# Information-Driven Sensor Path Planning by Approximate Cell Decomposition

Chenghui Cai, *Member, IEEE*, and Silvia Ferrari, *Senior Member, IEEE*

*Abstract*—A methodology is developed for planning the sensing strategy of a robotic sensor deployed for the purpose of classifying multiple fixed targets located in an obstacle-populated workspace. Existing path planning techniques are not directly applicable to robots whose primary objective is to gather sensor measurements using a bounded field of view (FOV). This paper develops a novel approximate cell-decomposition method in which obstacles, targets, sensor's platform, and FOV are represented as closed and bounded subsets of an Euclidean workspace. The method constructs a connectivity graph with observation cells that is pruned and transformed into a decision tree from which an optimal sensing strategy can be computed. The effectiveness of the optimal sensing strategies obtained by this methodology is demonstrated through a mine-hunting application. Numerical experiments show that these strategies outperform shortest path, complete coverage, random, and grid search strategies, and are applicable to nonoverpass capable robots that must avoid targets as well as obstacles.

*Index Terms*—Demining, fusion, geometric sensing, information theory, robotic sensors, sensor path planning.

## I. INTRODUCTION

SENSOR planning refers to the problem of determining a strategy for gathering sensor measurements to support a sensing objective, such as target classification. When the sensors are installed on robotic platforms an important part of the problem is planning the sensor path [1]–[3]. Several approaches have been proposed for planning the path of mobile robots with on-board sensors to enable navigation and obstacle avoidance in unstructured dynamic environments, e.g., [4]–[9]. However, these methods are not directly applicable to robotic sensors whose primary goal is to support a sensing objective, rather than to navigate a dynamic environment [10]. The reason is that these methods focus on how the sensor measurements can best support the robot motion, rather than focusing on the robot motions that best support the sensing objective [10]. This paper addresses the problem of planning the path and measurements of a robotic sensor, in order to classify multiple targets distributed in an obstacle-populated workspace. This problem, known as the treasure hunt [11], arises in many applications, such as robotic mine hunting [12], cleaning [1], and monitoring of urban environments [13], manufacturing plants [14], and endangered species [15].

The most popular approaches to sensor path planning include coverage path-planning [2], [10], random [2], grid [16], and optimal search strategies [16], [17]. Optimal search strategies typically outperform other approaches in applications where *a priori* information is available, such as sensor models, environmental conditions, and prior measurements [16]. However, they do not yet provide a systematic and general approach to geometric sensor planning. Geometric sensing problems require a description of the geometry and position of the targets and of the sensor's field of view (FOV) [18]. Viewpoint planning has been shown by several authors to be an effective approach for optimally placing or moving vision sensors based on the target geometry and sensor FOV, using weighted functions or tessellated space approaches [18]–[20]. Probabilistic deployment is an effective approach for detecting targets in a region of interest (ROI) by computing a search path based on the probability of finding a target in every unit bin of a discretized obstacle-free workspace [1], [21], [22].

In this paper, an approximate cell decomposition approach is developed for solving the aforementioned treasure hunt problem. Its advantage over existing sensor path planning techniques is that it takes into account the motion and geometry of closed and bounded subsets of an Euclidean space representing the sensor's platform and FOV, as well as the geometry and position of multiple fixed targets and obstacles in the ROI. Traditionally, approximate cell decomposition has been used to plan the motions of a robot with geometry $\mathcal{A}$, in order to avoid collisions with multiple fixed obstacles in a workspace $\mathcal{W}$ (comprising the ROI). In this paper, the approximate-and-decompose approach proposed in [23] is modified to plan the motions of a robotic sensor with FOV $\mathcal{S}$ and platform geometry $\mathcal{A}$, in order to make measurements from multiple targets in $\mathcal{W}$, while avoiding obstacles. Since the sensor is installed on-board the robot, the configuration of both $\mathcal{A}$ and $\mathcal{S}$ can be specified with respect to the same coordinate frame. Then, the free configuration space is decomposed to obtain a connectivity graph with observation cells that each enable a unique set of measurements from one or more targets in $\mathcal{W}$. This novel decomposition can be considered as a systematic approach for constructing so-called *detection cells*, used for information-driven sensor planning in [24] and [25].

Information-driven sensor planning has been shown by several authors to be a general and effective framework for

C. Cai is with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 USA (e-mail: cc88@duke.edu).

S. Ferrari is with the Department of Mechanical Engineering and Material Science, Duke University, Durham, NC 27708 USA (e-mail: sferrari@duke.edu).
Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2008.2008561

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

CAI AND FERRARI: INFORMATION-DRIVEN SENSOR PATH PLANNING BY APPROXIMATE CELL DECOMPOSITION 673

computing the expected measurements' benefit in sensor planning problems [24]–[27]. While robot path planning typically aims to optimize a deterministic additive function such as Eucledian distance, sensor path planning aims to optimize a stochastic sensing objective that is not necessarily additive. Moreover, the sensor's position and parameters (or mode) must be planned prior to obtaining sensor measurements. Therefore, while the measurements ultimately determine performance with respect to the sensing objective (e.g., classification), they cannot be factored into the planning problem [24]–[28]. Recently, the authors showed that using an additive expected entropy reduction (EER) function leads to improved target classification in non-Gaussian sensor fusion [29]. In this paper, EER is used to formulate the expected benefit of the sensor measurements in terms of a posterior probability mass function (PMF) obtained from *a priori* information (Section III). A procedure is presented for pruning and transforming the connectivity graph into a decision tree that is used to determine the sensing strategy with maximum expected measurement profit.

The sensor path planning methodology presented in Section IV is demonstrated through a mine-hunting application in Section V. The objective of modern demining systems is to clear mines and unexploded ordnance (UXO) safely, rapidly, and at low cost by first surveying the ROI with remote sensors and, then, deploying a robotic sensor on the ground [12], [30], [31]. As shown in Section VI, the proposed method achieves better efficiency than complete coverage, random, and grid searches (adapted from [2], [10], and [16]) in ROIs exhibiting low-to-high densities of targets, obstacles and narrow passages, and nonuniform soils, weather, and vegetation. Since the method includes obstacle-avoidance capabilities, it can also be used to plan the path of sensors installed on non-overpass capable platforms [12] that must avoid driving over targets.

## II. PROBLEM FORMULATION AND ASSUMPTIONS

The objective of the robotic sensor is to infer hidden *hypothesis* variables associated with multiple targets from fused sensor measurements and environmental information. Let $\mathcal{W} \subset \mathbb{R}^2$ denote an Euclidean sensor workspace, or ROI, populated with $r$ fixed targets $\mathcal{T}_1, \ldots, \mathcal{T}_r$ and $n$ fixed obstacles $\mathcal{B}_1, \ldots, \mathcal{B}_n$, with geometries and positions estimated from prior sensor measurements. Assume that with each target $\mathcal{T}_i$, there is associated one hypothesis variable $y_i$ that is discrete and, possibly, random, with a finite range $\mathcal{Y} = \{y_i^1, \ldots, y_i^p\}$, where $y_i^k$ denotes the $k$th value of $y_i$. $y_i$ cannot be directly measured or observed, but it can be inferred from a set of test variables or *measurements*, $M_i = \{m_{i1}, \ldots, m_{i\wp}\}$, through a known joint PMF, $P(y_i, M_i)$, that is available from a probabilistic sensor model (Section III-A). Every measurement variable $m_{i\ell}$ is random and discrete, with a finite range $\mathcal{M}_{i\ell} = \{m_{i\ell}^1, \ldots, m_{i\ell}^{N_\ell}\}$, where $m_{i\ell}^k$ denotes the $k$th value of $m_{i\ell}$, and $\ell = 1, \ldots, \wp$. For example, $y_i$ may represent the classification of the $i$th target, and the measurements in $M_i$ may pertain $\wp$ target features that can be measured when the $i$th target lies in the sensor's FOV. The values of $M_i$ are unknown *a priori*, and are uncertain due to random measurement errors.

The geometry of the robotic platform is denoted by $\mathcal{A}$, and its configuration $q$ specifies the position and orientation of a moving Cartesian frame $\mathcal{F}_\mathcal{A}$, embedded in $\mathcal{A}$, with respect to a fixed Cartesian frame $\mathcal{F}_\mathcal{W}$. A sensor with an FOV represented by $\mathcal{S}$ is mounted on $\mathcal{A}$ with a fixed position and orientation that also is specified through $\mathcal{F}_\mathcal{A}$. The geometric objects $\mathcal{S}$ and $\mathcal{A}$ are assumed to be closed and bounded subsets of $\mathcal{W}$. This paper addresses the problem of planning the path of the robotic sensor in $\mathcal{W}$ for the purpose of enabling sensor measurements from the targets, while avoiding collisions with the obstacles. A novel cell decomposition methodology, presented in Section IV-A, is used to discretize the robot configuration space into cells that each enable measurements over a unique set of targets in $\mathcal{W}$. By this methodology, at every time index $t_k$, the sensor occupies one and only one cell, and makes a test decision $u(t_k)$ on which mode to use to make the available measurements, and an action decision $a(t_k)$ on which adjacent cell to move to at time $t_{k+1}$ (see [32, Ch. 4] for a review of test and action decisions). Both $u$ and $a$ are discrete decision variables, with finite ranges $U_k$ and $A_k$ representing admissible test and action decisions at time $t_k$, respectively [33, p. 13].

Several approximate information-theoretic functions have been developed for assessing the measurement benefit based on sensor models and prior information in a manner that is computationally tractable [26]. In this paper, the measurement benefit function $B(t_k)$ is defined as the EER of the measurements, conditioned upon prior information (Section III-B). The measurement cost depends on the energy or power required to operate the sensor in a given sensor mode $J(t_k)$ and on the distance traveled $D(t_k)$, which is assumed to be representative of the energy or power expended by the platform. Then, the robotic sensor's performance at $t_k$ is defined as the *measurement profit*

$$R(t_k) = w_B \cdot B(t_k) - w_J \cdot J(t_k) - w_D \cdot D(t_k) \quad (1)$$

where $w_B$, $w_J$, and $w_D$ are user-defined constant weights for the respective objective functions, chosen based on their units and on the desired tradeoff between measurement benefit and cost. As shown in [34], $B$ is an additive reward function of $P(y_i, M_i)$, and $J$ and $D$ are additive by definition. Thus, a decision strategy for the robotic sensor can be obtained by solving the following problem.

*Problem 2.1 (Treasure Hunt Problem):* Given a layout $\mathcal{W}$ and a joint PMF $P(y_i, M_i)$, $i = 1, \ldots, r$, find the sequence of decisions or *strategy* $\sigma^* = \{u(t_k), a(t_k) | k = 0, \ldots, f\}$ that maximizes the total expected measurement profit

$$V = \mathbb{E}\left\{ \sum_{k=0}^{f} R(t_k) \right\} \quad (2)$$

and provides a motion plan $\tau^* \equiv \{a(t_0), a(t_1), \ldots, a(t_f)\}$ for a robotic sensor with FOV $\mathcal{S}$ and platform $\mathcal{A}$ between an initial and a final configuration in $\mathcal{W}$.

## III. BACKGROUND ON INFORMATION-DRIVEN SENSOR PLANNING

A basic difficulty in sensor planning consists of assessing the benefit of the sensor measurements prior to obtaining them from the targets. Several information-theoretic metrics have been proposed for this purpose. Relative entropy was used in [27] to solve a multisensor–multitarget assignment problem, and in [24] and [25] to manage agile sensors with Gaussian models for target detection and classification. Information entropy and the Mahalanobis distance measure were used in [26] for sensor selection in *ad-hoc* sensor networks. Recently, the authors showed that using entropy reduction leads to improved target classification and feature inference from multiple heterogeneous sensor measurements, when the sensor models are not necessarily Gaussian [29]. Therefore, EER is used to formulate the measurement benefit function $B$ in terms of a joint PMF known as the *sensor model*, as explained in the following sections.

### A. Review of BN Sensor Modeling

A common approach for modeling the sensor measurement process is to utilize a joint PMF of the relevant variables, which may include target classification and features, sensor measurements and parameters (or mode), and environmental conditions. The joint PMF of a particular sensor may be obtained by means of estimation algorithms (e.g., [25], [26]), or by learning algorithms using, for example, wavelets or mixtures of Gaussians [16], [35]. In this paper, we adopt the method presented in [36] and [37], in which the PMF is learned from data and represented by a Bayesian network (BN) model. The advantages of BN models are that they can easily deal with many variables, they are accompanied by very efficient learning and inference algorithms, and they provide a convenient factorization of the joint PMF that can be used to simplify the computation of posterior PMFs required by information theoretic metrics (Section III-B).

A BN model is comprised of a directed graph and a set of conditional probability tables (CPTs) that together specify the multivariate joint PMF of a set of discrete and random variables known as the BN *universe* [32]. Every random variable in the BN universe is assumed to have a finite range, and is represented by a node in the graph. Arcs between the nodes represent conditional probability relationships between the variables. As shown in [36] and [37], the BN model of a sensor measurement process is obtained by defining the universe $X_\mathcal{S}$ as the set of all variables that influence the sensor measurements, such as, the sensor mode $v$, the set of environmental conditions $E$, the set of measurements $M$, the set of actual target features $F$ to be inferred from $M$, and the target classification $y$, i.e., $X_\mathcal{S} \equiv \{v, E, M, F, y\}$, where nodes are denoted by lower case letters, and upper case letters denote sets of nodes. The BN structure and CPTs are determined from a database of prior sensor measurements, using BN batch learning algorithms [38]. The database consists of several cases in which all nodes in $X_\mathcal{S}$ are sampled by obtaining sensor measurements over known targets, under known operating and environmental conditions, as shown in [36].
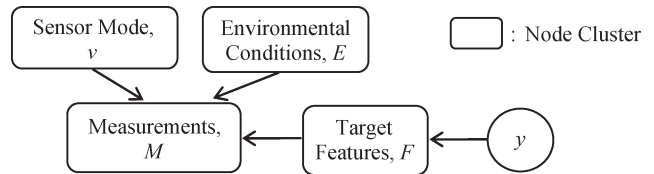


Fig. 1. Typical architecture of BN sensor model.

After the BN model is determined, it specifies the joint PMF underlying the sensor measurements in terms of the recursive factorization

$$P(X_\mathcal{S}) \equiv P(v, E, M, F, y) = \prod_{x_l \in X_\mathcal{S}} P(x_l \mid pa(x_l)) \quad (3)$$

$$= P(M \mid v, E, F)P(F \mid y)P(y)P(v)P(E) \quad (4)$$

where $pa(x_l)$ denotes the set of parents of node $x_l$ in $X_\mathcal{S}$, and factors in (3) are conditional PMFs given by the BN CPTs. Since the parents of a node $x_l$ are all the nodes in the BN with an outgoing arc to $x_l$, the factorization in (3) reflects the BN graph structure, which is learned from data and, thus, depends on the sensor type. The factorization in (4), shown in Fig. 1, has been shown to apply to various sensor types [36], [37].

