2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Sensor Placement Algorithms for Triangulation Based Localization

Onur Tekdas and Volkan Isler

Abstract— **Robots operating in a workspace can localize themselves by querying nodes of a sensor-network deployed in the same workspace. This paper addresses the problem of computing the minimum number and placement of sensors so that the localization uncertainty at every point in the workspace is less than a given threshold. We focus on triangulation based state estimation where measurements from two sensors must be combined for an estimate.**

We show that the general problem for arbitrary uncertainty models is computationally hard. For the general problem, we present a solution framework based on integer linear programming and demonstrate its practical feasibility with simulations. Finally, we present an approximation algorithm for a geometric uncertainty measure which simultaneously addresses occlusions, angle and distance constraints.

I. INTRODUCTION

A sensor network is a network of small, cheap devices equipped with sensing, communication and computation capabilities. With concurrent advances in robotics, embedded sensing, computation and communication technologies, sensor networks are becoming increasingly popular in automation applications such as surveillance, inventory control and traffic management.

The presence of a sensor-network in a robot's workspace can provide robust, scalable solutions to a number of fundamental robotics problems. For example, robots can localize themselves by querying the nodes of a network. In addition to localization, sensors can assist in other robot tasks such as navigation and search.

In the present work, we address the problem of placing sensors so that when a robot queries sensors to estimate its own position, the uncertainty in the position estimation is small. We focus on triangulation-based localization where two sensors are needed for estimating the position of the robot. A good example of this scenario is a robot localizing itself in a camera network. As is well known, a robot cannot localize itself with a single measurement from a single camera. At least two different camera measurements are required for triangulation. However, the quality of the localization is a function of the robot-camera geometry. We consider a scenario where the location of the cameras are known apriori to the robot. To localize itself, the robot queries two cameras and merges their measurements. The problem we address is: *given the workspace and an error threshold, what is the minimum number, and placement of cameras so that the error in localization is less than the threshold at every point in the workspace?*

The authors are with the Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180, USA. Emails: {tekdao,isler}@cs.rpi.edu

In this paper, we build up our previous work on sensor-placement [1] where we focused on a specific uncertainty function: If the robot, located at position x queries cameras located at s_1 and s_2, the uncertainty in the estimation is proportional to $\frac{d(s_1,x)\times d(s_2,x)}{\sin \angle c_1 x c_2}$. In our earlier work, we presented an approximation algorithm for this uncertainty model that deviates from the optimal solution only by a constant factor both in the number of cameras used and the uncertainty in localization. However, in our previous work the issue of visual-occlusions in the workspace was not addressed (Equivalently, it was assumed that there were no obstacles in the workspace.). In this paper, we extend our previous results in the following directions:

- We show, via a simple reduction, that the general placement problem is NP-Complete.
- We present a general framework based on integer linear programming (ILP) which can be used to solve the placement problem for arbitrary uncertainty models while incorporating sensing constraints such as occlusions. We demonstrate the practical feasibility of this approach through simulations.
- Given the difficulty of the general problem, we focus on a restricted uncertainty function for cameras and present an approximation algorithm which runs in polynomial time, guarantees a bounded deviation from the optimal solution and addresses occlusion constraints.

A. Related work

One of the most well-known placement problems is the Art Gallery Problem [2] where a minimum number of omnidirectional cameras is sought to guard every point in a gallery represented by a polygon. Art gallery problems emphasize visibility/occlusion issues and there is no explicit representation of the quality of guarding – which is the focus of this paper.

Coverage and placement problems received a lot of attention recently. The problem of relocating sensors to improve coverage has been studied in [3]. In this formulation, the sensors can individually estimate the positions of the targets. However, the quality of coverage decreases with increasing distance.

