175–189.

Pahalawatta, P., Pappas, T., , and Katsaggelos, A. 2004. Optimal sensor selection for video-based target tracking in a wireless sensor network. In *Proceedings of the IEEE International Conference on Image Processing*, 3073–3076.

Schrijver, A. 1986. *Theory of Linear and Integer Programming*. John Wiley & Sons, New York.

Scott, W., Roth, G., and Rivest, J. 2003. View planning for automated three-dimensional object reconstruction and inspection. *ACM Comput. Surv. 35*, 1, 64–96.

Sivaram, G., Ramakrishnan, K., Atrey, P., Singh, V., and Kankanhalli, M. 2006. A design methodology for selection and placement of sensors in multimedia surveillance systems. In *Proceedings of the ACM International Workshop on Video Surveillance and Sensor Networks*, 149–158.

Tarabanis, K., Allen, P., and Tsai, R. 1995. A survey of sensor planning in computer vision. *IEEE Trans. Robotics Autom. 11*, 1, 86–104.

Wren, C., Erdem, U., and Azarbayejani, A. 2005. Automatic pan-tilt-zoom calibration in the presence of hybrid sensor networks. In *Proceedings of the ACM International Workshop on Video Surveillance and Sensor Networks*, 113–120.

Zierhut, H. Motion detector for space surveillance. U.S. Patent No. 4529874.

Received July 2007; revised March 2008; accepted May 2008