By this approach, non-Gaussian sensor models can be obtained and used for sensor planning. In the case of multiple heterogeneous sensors, the BN models of each sensor type are used in combination with the Dempster–Shafer rule of evidence combination to obtain a fused posterior PMF [36]. In this paper, the factorization (4) is used to compute the expected benefit of the measurements before they become available, as explained in the next section.

### B. Measurement Benefit Function

In this paper, the measurement benefit function is derived using the so-called EER, following an approach that was first presented in [29]. Entropy reduction is formulated in terms of conditional entropy, which can be used to represent the uncertainty in a discrete and random variable $y$, given the value of another discrete and random variable $z$, based on their joint and conditional PMFs

$$H(y \mid z) = -\sum_{z^j \in \mathcal{Z}} \sum_{y^k \in \mathcal{Y}} P(y = y^k, z = z^j) \log_2 P(y = y^k \mid z = z^j)$$

$$(5)$$

where $\sum$ denotes marginalization, and $y^k$ denotes the $k$th value in the finite range of $y$, $\mathcal{Y} = \{y^1, \ldots, y^p\}$ [39]. Although entropy is not additive, it can be shown [34] that the entropy reduction or mutual information $I(y; z_j \mid z_i) = H(y \mid z_i) - H(y \mid z_i, z_j)$ is additive, and represents the reduction in uncertainty brought about by $z_j$, given prior information or evidence about $z_i$.

Suppose the sensor measurements are sought to reduce the uncertainty in the target classification $y$. Then, entropy reduction can be used to represent the benefit of a new (posterior) set of measurements $M = \{m_1, \ldots, m_\wp\}$, given an *a priori evidence set* $\mathcal{E}^0 = \{E^0, M^0, v^0\}$, which may include known

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

CAI AND FERRARI: INFORMATION-DRIVEN SENSOR PATH PLANNING BY APPROXIMATE CELL DECOMPOSITION 675

environmental conditions, as well as the measurements and mode of a previously deployed sensor, where it is assumed that every measurement $m_\ell \in M$ is a discrete and random variable, with a finite range $\mathcal{M}_\ell = \{m_\ell^1, \ldots, m_\ell^{N_\ell}\}$. The superscript $(\cdot)^0$ denotes one or more random variables whose values are known *a priori*. Since the actual mutual information cannot be determined prior to measuring $M$, the EER

$$\Delta H(y; \, M \mid \mathcal{E}^0) \equiv H(y \mid \mathcal{E}^0)$$
$$- \sum_{\ell=1}^{\wp} \sum_{m_\ell^k \in \mathcal{M}_\ell} \left[ H\left( y \mid m_\ell = m_\ell^k, \mathcal{E}^0 \right) P\left( m_\ell = m_\ell^k \mid \mathcal{E}^0 \right) \right] \quad (6)$$

adopted from [29], is used to represent the expected reduction in uncertainty in $y$ that would be brought about by $M$, given $\mathcal{E}^0$.

The conditional entropy $H(y \mid \mathcal{E}^0)$ in (6) is computed from (5), and from the posterior PMF $P(y \mid \mathcal{E}^0)$ obtained by a junction-tree algorithm [40]. $H(y \mid m_\ell = m_\ell^k, \mathcal{E}^0)$ is computed from (5) using the posterior PMF in (7), shown at the bottom of the next page. Where $\sum_{f_j \in F}$ and $\sum_{f_j^l \in \mathfrak{F}_j}$ denote marginalization over the finite range $\mathfrak{F}_j$ of every target feature $f_j \in F$, and $P(F \mid y, \mathcal{E}^0)$ and $P(M \mid F, \mathcal{E}^0)$ are obtained by evidence-updating algorithms [40]. The relationship in (7) is derived using the simplification

$$P(M \mid y) = \sum_{f_j \in F} \sum_{f_j^l \in \mathfrak{F}_j} P\left( M | f_j^l, y \right) P\left( f_j^l, y \right)$$
$$= \sum_{f_j \in F} \sum_{f_j^l \in \mathfrak{F}_j} P\left( M | f_j^l \right) P\left( f_j^l, y \right) \quad (8)$$

obtained by noting that $y$ and $M$ are $d$-separated given $F$ [40]. Finally, the last term in (6) is $P(M \mid \mathcal{E}^0) = \sum_{f_j \in F} \sum_{f_j^l \in \mathfrak{F}_j} P(M \mid f_j = f_j^l) P(f_j = f_j^l \mid \mathcal{E}^0)$, where $P(F \mid \mathcal{E}^0)$ is obtained using a junction-tree algorithm [40].

In the presence of multiple targets, let $y_i$, $M_i$, and $\mathcal{E}_i^0$ denote the classification, measurement set, and evidence set corresponding to target $i$. Then, the benefit of performing a set of multiple measurements $Z = \{M_j, M_k, \ldots, M_l\}$, in order to classify targets $j$, $k$, $\ldots$, and $l$, is the cumulative EER

$$B(Z) \equiv \Delta H\left( y_j, y_k, \ldots, y_l; Z \mid \mathcal{E}_j^0, \mathcal{E}_k^0, \ldots, \mathcal{E}_l^0 \right)$$
$$= \sum_{M_i \in Z} \Delta H\left( y_i; M_i \mid \mathcal{E}_i^0 \right). \quad (9)$$

Since the BN sensor model (4) holds for any target, every term in the above summation can be computed using the BN CPTs, as shown by (5)–(8). The methodology presented in the next section decomposes the robotic sensor configuration space into discrete cells that each enable one set of measurements $Z$. Then, the measurement benefit of each cell can be computed from (9), and the sensor movements from cell to cell can be planned based on the most profitable measurement sequence.

## IV. METHODOLOGY: INFORMATION-DRIVEN ROBOTIC SENSOR PATH PLANNING

Existing robot path planning methods have been devised to account for the presence of obstacles that the robot must avoid to reach a goal configuration in $\mathcal{W}$ [41]. Cell decomposition is a well-known obstacle avoidance method that decomposes the obstacle-free robot configuration space into a finite collection of nonoverlapping convex polygons, known as cells, within which a robot path is easily generated. Although it is computationally intensive, its advantage over other robot path-planning approaches, such as roadmap or potential field methods, is that, under proper assumptions, cell decomposition is resolution complete. Exact cell decomposition, which has been applied to restricted classes of robot geometries, such as planar objects, 3-D convex polytopes, and polyhedral objects [42]–[44], is guaranteed to find a free path in $\mathcal{W}$, whenever one exists, and otherwise to return failure. Approximate cell decomposition [45]–[52] requires all cells to have the same predefined shape in order to simplify the implementation and reduce the sensitivity to numerical approximations, and has a precision that can be made arbitrarily small at the expense of the running time.

Let the configuration space $\mathcal{C}$ denote the space of all possible robot configurations. A C-obstacle is a subset of $\mathcal{C}$ that causes collisions with at least one obstacle in $\mathcal{W}$, i.e., $\mathcal{CB}_i \equiv \{q \in \mathcal{C} \mid \mathcal{A}(q) \cap \mathcal{B}_i \neq \emptyset\}$, where $\mathcal{A}(q)$ denotes the subset of $\mathcal{W}$ occupied by the platform geometry $\mathcal{A}$ when the robot is in the configuration $q$ [41]. Then, the union $\bigcup_{j=1}^{n} \mathcal{CB}_j$ is the *C-obstacle region*, and the obstacle-free robot configuration space is defined as

$$\mathcal{C}_{\text{free}} \equiv \mathcal{C} \setminus \bigcup_{j=1}^{n} \mathcal{CB}_j = \left\{ q \in \mathcal{C} \mid \mathcal{A}(q) \cap \left( \bigcup_{j=1}^{n} \mathcal{B}_j \right) = \varnothing \right\}. \quad (10)$$

The approximate rectangloid decomposition method, referred to as *approximate-and-decompose* [23], can be utilized to obtain an approximate cell decomposition of $\mathcal{C}_{\text{free}}$ for a robot $\mathcal{A}$ that is capable of translating and rotating in $\mathcal{W}$. In this method, cells of a predefined rectangloid shape are used to decompose the bounding and bounded approximations of the obstacles, until the connectivity of $\mathcal{C}_{\text{free}}$ is properly represented [23]. Then, the union of the cells that are strictly outside the C-obstacle region is used to construct a connectivity graph representing the adjacency relationships between the cells. Finally, the connectivity graph is searched for the shortest path between the two cells containing the initial and final configurations, $q_0$ and $q_f$.

### A. Approximate Cell Decomposition in the Presence of Targets

In this section, a method based on the approximate-and-decompose approach [23] is developed for robotic sensor path planning in the presence of targets. It is assumed that the sensor's position and orientation are fixed with respect to $\mathcal{A}$ and, therefore, can be specified by the configuration vector $q$. In the presence of targets, $q$ must avoid intersections between the platform geometry and the obstacles to prevent collisions, and must enable intersections between the FOV and the targets
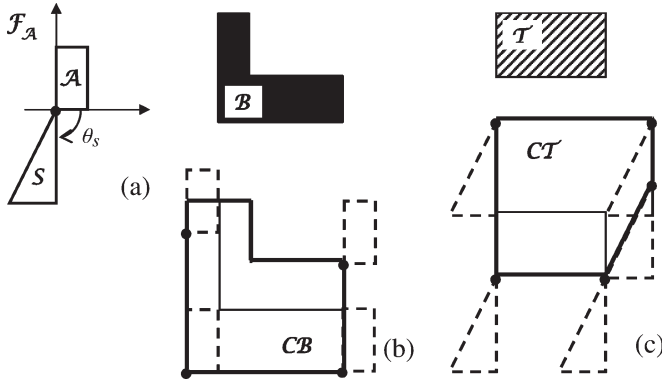
This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

676 IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 39, NO. 3, JUNE 2009

Fig. 2. Example of (b) C-obstacle and (c) C-target (solid lines) obtained for (a) a sensor with FOV $\mathcal{S}$ that is installed on a platform with geometry $\mathcal{A}$, at a fixed orientation $\theta_s$.

to make sensor measurements. The method developed in this section plans the robot motion based on the targets that can be measured by the on-board sensor, by treating the targets as the dual of the obstacles. Without loss of generality, the method is presented for $\mathcal{C} = \mathcal{W} \times \Theta$, where $\mathcal{W} \subset \mathbb{R}^2$ and $\Theta = [\theta^{\min}, \theta^{\max}]$ is the range of possible robot orientations. Then, the configuration vector is defined as $q \equiv [x \quad y \quad \theta]^{\mathrm{T}}$, where $(x, y)$ and $\theta$ denote the robot coordinates and orientation in $\mathcal{F}_{\mathcal{W}}$, respectively.

As shown by the following definitions, the subset of $\mathcal{C}$ that enables intersections with $\mathcal{T}_i$ to make the set of sensor measurements $M_i$ can be defined similarly to a C-obstacle.

*Definition 4.1 (FOV):* The FOV of a sensor mounted on $\mathcal{A}$ is a closed and bounded subset $\mathcal{S}(q) \subset \mathcal{W}$ such that the measurement set of a target located at any point $p \in \mathcal{S}(q)$ can be obtained by the sensor when the robot occupies the configuration $q \in \mathcal{C}$.

*Definition 4.2 (C-Target):* The target $\mathcal{T}_i$ in $\mathcal{W}$ maps in the robot's configuration space $\mathcal{C}$ to the C-target region $\mathcal{CT}_i = \{q \in \mathcal{C} \mid \mathcal{S}(q) \cap \mathcal{T}_i \neq \varnothing\}$.

The relationship between a C-target and a C-obstacle is shown in Fig. 2. For simplicity, in this simple example it is assumed that $\mathcal{A}$ is a platform that can translate freely but cannot rotate. $\mathcal{S}$ is a triangle with a fixed orientation $\theta_s$ with respect to $\mathcal{F}_{\mathcal{A}}$. For this robotic sensor [Fig. 2(a)], the C-obstacle $\mathcal{CB}$ corresponding to an L-shaped obstacle $\mathcal{B}$ (black) is shown in Fig. 2(b), and the C-target $\mathcal{CT}$ corresponding to a rectangular target $\mathcal{T}$ (diagonal pattern) is shown in Fig. 2(c).

In classical cell decomposition, $\mathcal{C}_{\text{free}}$ is decomposed into cells that are represented by nodes in the connectivity graph. Then, the graph is searched by minimizing an Euclidian distance function: $D : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$. However, the robotic sensor's

performance (1) also depends on the measurement set, as shown in (9). Thus, in order to obtain a single-valued performance function, $R : \mathcal{C} \times \mathcal{C} \times Z \to \mathbb{R}$, the decomposition presented in this section obtains cells with the following properties:

*Definition 4.3:* A *void cell* is a convex polygon $\kappa$ in $\mathcal{C}_{\text{free}}$ with the property that none of the targets are observable from any of the configurations in $\kappa$.

*Definition 4.4:* An *observation cell* is a convex polygon $\bar{\kappa}$ in $\mathcal{C}_{\text{free}}$ with the property that every configuration in $\bar{\kappa}$ enables a nonempty set of measurements $Z(\bar{\kappa}) = \{M_i \mid q \in \bar{\kappa}, q \in \mathcal{CT}_i\}$.

Additionally, in order to construct a conservative but efficient representation of $\mathcal{C}_{\text{free}}$, cells are required to have a simple predefined shape, e.g., a rectangloid. This leads to an approximate decomposition that is obtained by repeated simple computations, regardless of the shape of the target and obstacle geometries, or of the dimensions of $\mathcal{C}$. From hereon, $\kappa$ denotes a rectangloid, i.e., a closed region in $\mathcal{C}$ defined as $\kappa = [x_\kappa, x'_\kappa] \times [y_\kappa, y'_\kappa] \times [\theta_\kappa, \theta'_\kappa]$, with dimensions $(x'_\kappa - x_\kappa)$, $(y'_\kappa - y_\kappa)$, and $(\theta'_\kappa - \theta_\kappa)$ chosen by the user to achieve the desired tradeoff between precision and running time. The size of the rectangloids is adapted based on the local geometry of the C-obstacle region, constructing successively smaller decompositions until an obstacle-free path is found in the corresponding connectivity graph.

Due to their predefined shape, the cells' boundaries in an approximate decomposition do not always coincide with the boundaries of C-obstacles/targets. Therefore, a subset of cells, referred to as *mixed*, may contain configurations from $\mathcal{C}_{\text{free}}$, as well as from the C-obstacle/target region. Mixed cells are excluded from the connectivity graph because they may lead to collisions with the obstacles, or to missed target measurements. Their volume can be minimized by approximating C-obstacles and C-targets as unions of nonoverlapping rectangloids, before performing the decomposition [23]. The C-obstacles' approximations are used to determine which rectangloids to exclude from the connectivity graph. The C-targets' approximations are decomposed into observation cells that are labeled by their measurement set, and then inserted in the connectivity graph. Therefore, two different approximations are used to obtain conservative representations of C-obstacles and C-targets:

*Definition 4.5:* A *bounding rectangloid approximation* of $\mathcal{CB}_j[\kappa] = \mathcal{CB}_j \cap \kappa$, denoted by $\mathcal{RB}_j[\kappa]$, is a collection of nonoverlapping rectangloids $\mathcal{R}_{jv}$, $v = 1, \ldots, p$, whose union contains $\mathcal{CB}_j[\kappa]$.