Our work here builds up on the previous work on placing triangulation based sensors [4], [1]. As mentioned before, in our previous work we presented an approximation algorithm for minimizing the uncertainty metric given by $\frac{d(s_1,x)\times d(s_2,x)}{\sin \angle c_1 x c_2}$. In [4], the authors present an approximation algorithm which addreses visual occlusions and angle constraints. In the second part of this paper, we extend these

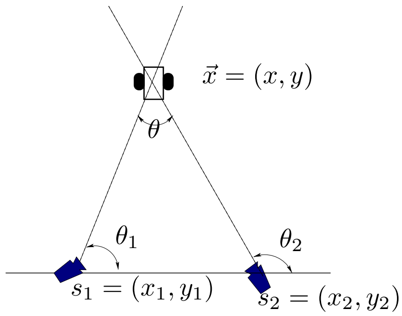

Fig. 1. The uncertainty in estimating the position of the target at x is given by: $U(s_1, s_2, x) = \frac{d(s_1,x) \times d(s_2,x)}{\sin \theta}$

previous results to simultaneously address visual occlusions, angle and distance constraints.

Other related results include [5] where the problem of controlling the configuration of a sensor team which employs triangulation for estimation has been studied. The authors present a numerical, particle-filter based framework. The problem of choosing the best subset of cameras for a given placement has been studied recently in [6]. In this work, the focus is on selecting a small subset of cameras to minimize a joint uncertainty measure. In the present work, we restrict ourselves to stereo-pairs but focus on placement issues. A recent related result was presented in [7] where the problem of relocating a sensor team whose members are restricted to lie on a circle and charged with jointly estimating the location of the targets was studied.

II. THE PLACEMENT PROBLEM

In this section, we formalize the placement problem and establish its hardness. We start with an overview of the error model for triangulation based state estimation.

A. Error model

The term triangulation refers to inferring the state $\vec{x}$ of a target (e.g.: a robot) by solving a system of simultaneous equations $\vec{z} = h(\vec{x})$ where $\vec{z}$ denotes the observation vector. As an example consider the process of estimating the position $\vec{x} = [x \;\; y]$ of a target (or a robot) using measurements from two cameras. We assume calibrated cameras, hence their location are known with respect to a common reference frame and their measurements can be interpreted as angles with respect to the horizontal axis (see Figure 1).

In this case, we have observables θ_1 and θ_2 and solve for the unknowns x and y in:

$$\tan \theta_1 = \frac{y_1 - y}{x_1 - x} \quad \tan \theta_2 = \frac{y_2 - y}{x_2 - x}$$

One way of establishing the accuracy of the estimation is to study the effect of small variations in the observables on the estimate. This effect can be established by studying the determinant of the Jacobian $H = \frac{\delta h}{\delta \vec{x}}$ which is commonly referred to as the Geometric Dilution of Precision (GDOP). In case of cameras, the $GDOP$ is given by

$$U(s_1, s_2, x) = \frac{d(s_1, x) \times d(s_2, x)}{|\sin \angle s_1 x s_2|} \tag{1}$$

where $d(x, y)$ denotes the Euclidean distance between x and y and $\theta = \angle s_1 x s_2$ is the angle between the sensors and the target (Figure 1). The details of this derivation can be found in [8]. In general, Equation 1 suggests that better measurements are obtained when the sensors are closer to the target and the angle is as close to 90 degrees as possible.

Similarly, the uncertainty in merging the measurements of two range sensors (which correspond to circles centered at the sensor location, passing through the target), can be shown to be:

$$U(s_1, s_2, x) = \frac{1}{|\sin \angle s_1 x s_2|} \tag{2}$$

In general, it is desirable to obtain a placement algorithm for arbitrary uncertainty measures $U(s_1, s_2, x)$ so as to incorporate additional sensing constraints such as occlusion, minimum clearance required by cameras, etc. In the next section, we formalize the sensor placement problem.

B. Problem formulation

Let $\mathcal{W}$ be the workspace which consists of all possible locations of the robot. We assume that $\mathcal{W}$ is discretized and given by a set of points. Similarly, let $\mathcal{S}$ be the set of candidate sensor locations. In addition to the two sets $\mathcal{W}$ and $\mathcal{S}$, we are given a function, $U(s_i, s_j, w)$ for all $s_i, s_j \in \mathcal{S}$ and $w \in \mathcal{W}$ which returns the uncertainty in localization when the robot is at location $w \in \mathcal{W}$ and queries sensors s_i and s_j. The function U can be easily defined to incorporate sensor limitations. For example, for cameras, we can define $U(s_i, s_j, w)$ to be infinite if one of the cameras can not see the point w.