*Definition 4.6:* A *bounded rectangloid approximation* of $\mathcal{CT}_i[\kappa] = \mathcal{CT}_i \cap \kappa$, denoted by $\mathcal{R}'\mathcal{T}_i[\kappa]$, is a collection of nonoverlapping rectangloids $\mathcal{R}'_{iv}$, $v = 1, \ldots, p'$, whose union

$$P\left(y \mid m_\ell^k, \mathcal{E}^0\right) = \frac{P(y \mid \mathcal{E}^0)P\left(m_\ell^k \mid y, \mathcal{E}^0\right)}{P\left(m_\ell^k \mid \mathcal{E}^0\right)}$$

$$= \frac{P(y \mid \mathcal{E}^0) \sum_{f_j \in F} \sum_{f_j^l \in \mathfrak{F}_j} P\left(m_\ell^k \mid f_j = f_j^l, \mathcal{E}^0\right) P\left(f_j = f_j^l \mid y, \mathcal{E}^0\right)}{\sum_{f_j \in F} \sum_{f_j^l \in \mathfrak{F}_j} P\left(m_\ell^k \mid f_j = f_j^l, \mathcal{E}^0\right) \sum_{y^k \in \mathcal{Y}} P\left(f_j = f_j^l \mid y = y^k, \mathcal{E}^0\right) P(y = y^k \mid \mathcal{E}^0)}, \quad \forall \ell, k \quad (7)$$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

CAI AND FERRARI: INFORMATION-DRIVEN SENSOR PATH PLANNING BY APPROXIMATE CELL DECOMPOSITION 677
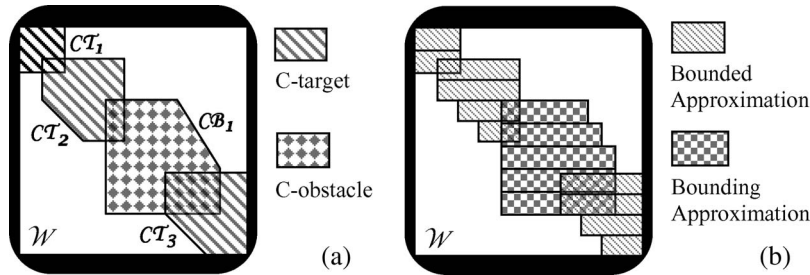


Fig. 3. Simple example of workspace $\mathcal{W}$ populated with (a) both C-obstacles (b) and C-targets and corresponding bounded and bounding approximations.

is contained in $\mathcal{CT}_i[\kappa]$. Examples of bounding and bounded approximations for the C-obstacle and C-targets in Fig. 3(a) are shown in Fig. 3(b).

The following procedure obtains an approximate rectangloid decomposition $\mathcal{K} = \mathcal{K}_{\mathrm{void}} \bigcup \mathcal{K}_z$, comprised of a set of void cells $\mathcal{K}_{\mathrm{void}}$ and a set of observation cells $\mathcal{K}_z$, for a workspace $\mathcal{W}$ populated with $n$ obstacles $\mathcal{B}_1, \ldots, \mathcal{B}_n$, and $r$ targets $\mathcal{T}_1, \ldots, \mathcal{T}_r$.

1) Cut $\Theta = [\theta^{\min}, \theta^{\max}]$ into $\nu$ nonoverlapping subintervals $\mathcal{I}_u = [\gamma_u, \gamma_{u+1}]$, where $u = 1, \ldots, \nu$, $\nu \geq 1$, $\gamma_1 = \theta^{\min}$, and $\gamma_{\nu+1} = \theta^{\max}$; then, let $\kappa^u = [x_\kappa, x'_\kappa] \times [y_\kappa, y'_\kappa] \times \mathcal{I}_u$.

2) Compute $\mathcal{CB}_j[\kappa^u]$, $j = 1, \ldots, n$ and $\mathcal{CT}_i[\kappa^u]$, $i = 1, \ldots, r$, for every $u = 1, \ldots, \nu$, by discretizing $\mathcal{I}_u$ into $k_u$ fixed values $\gamma_u + l\Delta\theta$, where $0 \leq l \leq k_u$, and $\Delta\theta = (\gamma_{u+1} - \gamma_u)/k_u$.

3) For every $u = 1, \ldots, \nu$ and $j = 1, \ldots, n$, compute the outer projection

$$\mathcal{OCB}_j[\kappa^u] = \{(x, y) \mid \exists \theta \in \mathcal{I}_u : (x, y, \theta) \in \mathcal{CB}_j[\kappa^u]\} \tag{11}$$

and generate bounding rectangloid approximation $\mathcal{RB}_j[\kappa^u]$ of $\mathcal{OCB}_j[\kappa^u] \times \mathcal{I}_u$; then, for every $u = 1, \ldots, \nu$ and $i = 1, \ldots, r$, compute the inner projection

$$\mathcal{ICT}_i[\kappa^u] = \{(x, y) \mid \forall \theta \in \mathcal{I}_u : (x, y, \theta) \in \mathcal{CT}_i[\kappa^u]\} \tag{12}$$

and generate bounded rectangloid approximation $\mathcal{R}'\mathcal{T}_i[\kappa^u]$ of $\mathcal{ICT}_i[\kappa^u] \times \mathcal{I}_u$.

4) For every $u = 1, \ldots, \nu$, generate a rectangloid decomposition $\mathcal{K}^u_{\mathrm{void}}$ of the void configuration space

$$\mathcal{C}^u_{\mathrm{void}} \equiv \kappa^u \setminus \left\{ \bigcup_{j=1}^{n} \mathcal{RB}_j[\kappa^u] \cup \bigcup_{i=1}^{r} \mathcal{R}'\mathcal{T}_i[\kappa^u] \right\} \tag{13}$$

and let $\mathcal{K}_{\mathrm{void}} = \cup_{u=1}^{\nu} \mathcal{K}^u_{\mathrm{void}}$.

5) For every $u = 1, \ldots, \nu$ and $i = 1, \ldots, r$, generate a rectangloid decomposition $\mathcal{K}^u_{z,i}$ of $\mathcal{C}^u_{z,i} \setminus \bigcup_{l \neq i} \mathcal{C}^u_{z,l}$, where

$$\mathcal{C}^u_{z,i} = \mathcal{R}'\mathcal{T}_i[\kappa^u] \setminus \bigcup_{j=1}^{n} \mathcal{RB}_j[\kappa^u] \tag{14}$$

and, for every $u = 1, \ldots, \nu$, generate a rectangloid decomposition $\mathcal{K}^u_{z,l}$ of $\bigcup_{i=1}^{r} \{\mathcal{C}^u_{z,i} \cap \bigcup_{l \neq i} \mathcal{C}^u_{z,l}\}$; then, let $\mathcal{K}_z = \bigcup_{u=1}^{\nu} \{\mathcal{K}^u_{z,i} \cup \mathcal{K}^u_{z,l}\}$.

As an example, the approximate cell decomposition $\mathcal{K}$ obtained for the workspace in Fig. 3 is shown in Fig. 4. As shown in Appendix I, the above decomposition is not significantly harder than one involving only obstacles. Let $n_\mathcal{B}$ and $n_\mathcal{T}$ denote the number of edges of all obstacles and all targets, respectively. It was shown in [23] that if all geometric objects, including $\mathcal{A}$ and $\mathcal{S}$, can be approximated by convex polygons, an approximate cell decomposition involving only obstacles can be performed in time $\mathcal{O}(n_\mathcal{B} \log(n_\mathcal{B}))$. We prove in Appendix I that the above decomposition, involving both obstacles and targets, can be performed in time $\mathcal{O}((n_\mathcal{B} + n_\mathcal{T})^2)$. By accounting for the targets, every observation cell obtained in step 5) is labeled by the index set of $Z(\bar{\kappa})$ (Definition 4.4), which can be used to compute the robotic sensor's performance (1) for every cell, and to search for the optimal sensor path, as shown in the next section.

### B. Connectivity Graphs and Optimal Sensing Strategy

In this section, a methodology is presented for representing the treasure hunt problem (Problem 2.1) by a decision tree that is used to compute the optimal sensing strategy $\sigma^*$. As a first step, a connectivity graph is obtained from the approximate cell decomposition presented in the previous section. Then, the connectivity graph is pruned and transformed into a decision tree, reducing the number of feasible channels and the time required to compute $\sigma^*$. The connectivity graph is used to represent the robotic sensor as a point in configuration space, and is defined as follows.

*Definition 4.7:* A *connectivity graph with observations* $\mathcal{G}$ is a nondirected graph where the nodes represent either an observation cell or a void cell, and two nodes $\kappa_i$ and $\kappa_j$ in $\mathcal{G}$ are connected by an arc $(\kappa_i, \kappa_j)$ if and only if the corresponding cells are adjacent in $\mathcal{C}_{\mathrm{free}}$.

Several approaches can be used to search $\mathcal{G}$ for a sequence of adjacent cells, or channel, $\tau^*$ connecting $q_0$ to $q_f$. In this section, we present an efficient approach for computing $\sigma^*$ from $\mathcal{G}$, based on the formulation in Problem 2.1. The search for $\sigma^*$ can be simplified by pruning the connectivity graph with observations based solely on distance, using a label-correcting algorithm presented in [34]. The Euclidean distance in $\mathcal{C}$

$$D(\kappa_i, \kappa_j) \equiv \max \|\mathcal{A}(\hat{q}_i) - \mathcal{A}(\hat{q}_j)\| = d_{ij} = d_{ji}. \tag{15}$$

taken from [53], is attached to every arch $(\kappa_i, \kappa_j)$ in $\mathcal{G}$, where $\hat{q}_i$ denotes the geometric centroid of $\kappa_i$, and $\|\cdot\|$ denotes

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

678    IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 39, NO. 3, JUNE 2009
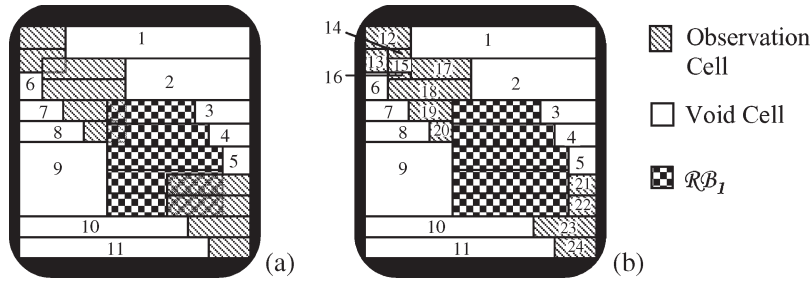


Fig. 4.    Approximate rectangloid decomposition of $\mathcal{W}$ in Fig. 3 into (a) $\mathcal{K}_{\text{void}}$, and (b) $\mathcal{K} = \mathcal{K}_{\text{void}} \bigcup \mathcal{K}_z$, where only cell indices are shown for simplicity.
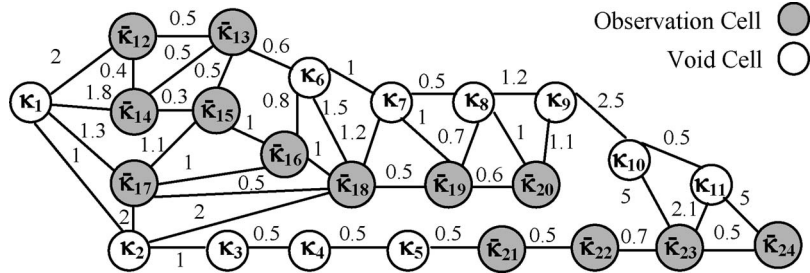


Fig. 5.    Connectivity graph $\mathcal{G}$ obtained from the approximate rectangloid decomposition in Fig. 4 (observation cells are labeled in gray).

the Eucledian norm. An example of the resulting connectivity graph is shown in Fig. 5, for the decomposition in Fig. 4.

Suppose at time $t_k$ the robotic sensor occupies a cell in $\mathcal{G}$. Then, based on the adjacency relationships, the number of cells that can be visited at $t_j > t_k$ grows exponentially with $j$. If the objective of minimizing the distance metric (15) is taken into account, however, the connectivity graph can be pruned to eliminate a significant number of suboptimal channels based on the principle of optimality [54]. We adopt the following definition from [34].

*Definition 4.8:* The *connectivity tree* $T_r$ associated with $\mathcal{G}$ and two cells $\kappa_0 \ni q_0$ and $\kappa_f \ni q_f$ is a tree graph with $\kappa_0$ as the root and $\kappa_f$ as the leaves. The nodes represent void or observation cells, and an additive distance metric $D(\kappa_i, \kappa_j)$ is attached to each arc $(\kappa_i, \kappa_j)$. A branch from the root to a leaf represents a channel $\tau$ joining $\kappa_0$ and $\kappa_f$, and obeys the following properties.

P1)  Two branches are said to be *information equivalent* if they join the same cells and contain the same set of observation cells, regardless of the order.
P2)  Two branches that are information equivalent can coexist in $T_r$ if and only if they represent the same channel.
P3)  A branch in $T_r$ represents a channel with the shortest overall distance of any other information-equivalent branch in $\mathcal{G}$.

The label-correcting algorithm presented in [34] prunes $\mathcal{G}$ and transforms it into a connectivity tree that is guaranteed to contain the shortest channel between $\kappa_0$ and $\kappa_f$, as well as a subset of channels that are *not* information equivalent and connect a subset of observation cells by the minimum distance. As an example, the connectivity tree obtained from the graph in Fig. 5 for $\kappa_0 = \kappa_1$ and $\kappa_f = \kappa_{11}$, is shown in Fig. 6.

Finally, the connectivity tree $T_r$ is transformed into a decision tree that is searched for $\sigma^*$ by maximizing the total

expected measurement profit $V$, defined in (2). Decision trees can be used to describe discrete-time discrete-state decision processes by representing decision variables as decision nodes (rectangular boxes), hypothesis variables as chance nodes (circles), and the value of the objective function as utility nodes or leaves (diamonds). A decision tree is solved by a rolling-back procedure that determines the optimal strategy by recursively estimating the utility of each branch (see [32, Sec. 4.4] for a comprehensive review). The decision tree used to represent Problem 2.1 is a tuple $DT = \{C, D, V, A\}$, with a set of chance nodes $C$, a set of decision nodes $D$, the measurement profit $V$ as the leaves, and a set of directed arcs $A$, which is obtained by the following assignments.

1)  The root is $\kappa_0$, and every node preceding a leaf is $\kappa_f$.
2)  Every chance node $\kappa_i \in C$ represents a void or an observation cell.
3)  An arc $(\kappa_i, \kappa_j) \in A$ represents the action decision to move from $\kappa_i$ to $\kappa_j$.
4)  Every decision node $u(t_k) \in D$ represents the test decision on the mode used to make measurements $Z(t_k) = Z(\bar{\kappa}_j)$, where $\bar{\kappa}_j$ is the node preceding $u(t_k)$.
5)  An arc $(\bar{\kappa}_j, u(t_k))$ is labeled with the test decision of highest profit.
6)  The utility node $V$ at the end of each branch represents the total measurement profit of the corresponding strategy $\sigma$.

The pseudocode for the above assignments is provided in Appendix II. As an example, the decision tree obtained from the connectivity tree in Fig. 6 is shown in Fig. 7. Every chance node representing an observation cell has attached an index set representing the targets that can be measured by the robotic sensor in that cell. For instance, $\bar{\kappa}_{15}$ in Fig. 7 enables measurements over targets $\mathcal{T}_1$ and $\mathcal{T}_2$ [based on Figs. 3 and 4(b)], thus $Z(\bar{\kappa}_{15}) = \{M_1, M_2\}$. Along this branch, $\bar{\kappa}_{15}$ can

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

CAI AND FERRARI: INFORMATION-DRIVEN SENSOR PATH PLANNING BY APPROXIMATE CELL DECOMPOSITION 679
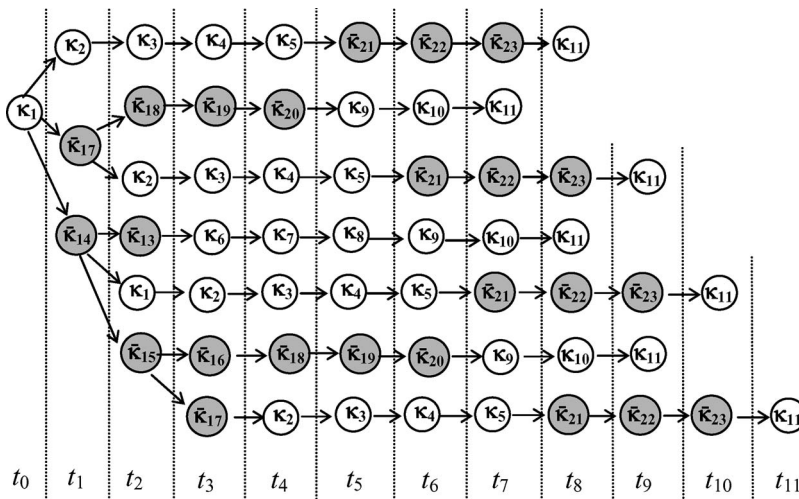


Fig. 6. Connectivity tree $T_r$ obtained from $\mathcal{G}$ in Fig. 5 for $\kappa_0 = \kappa_1$ and $\kappa_f = \kappa_{11}$, with the time index label shown at the bottom of each column.
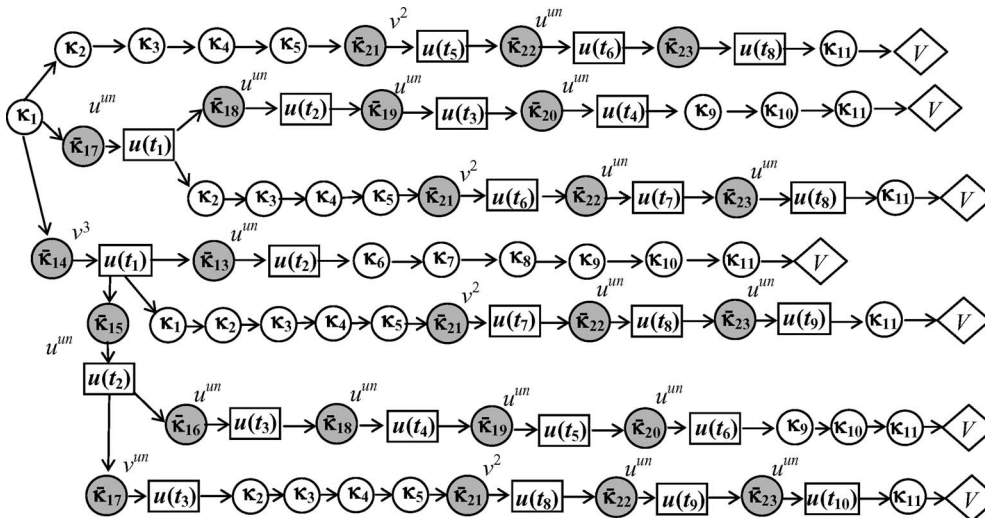


Fig. 7. Decision tree $DT$ obtained from $T_r$ in Fig. 6, where $v^l$ represents the decision of making measurements using the $l$th sensor mode, and $u^{un}$ represents the decision of not making any measurements.

be visited at time $t_2$ (Fig. 6), therefore $Z(t_2) = Z(\bar{\kappa}_{15})$ and, from (9), the profit of the test-decision $u(t_2)$ is the measurement benefit $B[Z(t_2)]$ minus the cost $J[u(t_2)]$ of using the sensor in a particular mode. Arc $(\bar{\kappa}_{15}, u(t_2))$ is labeled with the test-decision value that has the highest profit, which, in this case, is $u^{un}$ (Fig. 6) because the EER is low compared to the cost of operating the sensor in any mode.

In Section VI, the optimal policy $\sigma^*$, computed from $DT$, is applied to plan the path of a robotic sensor used for landmine classification, as explained in the following section.

## V. APPLICATION: SENSOR PATH PLANNING FOR LANDMINE CLASSIFICATION

A modern paradigm for demining systems is to deploy a remote infrared (IR) sensor on an airborne platform to obtain cursory measurements over the entire minefield [12], [31]. Subsequently, a ground-penetrating radar (GPR) is deployed on-board a ground robot to obtain additional measurements

which are fused with IR measurements to estimate the features of objects buried in heterogeneous soils, and to classify them as either landmines or clutter [30]. Because of their different operating principles and modes, the effectiveness of IR sensors is heavily influenced by environmental conditions and can be significantly improved through fusion with GPR measurements [36]. Thus, deploying a robotic GPR sensor can reduce classification errors, such as false alarms, which cause cans or other debris to be mistaken for mines, thereby preventing needless excavation costs. At the same time, its path must be planned to safely avoid obstacles and, possibly, mines, in case the robotic platform is not overpass capable [12]. The distance traveled by the GPR must be minimized to reduce time and energy consumption, in order to clear mines safely, rapidly, and at low cost.

### A. Demining System Simulation and BN Sensor Models

The sensor path planning methodology presented in Section IV is tested on the simulation of a demining sensor

TABLE I
GPR AND IR BN NODES

| Symbol: | Nodes: | Range: |
|---|---|---|
| $y$ | Target classification | {clutter, mine, empty bin} |
| $v$ | GPR mode: $v_{GPR}$ | {depth search, resolution search, anti ground-bounce-effect search} |
| | IR mode: $v_{IR}$ | {surface-mine search, shallow-buried-mine search} |
| $E$ | Soil moisture (%): $s_r$ | {dry [0, 10], wet (10, 40), saturated ($> 40$)} |
| | Soil composition: $s_c$ | {very-sandy, sandy, high-clay, clay, silt} |
| | Soil uniformity: $s_u$ | {yes, no} |
| | Vegetation: $g$ | {no-vegetation, sparse, dense} |
| | Weather: $w$ | {clear, overcast, raining} |
| | Illumination: $i$ | {low (7-10 a.m. and 6-9 p.m.), medium (10-1 p.m.), high (1-6 p.m.)} |
| $F$ | Depth (cm): $d$ | {surface [0], shallow-buried (0, 12], buried (12, 60], deep-buried ($> 60$)} |
| | Size (cm): $z$ | {small (2, 13], medium (13, 24], large (24, 40], extra-large ($> 40$)} |
| | Shape: $s$ | {cylinder, box, sphere, long-slender, irregular} |

system developed in [36]. The simulation generates a rectangular minefield of chosen dimensions, or workspace $\mathcal{W} \subset \mathbb{R}^2$, that includes several buried mines, clutter objects (CLUT), obstacles, and heterogenous environmental conditions. A 2-D grid is superimposed on the minefield dividing it into unit-square bins. Soil characteristics, vegetation, and time-varying meteorological conditions, modeled according to [31] and [55], are assigned to each bin, either at random or at user-specified positions. Targets consist of antitank mines, antipersonnel mines (APM), UXO, and CLUT that are sampled and reproduced using the Ordata Database [56], which contains over 5000 explosive items and 3000 metallic and plastic objects that resemble APMs. Each target occupies one or more bins in the minefield depending on its size $z$, and is characterized by a depth $d$, and shape $s$ (Table I). Thus, $\mathcal{T}_i \subset \mathcal{W}$ represents the geometry of the set of bins from which prior IR measurements of the $i$th target are obtained in $\mathcal{W}$.

In the simulation, as soon as the FOV of the GPR sensor, $\mathcal{S}$, intersects a bin containing a target, measurements are reproduced and deteriorated based on the target features, the sensor's mode and working principles, and the environmental conditions in the bin [36], [57]. IR sensors detect anomalies in infrared radiation and, based on their height above the ground, build an image of a horizontal area, obtaining cursory measurements of $z$ and $s$ for shallow-buried objects. Because they rely on temperature variations, their performance is highly influenced by illumination, weather, vegetation, and soil properties. GPR sensors emit radio waves that penetrate the ground and process their reflections at the boundaries of materials characterized by different refraction indexes producing images of underground vertical slices over $\mathcal{S}$ [58]. The frequency of the radio wave and its bandwidth determine the GPR mode $v_{GPR}$. Since penetration depth increases at lower frequencies and image resolution improves at higher frequencies, the optimal GPR mode depends on target features and on environmental conditions. For example, very high frequencies may be required in the presence of ground discontinuities to overcome the so-called ground-bounce effect [31].

The BN models of GPR and IR sensors, obtained in [36] and shown in Fig. 8, are implemented in this paper. These models are used to infer the features $F_i = \{d_i, z_i, s_i\}$ of the $i$th target from the evidence set $\mathcal{E}_i = \{v_i, E_i, M_i\}$, which includes *measured* target features $M_i = \{d_{m_i}, z_{m_i}, s_{m_i}\}$ extracted from sensor images, the mode $v_i$, and the environmental conditions $E_i$ [36], [57]. Moreover, the GPR BN model [Fig. 8(b)] is used to compute the GPR measurement benefit, as explained in Section III-B. The ground robot is simulated using the nonholonomic unicycle model in $\mathcal{F}_{\mathcal{W}}$ [59], [60], and a geometry $\mathcal{A} \subset \mathcal{W}$ specified by the user. On-board the ground robot is a GPR sensor with an FOV $\mathcal{S} \subset \mathcal{W}$ specified by the user, that moves with $\mathcal{A}$ in $\mathcal{F}_{\mathcal{W}}$. When the sensor is turned on, $\mathcal{S}$ appears on the screen and the simulation produces measurements for all bins intersected by $\mathcal{S}$. When the sensor is turned off, $\mathcal{S}$ disappears and no measurements are made.

### B. GPR Sensor Path Planning Implementation

After a minefield is simulated, the methodology presented in Section IV is implemented to deploy the robotic GPR sensor. The weights in the performance function (1) are chosen to be $w_B = 20$, $w_J = w_D = 1$, based on tests and heuristics. The obstacle-free channel $\tau^* \subset \sigma^*$ is transformed into a free path contained in the interior of $\tau^*$ using a well-known methodology taken from [41, pp. 204–207], which connects the midpoints of $\partial\kappa_i \cap \partial\kappa_j$ between every two adjacent cells $\kappa_i$ and $\kappa_j$ in $\tau^*$ (where $\partial\kappa_i$ denotes the boundary of $\kappa_i$). Then, $\sigma^*$ is executed by moving the robotic sensor along the free path, and by turning the sensor on or off based on the sequence of test decisions in $\sigma^*$.

Various sources of uncertainty cause the expected measurement profit (2) to differ from the actual measurement profit. First, the actual GPR measurements are unknown prior to deploying the sensor. Second, both IR and GPR measurements are subject to errors, which may cause some targets to go undetected, and other targets to be misclassified even after all measurements are obtained and fused. Thus, the method's

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

CAI AND FERRARI: INFORMATION-DRIVEN SENSOR PATH PLANNING BY APPROXIMATE CELL DECOMPOSITION 681
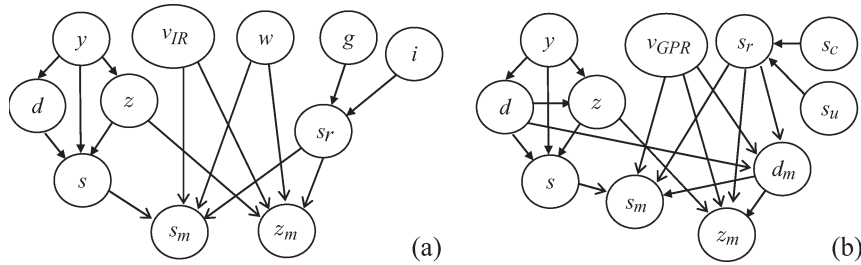


Fig. 8. Architectures of IR and GPR BN sensor models (taken from [36]), with nodes defined in Table I.

effectiveness is evaluated by computing four efficiency metrics *a posteriori*, i.e., after $\sigma^*$ is executed and the fused GPR–IR measurements are processed. The first metric $\eta_y$ represents the classification improvement per unit distance brought about by the GPR. Let $\Delta N_y$ denote the difference between the number of targets that are properly classified using fused GPR–IR measurements, and the number of targets that are properly classified using only IR measurements. Then, $\eta_y(\sigma^*) = \Delta N_y(\sigma^*)/D_{\text{tot}}$, where $D_{\text{tot}}$ is the total distance traveled by the GPR.

The metric $\eta_N(\sigma^*) = N_m(\sigma^*)/D_{\text{tot}}$ represents the number of targets detected by the GPR, $N_m$, per unit distance. Thus, it is useful for assessing the effectiveness of the approximate cell decomposition approach presented in Section IV-A, which is devised to construct and identify observation cells. Although the EER in (6) is used to estimate the entropy reduction, the *actual* entropy reduction can be determined only after obtaining the GPR measurements. From Section III-B, the actual entropy reduction brought about by the GPR measurements $M_i$ over $\mathcal{T}_i$, given prior evidence about $\mathcal{E}_i^0$, is $I(y_i; M_i \mid \mathcal{E}_i^0) \equiv H(y_i \mid \mathcal{E}_i^0) - H(y_i \mid \mathcal{E}_i^0, M_i)$. Then, $B_{\text{tot}} \equiv \sum_{t_k=t_0}^{t_f} \sum_{M_i \in Z(t_k)} I(y_i; M_i \mid \mathcal{E}_i^0)$ is the actual entropy reduction brought about by all GPR measurements obtained by $\sigma^*$, and the metric $\eta_H(\sigma^*) = B_{\text{tot}}/D_{\text{tot}}$ represents the actual measurement benefit per unit distance.