Let $S = \{s_1, \ldots, s_n\} \subseteq \mathcal{S}$ be a set of sensors placed at locations s_1 through s_n. When there is no danger of confusion, we will use s_i to denote the location of sensor i as well. For a given placement S and a location $w \in \mathcal{W}$, let $assign(w, S) = \arg\min_{s_i, s_j \in S} U(s_i, s_j, w)$ be the assignment function which chooses the best pair of sensors for location w.

The uncertainty of a placement is defined as $U(S, \mathcal{W}) = \max_{w \in \mathcal{W}} U(w, assign(w, S))$.

We can now define the sensor placement problem:

Given a workspace $\mathcal{W}$, candidate sensor locations $\mathcal{S}$, an uncertainty function U and an uncertainty threshold U^*,

find a placement S with minimum cardinality such that $U(S, \mathcal{W}) \leq U^*$.

C. Hardness of the sensor placement problem

The hardness of the sensor-placement problem can be easily obtained by establishing its relation to the well-known k-center problem, which is NP-Complete. In the k-center problem, we are given a set of locations for centers and a set of targets along with a distance function $d(i, j)$ between the centers and the targets. The objective is to minimize the maximum distance between any target and the center closest to it [9]. The converse problem, where the maximum distance from each vertex to its center is given and the number of centers is to be minimized, is also NP-Complete [10]. Further, this problem is equivalent to the dominating set problem [10] which is not only NP-complete, but also can not be approximated within a factor better than $\log n$ in polynomial time [11]. Here, n denotes the number of target locations. The converse problem can be easily seen to be a special case of the sensor placement problem where the uncertainty function is chosen as $U(s_i, s_j, w) = \min\{d(s_i, w), d(s_j, w)\}$. Hence, sensor placement is at least as hard as the mentioned problems.

D. A mathematical programming formulation

There are many different types of sensors with different measurement characteristics. Since the general placement problem is hard, when designing placement algorithms, constraints imposed by the estimation process must be utilized. However, designing a dedicated placement algorithm for every type of sensor is a tedious process. Therefore, in this section, we present a general solution framework which can be utilized to solve placement problems that arise in practice.

The general sensor placement problem can be formulated as an integer linear programming (ILP) problem as follows:

minimize

$$\sum_j y_j \tag{3}$$

subject to

$$y_j \geq x^u_{ij} \qquad \forall u, i, j \tag{4}$$

$$x^u_{ij} = 0 \qquad \forall u, i, j \;\text{ with }\; U(u, i, j) \geq U^* \tag{5}$$

$$\sum_i z^u_i = 2 \qquad \forall u \tag{6}$$

$$\sum_i x^u_{ij} = z^u_i \qquad \forall u, j \tag{7}$$

$$\sum_j x^u_{ij} = z^u_j \qquad \forall u, i \tag{8}$$

We define a binary variable y_j for every location j. If $y_j = 1$, a sensor will be placed at location j. Other binary variables are z^u_i and x^u_{ij}. The index u varies over all possible target locations whereas i and j vary over candidate sensor locations. Variables z^u_i become 1 when the sensor at location i is assigned to target location u. Variables $x^u_{i,j}$ are set to 1 if the target location u is monitored by sensors at locations i and j.

Equation 3 is the cost function, i.e. the total number of sensors. The constraints on the placement are imposed by Equations 4 – 8.

The first constraint (Equation 4) ensures that if a sensor at location j will be assigned to a target u, then a sensor must be placed at location j in the first place. Equation 5 guarantees sensing and quality constraints: it prevents sensor pairs which do not satisfy the constraints from being assigned to a target location.

Equation 6 guarantees that two sensors are placed to monitor the target u.

Finally, Equations 7 and 8 make the connection between the variables x^u_{ij} and z^u_i. The variable x^u_{ij} can be 1 if and only if i and j are the locations for the sensor pair which is assigned to monitor the target u. All the other x^u_{ij} variables with same u but different i and j locations will be 0 (due to Equation 6). Therefore, if i' and j' are the two locations for the sensors to be assigned for the target u', the total of sum $x^{u'}_{i'j}$ will be equal to $z^{u'}_{i'}$ and $z^{u'}_{j'}$.