After the GPR measurements are obtained, BN inference is used to obtain the most likely target classification, denoted by $\hat{y}_i$, and its confidence level (CL), $P(y_i = \hat{y}_i \mid \mathcal{E}_i^0)$, using the approach in [36]. Since in many applications a misclassification with a high CL is far worse than one with a low CL, the following error metric has been proposed in [36]:

$$e\left(y_i, y_i^* \mid \mathcal{E}_i^0\right) = \mathbf{p}_i \cdot \mathbf{g}_i \qquad (16)$$

to adjust the classification error by the posterior PMF, where $\mathbf{g}_i$ is a $p \times 1$ vector representing the error between the true value $y_i^*$ and all possible values in $\mathcal{Y}$, $\mathbf{p}_i \equiv [P(y_i^1 \mid \mathcal{E}_i^0) \cdots P(y_i^p \mid \mathcal{E}_i^0)]^{\text{T}}$, and $(\cdot)$ is the dot product. In applications where the values in $\mathcal{Y}$ are not numeric, $\mathbf{g}_i$ is defined based on heuristics and expert knowledge [36], [57]. Let $\Delta e(\sigma^*)$ denote the total reduction in the error (16) that is brought about by $\sigma^*$. Then, the *classification gain* $\eta_{\text{CL}}(\sigma^*) = \Delta e(\sigma^*)/D_{\text{tot}}$ is used to quantify the improvement in classification accuracy per unit distance, given $\mathcal{E}_i^0$.

## VI. RESULTS

The implementation of the sensor path planning methodology presented in Section IV was tested on a variety of
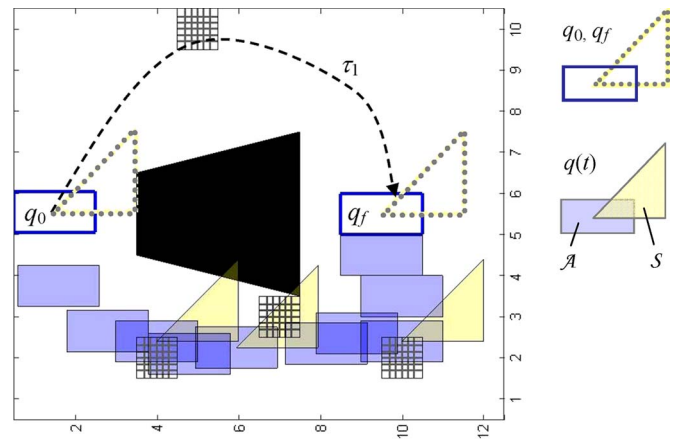


Fig. 9. Influence of targets in $\mathcal{W}$ on the optimal sensor path $\tau^*$, which is plotted by sample sensor configurations.

minefields exhibiting low-to-high densities of targets, obstacles, and narrow passages, as well as nonuniform soils, weather, and environmental conditions. In the next section, a few simple examples are used to illustrate the influence that the geometries of the workspace and of the robotic sensor have on the optimal sensor path. These examples show that both prior information and geometric characteristics must be taken into account when planning the sensor path. The results in Section VI-B demonstrate that the proposed method outperforms complete coverage, random, and grid searches (adapted from [2], [10], [16] using $\mathcal{G}$) and, unlike existing approaches, it also is applicable to nonoverpass capable platforms.

### A. Influence of Prior Information and Sensor Geometries on the Sensor Path

In this section, the optimal sensor path $\tau^* \subset \sigma^*$ obtained by the approximate-and-decompose methodology is shown by plotting sample sensor configurations on the workspace. Other hypothetical paths are schematized by dashed lines for comparison. The first example in Fig. 9 shows that both the location and geometry of targets and obstacles in $\mathcal{W}$ must be accounted for in planning the path of a robotic sensor. Suppose the sensor must travel from $q_0$ to $q_f$, and $\mathcal{W}$ contains one obstacle (black trapezoid) and four equally important targets (gridded squares) (Fig. 9). Although two obstacle-free paths of approximately the same distance can be found from $q_0$ to $q_f$, $\tau_1$ allows the sensor to visit only one target, while $\tau^*$ allows the sensor to visit three targets. It can be seen that by traveling along $\tau_1$ the sensor could not have obtained measurements over these three targets, and
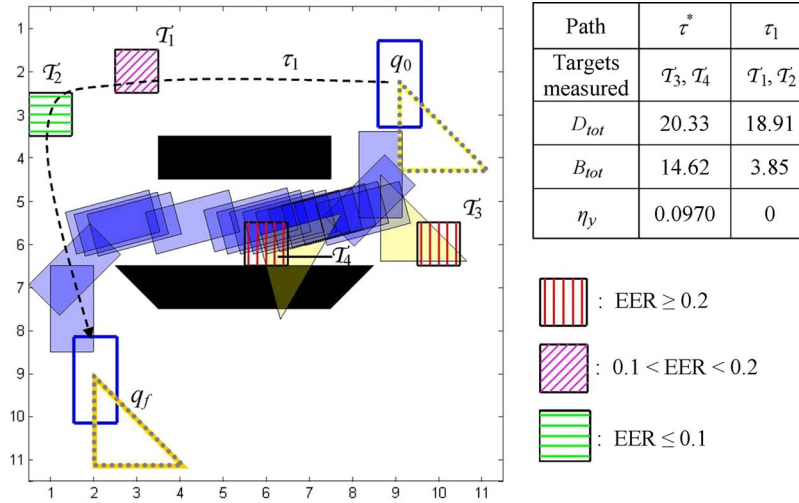
| Path | $\tau^*$ | $\tau_1$ |
|---|---|---|
| Targets measured | $\mathcal{T}_3, \mathcal{T}_4$ | $\mathcal{T}_1, \mathcal{T}_2$ |
| $D_{tot}$ | 20.33 | 18.91 |
| $B_{tot}$ | 14.62 | 3.85 |
| $\eta_y$ | 0.0970 | 0 |

: EER $\geq 0.2$

: $0.1 < $ EER $< 0.2$

: EER $\leq 0.1$

Fig. 10.   Influence of EER on the optimal sensor path $\tau^*$, which is plotted by blue and yellow robot ($\mathcal{A}$) and sensor ($\mathcal{S}$) geometries, respectively.



| Path | $\tau^*$ | $\tau_1$ | $\tau_2$ |
|---|---|---|---|
| Targets measured | $\mathcal{T}_3, \mathcal{T}_4, \mathcal{T}_5$ | $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_5$ | $\mathcal{T}_3, \mathcal{T}_6, \mathcal{T}_7$ |
| $D_{tot}$ | 19.46 | 18.78 | 19.32 |
| $B_{tot}$ | 16.75 | 9.66 | 10.06 |
| $\eta_y$ | 0.1542 | 0.0532 | 0.1035 |

**Soil Moisture $s_r$ (%):**

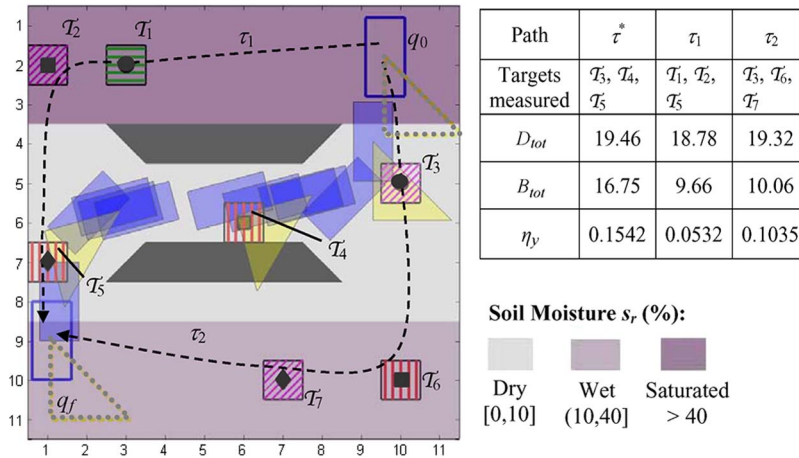| Dry [0,10] | Wet (10,40] | Saturated > 40 |
|---|---|---|

Fig. 11.   Influence of soil moisture $s_r$ on the optimal sensor path $\tau^*$, which is plotted by blue and yellow robot ($\mathcal{A}$) and sensor ($\mathcal{S}$) geometries, respectively, where $\mathcal{T}_1 = \mathcal{T}_3$ (circle), $\mathcal{T}_2 = \mathcal{T}_6$ (square), and $\mathcal{T}_5 = \mathcal{T}_7$ (diamond).

that $\tau^*$ must be planned at the on-set of the sensor motion in order to minimize distance.

Consider another simple example in which there exist two paths $\tau_1$ and $\tau^*$ of approximately equal distance that visit the same number of targets (Fig. 10). Based on prior IR measurements, the EER of individual targets is either high (EER $\geq$ 0.2), medium (0.1 $<$ EER $<$ 0.2), or low (EER $\leq$ 0.1), where the EER is discretized for illustration purposes. If only the targets' locations and geometries were taken into account, $\tau_1$ and $\tau^*$ would be considered equivalent. Instead, by maximizing the measurement benefit (Fig. 10), the optimal path $\tau^*$ obtains a much higher classification improvement than $\tau_1$. The EER in (6) also accounts for the influence of known environmental conditions on the sensor measurements. Consider an example in which the same three targets (represented by the same symbol) are buried in different soils, leading to different values of EER depending on how favorable the conditions are to the GPR sensor (Fig. 11). As before, by visiting targets with higher EER (plotted by the same color-pattern scheme used in Fig. 10), the sensor obtains improved classification efficiency along $\tau^*$, as shown in Fig. 11. Although there exist three paths of approximately equal distance that visit the same three

targets, the EER is highest along $\tau^*$ (Fig. 11), because the fused GPR–IR performance, particularly for the target represented by a diamond, is better in dry soils.

The example in Fig. 12 shows that the platform geometry $\mathcal{A}$ must be accounted for in planning the sensor path. In this workspace, there exist three candidate paths ($\tau_1$, $\tau_2$, and $\tau_3$) that visit targets of approximately equal measurement benefit (Fig. 12). Suppose two robots of different geometries, $\mathcal{A}_1$ and $\mathcal{A}_2$, with the same on-board GPR are deployed in the minefield shown in Fig. 12. The first robot $\mathcal{A}_1$ is small enough to maneuver inside the narrow passage and visit high-EER targets by traveling a shorter distance than $\tau_2$ or $\tau_3$, namely, $D_{tot} = 20.47$ [Fig. 12(a)], whereas $\mathcal{A}_2$ is too large for $\tau_1$ and, thus, must take a longer path $\tau^* = \tau_2$, with $D_{tot} = 27.50$ [Fig. 12(b)]. Finally, the example in Fig. 13 shows that the FOV geometry must be accounted for in planning the sensor path. As shown by $\tau^*$ in Fig. 13(a), the sensor with a larger FOV, $\mathcal{S}_1$, can make measurements over $\mathcal{T}_1$, a target with high EER, without entering the narrow passage. First, the sensor performs a rotation to enable the measurements over $\mathcal{T}_1$ ($\mathcal{S}_1$ intersects $\mathcal{T}_1$). Then, it proceeds to make measurements over $\mathcal{T}_4$ and $\mathcal{T}_5$, and finally it performs a second rotation to reach $q_f$. Instead, for the sensor
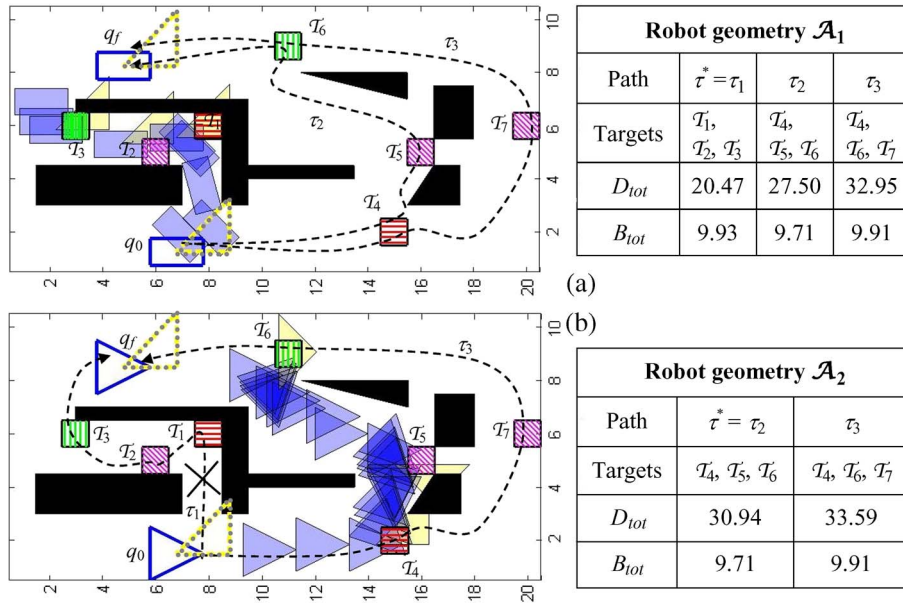
This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

CAI AND FERRARI: INFORMATION-DRIVEN SENSOR PATH PLANNING BY APPROXIMATE CELL DECOMPOSITION 683



| **Robot geometry $\mathcal{A}_1$** | | | |
|---|---|---|---|
| Path | $\tau^* = \tau_1$ | $\tau_2$ | $\tau_3$ |
| Targets | $\mathcal{T}_1$, $\mathcal{T}_2$, $\mathcal{T}_3$ | $\mathcal{T}_4$, $\mathcal{T}_5$, $\mathcal{T}_6$ | $\mathcal{T}_4$, $\mathcal{T}_6$, $\mathcal{T}_7$ |
| $D_{tot}$ | 20.47 | 27.50 | 32.95 |
| $B_{tot}$ | 9.93 | 9.71 | 9.91 |

| **Robot geometry $\mathcal{A}_2$** | | |
|---|---|---|
| Path | $\tau^* = \tau_2$ | $\tau_3$ |
| Targets | $\mathcal{T}_4$, $\mathcal{T}_5$, $\mathcal{T}_6$ | $\mathcal{T}_4$, $\mathcal{T}_6$, $\mathcal{T}_7$ |
| $D_{tot}$ | 30.94 | 33.59 |
| $B_{tot}$ | 9.71 | 9.91 |

Fig. 12. Influence of platform geometry $\mathcal{A}$ on the optimal sensor path $\tau^*$, which is plotted by sample configurations.



| **Sensor geometry $\mathcal{S}_1$** | | | | |
|---|---|---|---|---|
| Path | Targets | $D_{tot}$ | $B_{tot}$ | $\eta_{CL}$ |
| $\tau^*$ | $\mathcal{T}_1, \mathcal{T}_4, \mathcal{T}_5$ | 21.3 | 14.4 | 0.0516 |
| $\tau_1$ | $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3$ | 22.6 | 14.6 | 0.0381 |

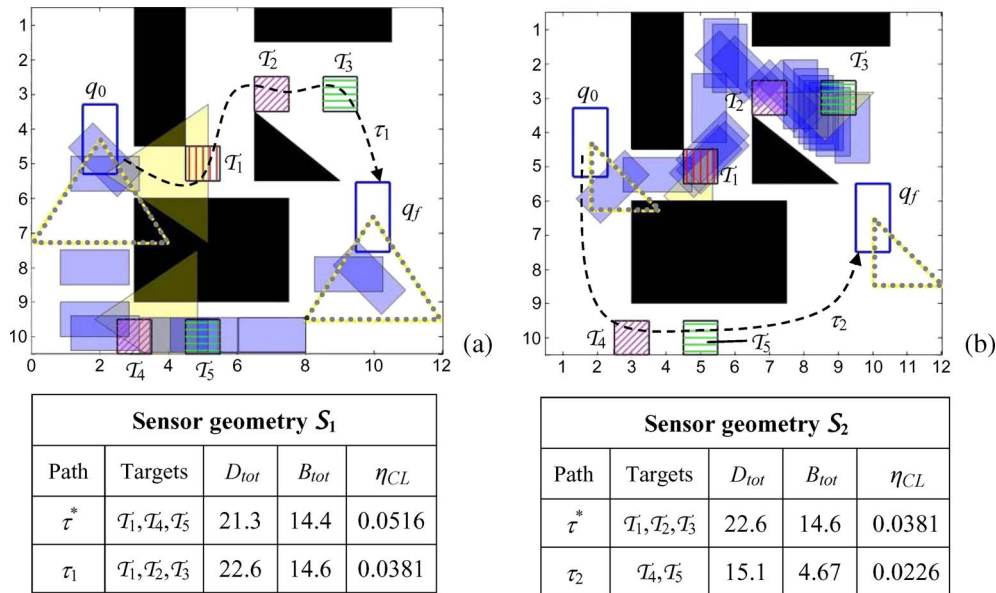| **Sensor geometry $\mathcal{S}_2$** | | | | |
|---|---|---|---|---|
| Path | Targets | $D_{tot}$ | $B_{tot}$ | $\eta_{CL}$ |
| $\tau^*$ | $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3$ | 22.6 | 14.6 | 0.0381 |
| $\tau_2$ | $\mathcal{T}_4, \mathcal{T}_5$ | 15.1 | 4.67 | 0.0226 |

Fig. 13. Influence of FOV geometry $\mathcal{S}$ on the optimal sensor path $\tau^*$, which is plotted by sample sensor configurations.

with FOV $\mathcal{S}_2$, $\tau^*$ is inside the narrow passage because $\mathcal{S}_2$ is too small for the sensor to make measurements over $\mathcal{T}_1$ without entering the passage [Fig. 13(b)], and the measurement benefit of $\tau_2$ is comparatively very low.

### B. Information-Driven Sensor Path Planning Efficiency Results

The sensor path planning method developed in this paper was tested on a variety of minefields with various sizes, geometries, and environmental conditions. A representative example of optimal sensor strategy $\sigma^*$, and corresponding path $\tau^* \subset \sigma^*$, is shown in Fig. 14 for a $40 \times 30$ (bin) minefield with 98 targets, 30 obstacles, and heterogeneous soils, vegetation, weather, and illumination. Here, $\mathcal{S}$ is a triangle that is plotted in yellow

whenever the test decision is to make measurements in one of the available GPR modes (the mode chosen is not shown for simplicity). $\mathcal{A}$ is a rectangle that is plotted in blue at sample configurations along the free path obtained from $\tau^*$. The index $i$ of each target that is measured by the GPR is shown next to the target geometry $\mathcal{T}_i$ in Fig. 14. The EER of the targets is plotted using the notation in Fig. 10 rotated by ninety degrees, and the environmental conditions are not shown for simplicity. This result shows that the sensor avoids obstacles and navigates through narrow passages in order to reach targets with high EER using a minimum distance.

For comparison, shortest path, complete coverage, random search, and grid search strategies are also obtained and implemented for the robotic GPR sensor. Since these methods do not account for the sensor's and targets' geometries [2], [10],
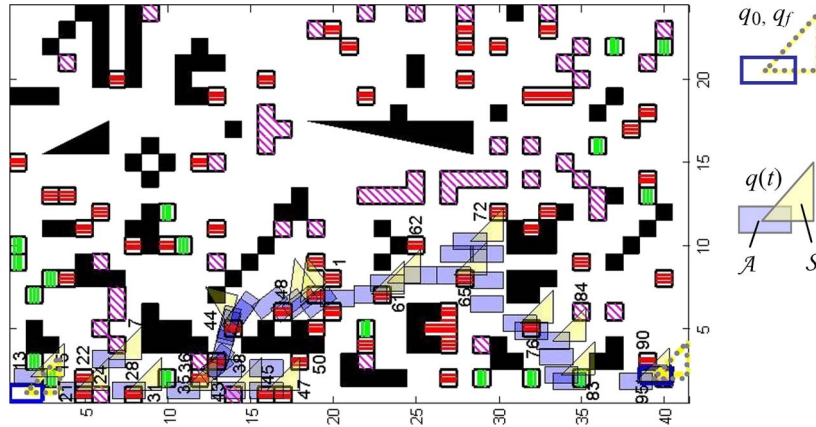
This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

684
IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 39, NO. 3, JUNE 2009

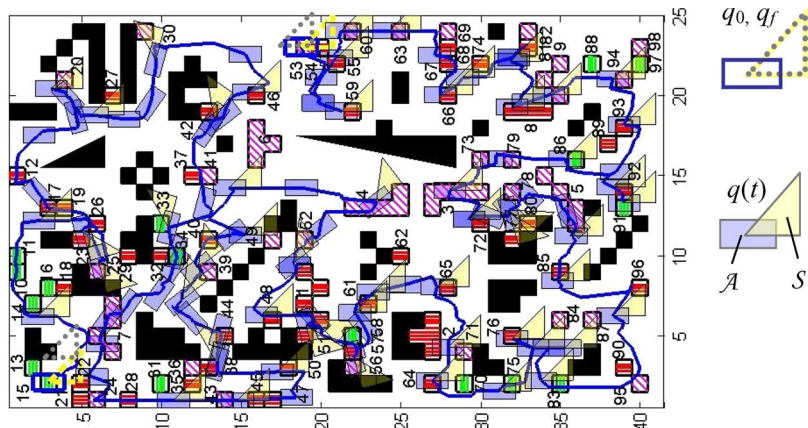Fig. 14. Example of optimal sensor strategy in a minefield with 98 targets and 30 obstacles.



Fig. 15. Complete coverage path $\tau_{\text{cover}}$ for the same minefield used in Fig. 14, with the platform's center-of-gravity trajectory plotted by a blue solid line for emphasis.

[16], they are modified to utilize the connectivity graph $\mathcal{G}$, presented in Section IV-A. Moreover, since these methods do not compute test decisions, the GPR measurements are obtained using a fixed depth-search mode ($v_{\text{GPR}}^1$ in Table I), without energy-saving considerations. After these strategies have been executed, the targets are classified based on fused GPR–IR measurements and environmental information, using the BN sensor models.

The shortest path is a classical robot path planning strategy [41] that here is applied to $\mathcal{G}$ in order to evaluate the efficiency metrics when observation cells are not taken into account. By this approach, $\mathcal{G}$ is searched by the A* algorithm to find the shortest path $\tau_{short}$ from $q_0 \in \kappa_0$ to $q_f \in \kappa_f$. One of the most important results in sensor path planning is a planner that generates a path that completely covers the obstacle-free space (e.g., see [1], [2], [10], [22], and [61]). This method, however, does not address complete coverage when the sensor has a bounded FOV, $\mathcal{S}$, or when there exist prior information about the position and geometry of multiple targets in $\mathcal{W}$. Here, the complete-coverage path $\tau_{\text{cover}}$ is the shortest sensor path that visits all $r$ targets in $\mathcal{W}$, while avoiding collisions with the obstacles. It is obtained by searching $\mathcal{G}$ for the shortest path between all observations cells. Like other complete-coverage path problems [10], computing $\tau_{\text{cover}}$ amounts to solving a traveling salesman problem, which is NP-hard [62]. Therefore, in large minefields,

an approximate solution to $\tau_{\text{cover}}$ is sought by a hierarchical search that divides $\mathcal{C}_{\text{free}}$ into regions in which the shortest path trough all observation cells can be determined by a minimum spanning tree algorithm [63]. By using the final configuration from each region to specify the initial configuration in an adjacent region, $\tau_{\text{cover}}$ is obtained for the entire minefield, as shown by the example in Fig. 15.

A popular approach in robotic demining and UXO clearance is to drive the robot in the minefield using a random search [2]. Random search consists of moving along a straight line until an obstacle is met, and then rotating a random amount before continuing along another straight line, while the on-board sensor is on at all times to detect targets along the path [2]. A more efficient random-search path, $\tau_{\text{rand}}$, can be obtained by randomly selecting a subset of observation cells in $\mathcal{G}$, and then applying the A* algorithm sequentially to find the shortest path between them. Another approach to demining, known as grid search [2], [16], is similarly applied to obtain the shortest path, $\tau_{\text{grid}}$, between a subset of observation cells in $\mathcal{G}$ that are approximately equidistant in $\mathcal{W}$ (on a grid).

All five methods are applied to four minefields of dimensions similar to Fig. 14, but with different targets, obstacles, and environmental conditions. For each method, several paths are computed in every field, and the average efficiency metrics are shown in Table II. It can be seen that the efficiency of the

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

CAI AND FERRARI: INFORMATION-DRIVEN SENSOR PATH PLANNING BY APPROXIMATE CELL DECOMPOSITION 685

TABLE II
AVERAGE SENSOR PATH-PLANNING EFFICIENCY

| Efficiency Metric | Method | | | | |
|---|---|---|---|---|---|
| | Optimal Strategy, $\sigma^*$ | Shortest Path ($\sigma^*$ Improvement) | Complete Coverage ($\sigma^*$ Improvement) | Random Search ($\sigma^*$ Improvement) | Grid Search ($\sigma^*$ Improvement) |
| $\eta_N$ | 0.4610 | 0.3053 (51.0%) | 0.2683 (71.8%) | 0.1441 (219.9%) | 0.2321 (98.6%) |
| $\eta_y$ | 0.0595 | 0.0407 (46.2%) | 0.0055 (981.8%) | 0.0114 (421.9%) | 0.0122 (387.7%) |
| $\eta_{CL}$ | 0.0446 | 0.0157 (184.1%) | 0.0153 (191.5%) | 0.0098 (355.1%) | 0.0133 (235.3%) |
| $\eta_H$ | 0.0599 | 0.0330 (81.5%) | 0.0410 (46.1%) | 0.0244 (145.5%) | 0.0343 (74.6%) |
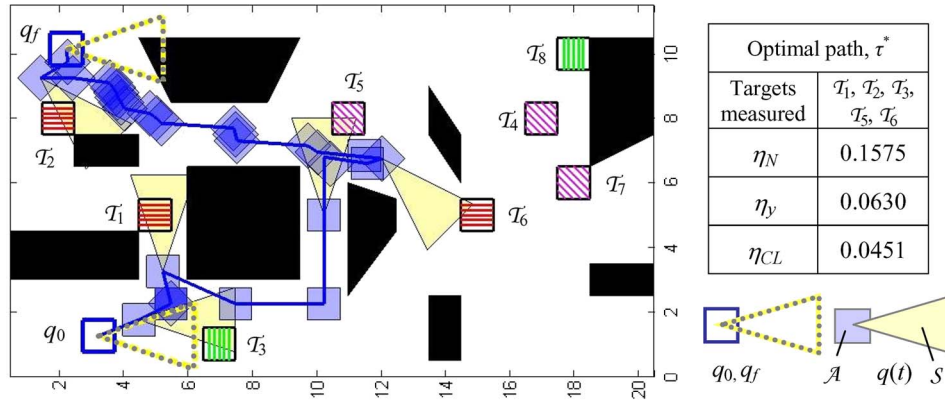


Fig. 16. Example of optimal sensor strategy for a nonoverpass capable platform ($\mathcal{A}$) that must avoid targets as well as obstacles, with the platform's center-of-gravity trajectory plotted in a blue solid line.
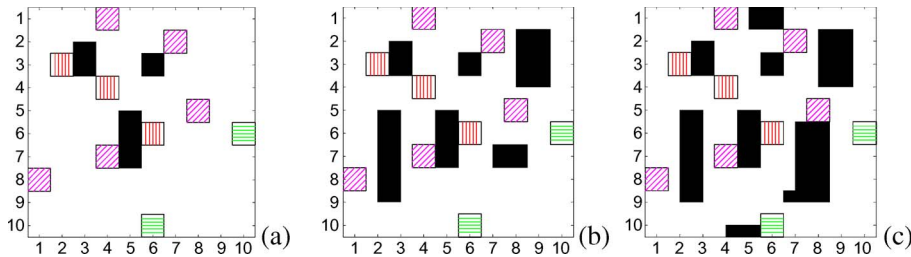


Fig. 17. Illustrative examples of (a) low, (b) medium, (c) and high obstacle-density workspaces.

optimal strategy $\sigma^*$ surpasses that of all other methods by as much as 981.8% (Table II). Although $\sigma^*$ obtains the highest efficiency for each metric, $\eta_y$ displays the highest improvement (Table II). The method with the worst classification improvement is complete coverage, because the robotic sensor travels a long distance to reach targets that have low measurement benefit. Under the other metrics, the method with the worst performance is random search, probably because it implements the worst criterion for measurement selection.

Another important application of the approximate-and-decompose method is planning the path of nonoverpass capable robotic sensors that can be seriously damaged or even destroyed when driving over landmines [12]. In this case, a sensor path that avoids collisions with both obstacles and targets is obtained by including an additional set of $r$ obstacles with the same geometries as the targets, to prevent intersections between $\mathcal{A}$ and $\mathcal{T}_1, \ldots, \mathcal{T}_r$. As shown in Fig. 16, the robotic sensor is still able to obtain measurements over targets with high
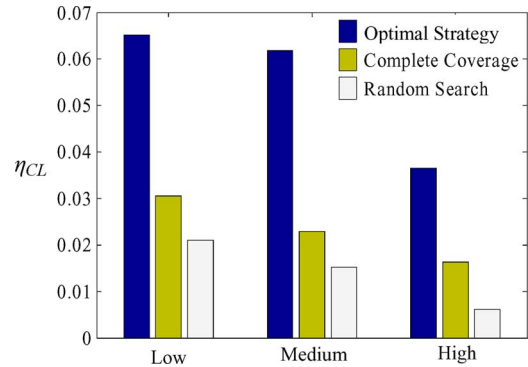


Fig. 18. Average error reduction for minefields with low, medium, and high obstacle-density workspaces.