Since the sensor placement problem is NP-Complete, this ILP can not be solved in polynomial time in its full generality. However, there are many efficient algorithms for solving ILPs in practice. In Section IV, we demonstrate the practical feasibility of this approach in simulations.

III. A LOG FACTOR APPROXIMATION ALGORITHM ADDRESSING OCCLUSIONS

In this section, we present an approximation algorithm for a modified version of the uncertainty metric for triangulation with bearing-only sensors such as cameras. As stated in Equation 1, the uncertainty in estimating the position of a target at location x from sensors s_1 and s_2 is given by:

$$U(s_1, s_2, x) = \frac{d(s_1, x) \times d(s_2, x)}{|\sin \angle s_1 x s_2|}$$

Our goal is to design a placement algorithm which minimizes this uncertainty metric and addresses occlusions in the workspace.

In a workspace with obstacles, the strategy of the optimum solution is not predictable and can result in a placement which is undesirable in practice. For example, an optimal placement can compensate an obtuse or an acute angle between target and sensors by placing sensors very close to the target. Similarly, it is possible to have one sensor very close while the second sensor is very far from the target: their product will still remain small.

Therefore, instead of minimizing the product, it makes sense to explicitly restrict the distances and the angle between the sensors and the target. In this section, we present an approximation algorithm for the problem of placing a minimum number of sensors with the following properties.

Let S be a placement of sensors, and x be a target location. We assume that the workspace is represented by a polygon and say that a camera at s_1 *sees* a point x inside the polygon, if the line segment s_1x lies completely inside the polygon.

The placement S is called a valid placement if, for all x in the workspace, two sensors $s_1(x), s_2(x) \in S$ can be assigned to x such that

(i) both $s_1(x)$ and $s_2(x)$ see x,
(ii) $\alpha^* \leq \angle s_1(x) x s_2(x) \leq \pi - \alpha^*$, and
(iii) $d(s_1(x), x) \leq D^*$ and $d(s_1(x), x) \leq D^*$

where D^* and α^* are user defined threshold values. In [4], Efrat et al. present an approximation algorithm for placing sensors that addresses constraints (i) and (ii). In this section, we present an extension of their algorithm to accommodate constraint (iii) as well. We start with some preliminaries.

A. Preliminaries

A set system is a pair (X, R) where X is a subset and R is a collection of some subsets of X. We say that a set of subsets $R' \subseteq R$ *cover* X if their union is equal to X. The minimum set cover problem is to find a minimum cardinality $R^* \subseteq R$ that covers X.

As an example, consider the following camera placement problem: we are given a set of candidate target locations X (which lie inside a polygon) along with a set of candidate camera locations $\mathcal{S}$. The goal is to place a minimum number of cameras such that every point in X is visible from at least one camera. This problem (which we call visibility cover) can be formulated as a set-covering problem for the set system (X, R) where R contains a subset $R(s)$ for each candidate sensor location $s \in \mathcal{S}$ where $R(s) = \{x | x \text{ is visible from } s\}$.

The following definition is introduced in [4]: A point x is *two-guarded* at angle α by sensors s_1 and s_2, if the angle $\angle s_1 x s_2$ is in the interval $[\alpha, \pi - \alpha]$ and both sensors can see x.

The algorithm in [4] proceeds in two stages. In the first stage, a visibility cover C_1 of X is computed. This gives a placement where each location x is assigned to a single sensor $s_1(x)$. In the second stage, a second set of sensors C_2 is computed such that, for each $x \in X$, there exists a sensor $s_2(x) \in C_2$ such that x is two-guarded by $s_1(x)$ and $s_2(x)$ at angle $\alpha/2$. The existence of the set C_2 is guaranteed by the following lemma.

Lemma 1 ([4]): Let C^* be a set of sensors that two-guard X at angle α and C_1 be a visibility cover of X. Then, for any point $x \in X$ there exist sensors $s_1 \in C^*$ and $s_2 \in C_1$ that two-guard x at angle $\alpha/2$.

Let OPT be the minimum set of sensors that two-guard x. It is shown in [4] that one can compute C_1 and C_2 above in polynomial time such that $|C_1 \cup C_2| = O(OPT \log OPT)$. In other-words, one can simultaneously satisfy condition (i), obtain a 2-approximation for (ii) and a $\log$ approximation to the number of sensors.