EER (plotted using the color-pattern scheme in Fig. 10), while its platform avoids all targets previously detected by the IR sensor. As can be expected, the average efficiencies are lower
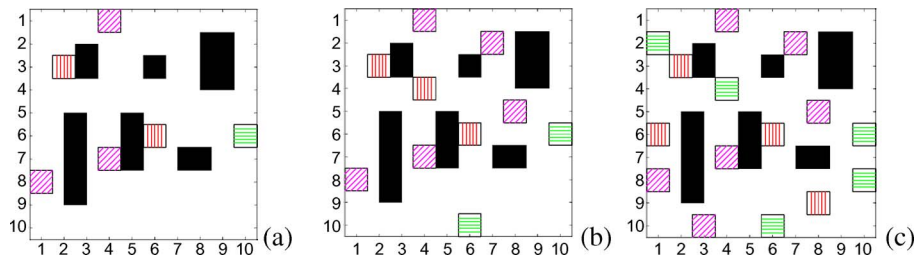
Fig. 19.    Illustrative examples of (a) low, (b) medium, (c) and high target-density workspaces.
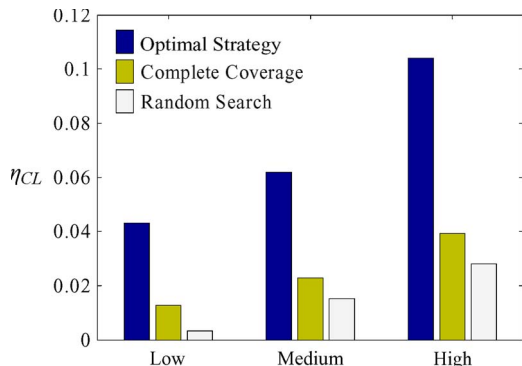


Fig. 20.    Average error reduction for minefields with low, medium, and high target-density workspaces.

than those obtained by overpass-capable platforms (Table II), because the sensor travels a longer distance in order to avoid collisions with the targets. For example, in Fig. 16 a narrow passage suddenly is unavailable because $\mathcal{T}_1$ is located inside it. However, the sensor is still able to measure $\mathcal{T}_1$ by rotating to intersect it with $\mathcal{S}$, and then proceeding to navigate through another mine-free narrow passage.

Since the method's efficiency depends on the characteristics of the workspace, extensive numerical simulations were performed to obtain average efficiency metrics for different densities of obstacles and targets. Fig. 17 shows three representative examples of workspaces that are considered to have low (a), medium (b), and high (c) obstacle density. The same (medium) target density is used in all minefields considered in this study. As shown in Fig. 18, for every level of obstacle density, the average error reduction obtained by the proposed method is significantly higher than that obtained by complete coverage and random search methods. Similar results are obtained for the other efficiency metrics (Section V-B). The smallest difference between the methods is found in high obstacle-density minefields, because they exhibit fewer feasible paths. Nevertheless, in these minefields the connectivity graph developed in Section IV-A is needed to find any feasible sensor path, including $\tau_{\text{cover}}$ and $\tau_{\text{rand}}$.

In another study, average efficiency metrics are computed for minefields with low, medium, and high target densities, as shown by three simple examples in Fig. 19, using the same (medium) obstacle density. As shown in Fig. 20, the average error reduction of the proposed method is significantly higher than that of complete coverage and random search methods for all levels of target density. The most significant improvement is obtained in minefields with high target density, because the sensor is able to visit more targets, taking full advantage of

the proposed methodology. Similar results were obtained for the other efficiency metrics (Section V-B), as well as for different environmental conditions, but they are omitted here for brevity. From these studies, it can be concluded that the efficiency of the optimal strategy $\sigma^*$ is significantly higher than that of the other approaches under a wide range of workspace conditions and characteristics.

## VII. SUMMARY AND CONCLUSION

An approximate cell decomposition approach to robotic sensor path planning is developed by treating geometric targets as the dual of obstacles, and by specifying the position and orientation of the sensor's platform and FOV with respect to the same moving Cartesian frame. The keystones of this novel approach are the concept of observation cells, and a new approximate-and-decompose procedure for constructing the connectivity graph. Thanks to this procedure, each observation cell enables measurements over a unique subset of targets in $\mathcal{W}$, such that their measurement profit can be estimated *a priori*, and accounted for in the search for the optimal path. Using a label-correcting pruning algorithm previously developed by the authors, the connectivity graph is pruned and transformed into a decision tree that is used to compute an optimal sensing strategy, including the sensor's motion, mode, and measurement sequence. The methodology is demonstrated by computing the sensing strategy of a robotic GPR sensor based on prior airborne IR measurements and environmental information. The numerical results show that the optimal strategy outperforms shortest path, complete coverage, random, and grid searches in ROIs exhibiting low-to-high densities of targets, obstacles and narrow passages, and nonuniform soils, weather, and vegetation. The method also presents other advantages, such as the applicability to nonoverpass capable platforms, and the construction of a connectivity graph that can be used to develop complete coverage, random, and grid strategies that account for the geometric properties of the robotic sensor and the workspace.

## APPENDIX I
## PROPERTIES OF APPROXIMATE CELL DECOMPOSITION IN THE PRESENCE OF TARGETS

This section analyzes the properties of the approximate cell decomposition presented in Section IV-A.

*Property 1:* For $\forall q \in \kappa^u$, if $q \notin \mathcal{RB}_j[\kappa^u]$, then $\mathcal{A}$ avoids collisions with $\mathcal{B}_j$, and if $q \in \mathcal{RT}_i[\kappa^u]$, then $\mathcal{S}$ can make measurements over $\mathcal{T}_i$.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

CAI AND FERRARI: INFORMATION-DRIVEN SENSOR PATH PLANNING BY APPROXIMATE CELL DECOMPOSITION 687

*Proof:* It follows from Definitions 4.5–4.6, that $\mathcal{CB}_j[\kappa^u] \subseteq \mathcal{OCB}_j[\kappa^u] \times \mathcal{I}_u \subseteq \mathcal{RB}_j[\kappa^u]$ and $\mathcal{R'T}_i[\kappa^u] \subseteq \mathcal{ICT}_i[\kappa^u] \times \mathcal{I}_u \subseteq \mathcal{CT}_i[\kappa^u]$, therefore $\mathcal{CB}_j[\kappa^u] \subseteq \mathcal{RB}_j[\kappa^u]$ and $\mathcal{R'T}_i[\kappa^u] \subseteq \mathcal{CT}_i[\kappa^u]$. Then, any configuration value $q^\ell \notin \mathcal{RB}_j[\kappa^u]$ also obeys $q^\ell \notin \mathcal{CB}_j[\kappa^u]$, and by the definition of C-obstacles $\mathcal{A}(q^\ell) \cap \mathcal{B}_j = \varnothing$. Therefore, by excluding bounding rectangloid approximations of C-obstacles from $\mathcal{K}$, $\mathcal{A}$ avoids collisions with the obstacles. On the other hand, any configuration value $q^k \in \mathcal{R'T}_i[\kappa^u]$ also obeys $q^k \in \mathcal{CT}_i[\kappa^u]$. Thus, by Definition 4.2, $\mathcal{S}(q^k) \cap \mathcal{T}_i \neq \varnothing$, and any cell obtained from $\mathcal{R'T}_i[\kappa^u] \setminus \bigcup_{j=1}^n \mathcal{RB}_j[\kappa^u]$ enables measurements by $\mathcal{S}$ over $\mathcal{T}_i$, while avoiding collisions between $\mathcal{A}$ and the obstacles. ∎

*Property 2:* If $\mathcal{A}$, $\mathcal{S}$, $\mathcal{B}_j$ $(j = 1, \ldots, n)$, and $\mathcal{T}_i$ $(i = 1, \ldots, r)$ are all convex polygons in an Euclidean workspace $\mathcal{W} \subset \mathbb{R}^2$, the time complexity of the approximate cell decomposition in the presence of targets (Section IV-A) is $\mathcal{O}((n_\mathcal{B} + n_\mathcal{T})^2)$, where $n_\mathcal{B}$ is the number of edges of all $n$ obstacles, and $n_\mathcal{T}$ is the number of edges of all $r$ targets.

*Proof:* If $\mathcal{A}$, $\mathcal{S}$, $\mathcal{B}_j(j = 1, \ldots, n)$, and $\mathcal{T}_i(i = 1, \ldots, r)$ are all convex polygons in $\mathcal{W} \subset \mathbb{R}^2$, then $\mathcal{CB}_j(j = 1, \ldots, n)$ and $\mathcal{CT}_i(i = 1, \ldots, r)$ are also convex [41]. Let $n_\mathcal{A}$ denote the number of edges of $\mathcal{A}$. In Steps 1)–2), $\mathcal{CB}_j[\kappa^u]$, $j = 1, \ldots, n$ and $\mathcal{CT}_i[\kappa^u]$, $i = 1, \ldots, r$ for every $u = 1, \ldots, \nu$ are computed by first discretizing $\mathcal{I}_u$ into $k_u$ values, with $\gamma_u + l\Delta\theta$ for $0 \leq l \leq k_u$, and $\Delta\theta = (\gamma_{u+1} - \gamma_u)/k_u$. This strategy converts the computation of $\mathcal{CB}_j[\kappa^u]$ and $\mathcal{CT}_i[\kappa^u]$ in a 3-D world to the computation of C-obstacles and C-targets in a 2-D world at different robot orientations (e.g., imagine $\mathcal{CB}_j[\kappa^u]$ and $\mathcal{CT}_i[\kappa^u]$ as a stack of $k_u$ polygonal regions). The time for computing the 2-D C-obstacles, $\mathcal{CB}_j[[x_\kappa, x'_\kappa] \times [y_\kappa, y'_\kappa] \times (\gamma_u + l\Delta\theta)]$ for $j = 1, \ldots, n$, at the robot orientations $(\gamma_u + l\Delta\theta)$ for $l = 0, 1, \ldots, k_u$ is $\mathcal{O}(n_\mathcal{B})$. The time for computing $\mathcal{CT}_i[[x_\kappa, x'_\kappa] \times [y_\kappa, y'_\kappa] \times (\gamma_u + l\Delta\theta)]$ for $i = 1, \ldots, r$, at the robot orientations $(\gamma_u + l\Delta\theta)$ for $l = 0, 1, \ldots, k_u$ is $\mathcal{O}(n_\mathcal{T})$ [64]. Since $k_u$ is a constant, the time for computing $\mathcal{CB}_j[\kappa^u]$ for $j = 1, \ldots, n$ is $\mathcal{O}(n_\mathcal{B})$. Moreover, the time for computing $\mathcal{CT}_i[\kappa^u]$ for $i = 1, \ldots, r$ is $\mathcal{O}(n_\mathcal{T})$. Based on the numerical approach for computing $\mathcal{CB}_j$ and $\mathcal{CT}_i$

$$\mathcal{OCB}_j[\kappa^u] = \bigcup_{l=1}^{k_u} \{\mathcal{CB}_j[[x_\kappa, x'_\kappa] \times [y_\kappa, y'_\kappa] \times (\gamma_u + l\Delta\theta)]\} \tag{17}$$

$$\mathcal{ICT}_i[\kappa^u] = \bigcup_{l=1}^{k_u} \{\mathcal{CT}_i[[x_\kappa, x'_\kappa] \times [y_\kappa, y'_\kappa] \times (\gamma_u + l\Delta\theta)]\} \tag{18}$$

for every $u = 1, \ldots, \nu$. Thus, the time for computing $\mathcal{OCB}_j[\kappa^u]$ in (17) is $\mathcal{O}((n_\mathcal{A} + c_j) \log(n_\mathcal{A} + c_j))$, where $c_j$ is the number of edges of $\mathcal{B}_j$ [65], and $\sum_{j=1}^n c_j = n_\mathcal{B}$. Since this approximate cell decomposition is resolution complete, it can be assumed that the number of orthogonal rectangloids in $\mathcal{RB}_j[\kappa^u]$ and $\mathcal{R'T}_i[\kappa^u]$ is linear in the number of edges of $\mathcal{B}_j$ and $\mathcal{T}_i$, respectively. Then, the time for generating $\mathcal{RB}_j[\kappa^u]$ for $\mathcal{OCB}_j[\kappa^u] \times \mathcal{I}_u$ [Step 3)] is $\mathcal{O}(c_j)$. Moreover, the time for generating

$\mathcal{RB}_j[\kappa^u]$ for $j = 1, \ldots, n$ is $\sum_{j=1}^n \{\mathcal{O}((n_\mathcal{A} + c_j) \log(n_\mathcal{A} + c_j)) + \mathcal{O}(c_j)\} \leq \mathcal{O}(n_\mathcal{B} \log n_\mathcal{B})$. By the same process, it can be shown that the time for generating $\mathcal{R'T}_i[\kappa^u]$ for $i = 1, \ldots, r$ [Step 3)] is $\mathcal{O}(n_\mathcal{T} \log n_\mathcal{T})$.

In Step 4), the decomposition of $\mathcal{C}^u_{\text{void}}$ into nonoverlapping rectangloids can be performed as the 2-D vertical decomposition presented in [64]. Thus, it can be carried out in time $\mathcal{O}((n_\mathcal{B} + n_\mathcal{T}) \log(n_\mathcal{T} + n_\mathcal{T}))$ [64]. Step 5) is comprised of two stages. The first stage decomposes $\mathcal{R'T}_i[\kappa^u] \setminus \{\bigcup_{j=1}^n \mathcal{RB}_j[\kappa^u] \cup \bigcup_{l=1,l\neq i}^r \mathcal{R'T}_l[\kappa^u]\}$ into cells from which only one target is observable. Moreover, the second stage decomposes $\mathcal{R'T}_i[\kappa^u] \cap \bigcup_{l=1,l\neq i}^r \mathcal{R'T}_l[\kappa^u]$ into cells from which two or more targets are observable. The time required by the first stage is $\mathcal{O}(n_\mathcal{B} + n_\mathcal{T})$, because convex polygons are characterized by the property that the boundaries of any pair intersect at most two points. Therefore, the time complexity of the common exterior is linear in the number of polygons' edges $(n_\mathcal{B} + n_\mathcal{T})$ [65]. Moreover, it can be easily shown that $\mathcal{R'T}_i[\kappa^u] \setminus \{\bigcup_{j=1}^n \mathcal{RB}_j[\kappa^u] \cup \bigcup_{l=1,l\neq i}^r \mathcal{R'T}_l[\kappa^u]\}$ is an orthogonal polygon without holes, since $\mathcal{RB}_j[\kappa^u]$ and $\mathcal{R'T}_i[\kappa^u]$ are connected for $\forall i, j$, and never contain each other by definition. The time for optimally partitioning an orthogonal polygon without holes into the minimum number of rectangloids is linear in the number of edges [66]. It follows that the first stage requires a time $\mathcal{O}(n_\mathcal{T}(n_\mathcal{B} + n_\mathcal{T}))$, and the second stage requires a time $\mathcal{O}(n_\mathcal{T}^2)$. It can be concluded that the entire approximate cell decomposition procedure requires a time $\mathcal{O}(n_\mathcal{B}) + \mathcal{O}(n_\mathcal{T}) + \mathcal{O}(n_\mathcal{B} \log n_\mathcal{B}) + \mathcal{O}(n_\mathcal{T} \log n_\mathcal{T}) + \mathcal{O}((n_\mathcal{B} + n_\mathcal{T}) \log(n_\mathcal{T} + n_\mathcal{T})) + \mathcal{O}(n_\mathcal{T}(n_\mathcal{B} + n_\mathcal{T})) + \mathcal{O}(n_\mathcal{T}^2) \leq \mathcal{O}((n_\mathcal{B} + n_\mathcal{T})^2)$, or simply $\mathcal{O}((n_\mathcal{B} + n_\mathcal{T})^2)$. ∎

## APPENDIX II

**Algorithm 1** Pseudocode for Transformation $T_r \Rightarrow DT$
add root $\kappa_0$
**for** every branch in $T_r$ **do**
    $\kappa_i = \kappa_0$
    **for** $t_k = t_1, \ldots, t_f$ **do**
        add new chance node $\kappa_j$ from column $t_k$ in $T_r$
        add arc $(\kappa_i, \kappa_j)$ and label it with corresponding action
        decision $a(t_k)$
        **if** $\kappa_j$ is an observation cell **then**
            add decision node $u(t_k)$ and arc $(\kappa_j, u(t_k))$
            label arc $(\kappa_j, u(t_k))$ with value $\arg\max\{w_B \cdot B[Z(t_k)] - w_J \cdot J[u(t_k)]\}$
        **end if**
        $\kappa_i = \kappa_j$
    **end for**
add a leaf with the total expected measurement profit $V$
**end for**
merge branches from $\kappa_0$ up to the last common node

## REFERENCES

[1] C. Hofner and G. Schmidt, "Path planning and guidance techniques for an autonomous mobile cleaning robot," *Robot. Auton. Syst.*, vol. 14, no. 3, pp. 199–212, Aug. 1995.

[2] E. U. Acar, "Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods," *Int. J. Robot. Res.*, vol. 22, no. 7/8, pp. 441–466, Jul. 2003.

[3] C. Kreucher, K. Kastella, and A. Hero, "Multi-platform information-based sensor management," in *Proc. SPIE Defense Transformation Netw.-Centric Syst. Symp.*, Orlando, FL, 2005, vol. 5820, pp. 141–151.

[4] N. Rao, S. Hareti, W. Shi, S. Iyengar, "Robot navigation in unknown terrains: Introductory survey of non-heuristic algorithms," in *Tech. Rep. ORNL/TM-12410*, Oak Ridge Nat. Lab., Oak Ridge, TN, 1993.

[5] N. Rao, "Robot navigation in unknown generalized polygonal terrains using vision sensors," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, no. 6, pp. 947–962, Jun. 1995.

[6] A. Lazanas and J. C. Latombe, "Motion planning with uncertainty: A landmark approach," *Artif. Intell.*, vol. 76, no. 1, pp. 287–317, Jul. 1995.

[7] K. Song and C. C. Chang, "Reactive navigation in dynamic environment using a multisensor predictor," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 6, pp. 870–880, Dec. 1999.

[8] Z. Sun and J. Reif, "On robotic optimal path planning in polygonal regions with pseudo-Euclidean metrics," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 37, no. 4, pp. 925–936, Aug. 2007.

[9] X.-C. Lai, S.-S. Ge, and A. Al-Mamun, "Hierarchical incremental path planning and situation-dependent optimized dynamic motion planning considering accelerations," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 37, no. 6, pp. 1541–1554, Dec. 2007.

[10] H. Choset, "Coverage for robotics—A survey of recent results," *Ann. Math. Artif. Intell.*, vol. 31, no. 1–4, pp. 113–126, Oct. 2001.

[11] C. Cai and S. Ferrari, "A Q-learning approach to developing an automated neural computer player for the board game of CLUE," in *Proc. Int. Joint Conf. Neural Netw.*, Hong Kong, 2008, pp. 2347–2353.

[12] R. Siegel, "Land mine detection," *IEEE Instrum. Meas. Mag.*, vol. 5, no. 4, pp. 22–28, Dec. 2002.

[13] S. Ferrari, C. Cai, R. Fierro, and B. Perteet, "A multi-objective optimization approach to detecting and tracking dynamic targets in pursuit-evasion games," in *Proc. Amer. Control Conf.*, New York, 2007, pp. 5316–5321.

[14] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, Aug. 2004.

[15] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, "Energy efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet," in *Proc. 10th Int. Conf. ASPLOS*, San Jose, CA, 2002, pp. 96–107.

[16] X. Liao and L. Carin, "Application of the theory of optimal experiments to adaptive electromagnetic-induction sensing of buried targets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 961–972, Aug. 2004.

[17] J. R. Spletzer and C. J. Taylor, "Dynamic sensor planning and control for optimally tracking target," *Int. J. Robot. Res.*, vol. 22, no. 1, pp. 7–20, Jan. 2003.

[18] G. D. Hager and M. Mintz, "Computational methods for task-directed sensor data fusion and sensor planning," *Int. J. Robot. Res.*, vol. 10, no. 4, pp. 285–313, Aug. 1991.

[19] S. Y. Chen and Y. F. Li, "Automatic sensor placement for model-based robot vision," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 1, pp. 393–408, Feb. 2004.

[20] S. Y. Chen and Y. F. Li, "Vision sensor planning for 3-D model acquisition," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 5, pp. 894–904, Oct. 2005.

[21] E. Gelenbe and Y. Cao, "Autonomous search for mines," *Eur. J. Oper. Res.*, vol. 108, no. 2, pp. 319–333, Jul. 1998.

[22] M. Qian and S. Ferrari, "Probabilistic deployment for multiple sensor systems," in *Proc. 12th SPIE Symp. Smart Struct. Mater., Sensors Smart Struct. Technol. Civil, Mech. Aerosp. Syst.*, San Diego, CA, 2005, vol. 5765, pp. 85–96.

[23] D. Zhu and J.-C. Latombe, "New heuristic algorithms for efficient hierarchical path planning," *IEEE Trans. Robot. Autom.*, vol. 7, no. 1, pp. 9–20, Feb. 1991.

[24] K. Kastella, "Discrimination gain to optimize detection and classification," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 27, no. 1, pp. 112–116, Jan. 1997.

[25] C. Kreucher, K. Kastella, and A. Hero, "Sensor management using an active sensing approach," *Signal Process.*, vol. 85, no. 3, pp. 607–624, Mar. 2005.

[26] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 61–72, Mar. 2002.

[27] W. Schmaedeke, "Information based sensor management," in *Proc. 2nd SPIE Signal Process., Sensor Fusion, Target Recog.*, Orlando, FL, 1993, vol. 1955, pp. 156–164.

[28] S. Ji, R. Parr, and L. Carin, "Nonmyopic multiaspect sensing with partially observable Markov decision processes," *IEEE Trans. Signal Process.*, vol. 55, no. 1, pp. 2720–2730, Jun. 2007.

[29] C. Cai and S. Ferrari, "Comparison of information-theoretic objective functions for decision support in sensor systems," in *Proc. Amer. Control Conf.*, New York, 2007, pp. 63–133.

[30] D. J. Daniels, "A review of GPR for landmine detection," *Int. J. Sens. Imaging*, vol. 7, no. 3, pp. 90–123, Sep. 2006.

[31] J. MacDonald, *Alternatives for Landmine Detection*. Santa Monica, CA: RAND, 2003.

[32] F. V. Jensen, *Bayesian Networks and Decision Graphs*. Berlin, Germany: Springer-Verlag, 2001.

[33] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. I and II. Belmont, MA: Athena Scientific, 1995.

[34] S. Ferrari and C. Cai, "Information-driven search strategies in the board game of CLUE," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 3, Jun. 2009. to be published.

[35] S. Perrin, E. Duflos, P. Vanheeghe, and A. Bibaut, "Multisensor fusion in the frame of evidence theory for landmines detection," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 34, no. 4, pp. 485–498, Nov. 2004.

[36] S. Ferrari and A. Vaghi, "Demining sensor modeling and feature-level fusion by Bayesian networks," *IEEE Sensors J.*, vol. 6, no. 2, pp. 471–483, Apr. 2006.

[37] C. Cai, S. Ferrari, and Q. Ming, "Bayesian network modeling of acoustic sensor measurements," in *Proc. IEEE Sensors*, Atlanta, GA, 2007, pp. 345–348.

[38] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," *Mach. Learn.*, vol. 20, no. 3, pp. 197–243, Sep. 1995.

[39] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[40] K. Murphy, *How to Use the Bayes Net Toolbox*, 2004. [Online]. Available: http://www.ai.mit.edu/ murphyk/Software/BNT/bnt.html

[41] J. C. Latombe, *Robot Motion Planning*. Norwell, MA: Kluwer, 1991.

[42] J. Schwartz and M. Sharir, *On the 'Piano Movers' Problem: II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds*. New York: Academic, 1983.

[43] K. Kedem and M. Sharir, "An efficient motion planning algorithm for convex polygonal object in 2-dimensional polygonal space," Courant Inst. Math. Sci., New York, Tech. Rep. 253.

[44] F. Avnaim, J. D. Boissonnat, and B. Faverjon, "A practical motion planning algorithm for polygonal objects amidst polygonal obstacles," INRIA, Sophia-Antipolis, France, Tech. Rep. 890.

[45] R. A. Brooks and T. Lozano-Pérez, "A subdivision algorithm in configuration space for findpath with rotation," in *Proc. 8th Int. Joint Conf. Artif. Intell.*, Karlsruhe, Germany, 1983, pp. 799–806. FRG.

[46] B. Faverjon, "Object level programming of industrial robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Francisco, CA, 1986, pp. 1406–1412.

[47] S. Kambhampati and L. S. Davis, "Multiresolution path planning for mobile robots," *IEEE J. Robot. Autom.*, vol. RA-2, no. 3, pp. 135–145, Sep. 1986.

[48] C. Laugier and F. Germain, "An adaptive collision-free trajectory planner," in *Proc. Int. Conf. Adv. Robot.*, Tokyo, Japan, 1985.

[49] Y. K. Hwang and P. C. Chen, "A heuristic and complete planner for the classical mover's problem," in *Proc. IEEE Int. Conf. Robot. Autom.*, Nagoya, Japan, 1995, pp. 729–736.

[50] G. Conte and R. Zulli, "Hierarchical path planning in a multi-robot environment with a simple navigation function," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, no. 4, pp. 651–654, Apr. 1995.

[51] T. Siméon, S. Leroy, and J. P. Laumond, "Path coordination for multiple mobile robots: A resolution-complete algorithm," *IEEE Trans. Robot. Autom.*, vol. 18, no. 1, pp. 42–49, Feb. 2002.

[52] S. X. Yang and C. Luo, "A neural network approach to complete coverage path planning," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 34, no. 1, pp. 718–725, Feb. 2004.

[53] B. Grünbaum, *Convex Polytopes*. New York: Wiley-Interscience, 1967.

[54] R. E. Bellman and S. E. Dreyfus, *Applied Dynamic Programming*. Princeton, NJ: Princeton Univ. Press, 1962.

[55] R. V. Dam, B. Borchers, J. Hendrickx, and S. Hong, "Soil effects on thermal signatures of buried nonmetallic landmines," in *Proc. 8th SPIE Detection Remediation Technol. Mines, Minelike Targets*, vol. 5089, pp. 1210–1218.

[56] "Explosive, Ordnance, Disposal, (EOD), and Technicians," *ORDATA*, 2006. [Online]. Available: http://maic.jmu.edu/ordata/mission.asp

[57] A. Vaghi, "Sensor management by a graphical model approach," M.S. thesis, Politecnico di Milano, Milan, Italy, 2004.

[58] J. Paik, "Image processing-based mine detection techniques using multiple sensors: A review," *Int. J. Subsurf. Sens. Technol. Appl.*, vol. 3, no. 3, pp. 203–252, 2002.

[59] R. Fierro and F. Lewis, "Control of a nonholonomic mobile robot using neural networks," *IEEE Trans. Neural Netw.*, vol. 9, no. 4, pp. 589–600, Jul. 1998.

[60] D. Cruz, J. McClintock, B. Perteet, O. Orqueda, Y. Cao, and R. Fierro, "Decentralized cooperative control—A multivehicle platform for research in networked embedded systems," *IEEE Control Syst. Mag.*, vol. 27, no. 3, pp. 58–78, Jun. 2007.

[61] E.-M. Arkin, S.-P. Fekete, and J.-S.-B. Mitchell, "Approximation algorithms for lawn mowing and milling," *Comput. Geom.*, vol. 17, no. 1, pp. 25–50, Oct. 2000.

[62] E. Lawer, J. Lenstra, A. Kan, and D. B. Shmoys, Eds., *The Traveling Salesman Problem*. New York: Wiley, 1985.

[63] J. B. Krustal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proc. Amer. Math. Soc.*, vol. 7, no. 1, pp. 48–50, Feb. 1956.

[64] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[65] K. Kedem, R. Livne, J. Pach, and M. Sharir, "On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles," *Discrete Comput. Geom.*, vol. 1, no. 1, pp. 59–71, Dec. 1986.

[66] W.-T. Liou, J.-J.-M. Tan, and R. Lee, "Minimum partitioning simple rectilinear polygons in O(nloglogn)-time," in *Proc. 5th Annu. ACM Symp. Comput. Geom.*, Saarbrüchen, Germany, 1989, pp. 344–353.

**Chenghui Cai** (S'02–M'09) received the B.S. and M.S. degrees from Tsinghua University, Beijing, China, in 2000 and 2003, respectively, and the Ph.D. degree from Duke University, Durham, NC, in 2008.

He is currently a Postdoctoral Research Associate of electrical and computer engineering with Duke University. His research interests include robotic sensor planning and management, machine learning and data mining, multiagent systems, Bayesian statistics, decision making under uncertainty, and computational intelligence in games.

Dr. Cai is a member of the American Society of Mechanical Engineers, Sigma Xi, and the Society for Industrial and Applied Mathematics.

**Silvia Ferrari** (S'01–M'02–SM'08) received the B.S. degree from Embry-Riddle Aeronautical University, Daytona Beach, FL, in 1997 and the M.A. and Ph.D. degrees from Princeton University, Princeton, NJ, in 1999 and 2002, respectively.

She is Assistant Professor of mechanical engineering and materials science with Duke University, Durham, NC, where she directs the Laboratory for Intelligent Systems and Controls. Her principal research interests include robust adaptive control of aircraft, learning and approximate dynamic programming, and optimal control of mobile sensor networks.

Dr. Ferrari is a member of the American Society of Mechanical Engineers, The International Society for Optical Engineers, and the American Institute of Aeronautics and Astronautics. She is the recipient of the Office of Naval Research Young Investigator Award in 2004, the National Science Foundation CAREER Award in 2005, and the Presidential Early Career Award for Scientists and Engineers Award in 2006.