In the next section, we show how this result can be extended to satisfy condition (iii). That is, we show how two sets C_1 and C_2 can be computed in a way that simultaneously satisfy conditions (i) and (iii), obtain a 2-approximation for (ii) and a $\log$ approximation to the number of sensors.

B. Computing C_1 and C_2

A standard algorithm to compute a cover of a given set system (X, R) is the greedy algorithm: we initialize all elements in X to be uncovered. Next, we select a subset R' from R which contains the most number of uncovered elements. We mark all elements of R' as covered and repeat this process until all elements of X are covered (or we run out of subsets in R). It is well known that the greedy algorithm is a $log|X|$-approximation, that is, the number of subsets chosen is guaranteed to be within a factor of $O(\log |X|)$ of the optimal solution.

For geometric set systems, however, once can usually do better:

Definition 2: Given a set system (X, R), let A be a subset of X. We say A is shattered by R if $\forall Y \subseteq A$, $\exists R' \in R$ such that $R' \cap A = Y$. The VC-dimension of (X, R) is the cardinality of the largest set that can be shattered by R [12].

In what follows, we will utilize two well-known properties of set systems with bounded VC-dimension.

(i) The VC-dimension of a set system obtained by the intersection or union of two set systems of constant VC-dimension is also constant [13].

(ii) Let (X, R) be a set system and (X', R') be its dual: $X' = R$ and $R' = \{R(x) : x \in X\}$ where $R(x)$ is the set of subsets in R which contain the element x. If (X, R) has a constant VC-dimension, so does its dual [14].

Our algorithms rely on the fact that, for sets systems with finite VC-dimension d, there are algorithms which can compute a set-cover of the set system whose size is at most $O(d \cdot \log OPT \cdot OPT)$ where OPT is the size of the minimum set-cover [15], [16]. In other words, in the finite (or bounded) VC-dimension case, one can obtain a $\log OPT$ approximation, as opposed to the $\log |X|$ approximation obtained by the greedy algorithm.

Let (X, R) be a set system where X is a set of points on the plane. We say (X, R) is a *disk set system* if R is obtained by intersecting X with the set of all possible disks on the plane. Similarly, we call (X, R) a *triangle set system* if R is obtained by intersecting X with all triangles. It is a well known fact that both disk and triangle set systems have constant VC-dimension. Another example of a set system with finite VC-dimension is the following. Let X be a set of points in a polygon P. For each possible point $p \in P$, let $V(p)$ be the set of those points in X that are visible from p. In [17] it was shown that the set system $(X, \{V(p) : p \in P\}$ has a constant VC dimension if P is simply-connected or has a bounded number of holes.

We now present the details of the algorithm to construct C_1. Recall that X is a set of candidate target locations we would like to cover and $\mathcal{S}$ is the set of candidate sensor locations. Both C and $\mathcal{S}$ are points sampled inside a polygon which represents the workspace. We are given thresholds D^* and α^* that specify the angle and distance constraints. Let OPT be a minimum cardinality sensor placement which satisfies constraints (i) – (iii).

To compute set C_1, we first build the set system (X, R') where

$$R' = \{R'(s) | s \in \mathcal{S}\}$$
$$R'(s) = \{x | x \in X \wedge x \text{ is visible from } s \wedge d(x, s) \leq D^*\}$$

The VC-dimension of this set system is constant. This is because the set system can be expressed as an intersection of a visibility set system and a disk set system.

Since there is a set-cover of (X, R') of size at most $|OPT|$, one can find a cover of size $O(OPT \log OPT)$ in polynomial time using [15], [16]. This gives us the set C_1. For each target location $x \in X$, let $s_1(x)$ be a sensor in C_1 which is visible from x with $d(x, s_1(x)) \leq D^*$.

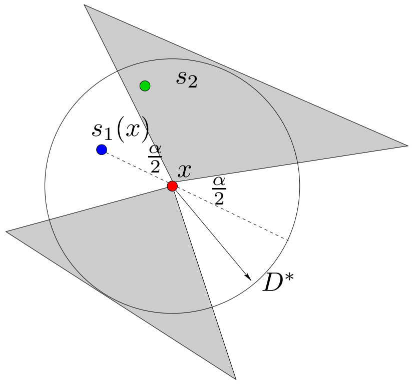

Fig. 2. The sensor s_2 covers x because it satisfies the distance constraint and, together with $s_1(x)$, it satisfies the angle constraint as well.

In order to compute C_2, we build the set system (X, R'') where

$$
\begin{aligned}
R'' &= \{R''(s) | s \in \mathcal{S}\} \\
R''(s) &= \{x | x \in X \wedge \\
&\quad x \text{ is visible from } s \wedge \\
&\quad d(x, s) \leq D^* \wedge \\
&\quad \angle s_1(x) x s \in [\frac{\alpha}{2}, \pi - \frac{\alpha}{2}]\}
\end{aligned}
$$

Lemma 1 can be easily modified to show that each $R''(s)$ is nonempty if the optimal solution which satisfies all three constraints exists. We now show that (X, R'') has a constant VC-dimension. Consider a point $x \in X$, together with sensor $s_1(x)$ assigned in the previous stage. We say that a sensor s_2 covers x if it sees x, satisfies both the distance constraint and the angle constraint together with $s_1(x)$. Now consider a set system $(\mathcal{S}, Q)$ where $\mathcal{S}$ is the set of candidate sensor locations and Q is obtained by inserting for each target location $x \in X$, the set of sensors which cover x. This set system can be obtained as follows: First, construct a set system corresponding to intersections with triangle pairs as shown in Figure 2. Second, intersect this new set system with visibility and disk set systems. Since all these set systems have finite VC-dimension, the resulting set system has finite VC-dimension as well. The set system (X, R'') is simply the dual of $(\mathcal{S}, Q)$ and hence, has a finite VC-dimension.

IV. Simulations

In this section, we present two simulations to demonstrate the feasibility of using an ILP solver for sensor placement.

We computed optimal placements for two environments which satisfy all three constraints (visibility, angle and distance) given in Section III. The left column in Figure 3 correspond to the solutions obtained by the ILP solver. For these simulations, we used the Cbc ILP solver on the NEOS server [18]. The first environment has 68 target locations and 84 sensor locations, whereas the second environment has 64 target locations and 70 sensor locations. The number of x_{ij}^u variables in the ILP introduced in Section II-D is mn^2 where m is the number of target locations and n is the number of candidate sensor locations. However, most of these variables are redundant. For example, if $U(u, i, j) > U^*$, we can remove the variable x_{ij}^u. This alone reduces the number of x_{ij}^u variables for the first environment from 479808 to 3470 and for the second environment from 313600 to 2324. The same approach can be applied to remove other redundant binary variables. For example, we can remove a variable z_i^u if u is not visible from i or the distance between them is greater than the threshold.

In these two simulations, we chose the maximum grid size (number of locations) for each environment such that the ILP can be solved under 5 minutes. The ILP for the first environment contained 4612 variables and 2184 constraints. The ILP for the second environment contained 3196 variables and 1668 constraints.

The right column in Figure 3 is obtained using the approximation algorithm presented in Section III. For simplicity, we used the greedy algorithm to compute sets C_1 and C_2. In the first environment, the approximation algorithm matched the performance of the ILP solution. In the second environment, however, it placed 18 sensors as opposed to the 16 placed by the ILP.

V. Conclusion and Future Work

In this paper we addressed the sensor placement problem in scenarios where robots operating in a workspace query the nodes of a sensor-network to localize themselves. Specifically, we studied the problem of computing the minimum number and placement of sensors so that the uncertainty at every point in the workspace is less than a given threshold. We focused on triangulation based state estimation where measurements from two sensors must be combined for an estimation.

First, we showed that the general problem for arbitrary uncertainty models is computationally hard. For the general version, we presented a framework based on integer linear programming which can be used to solve placement problems in practice. We demonstrated the practical feasibility of this approach with simulations. Finally, we presented an approximation algorithm for a geometric uncertainty measure which simultaneously addresses visual occlusions, angle and distance constraints.

Our future work includes the deployment of a real camera network in our building and to address placement (calibration) uncertainties. Future research also includes improving the log approximation ratio achieved by the approximation algorithm.

Acknowledgement

This work is supported in part by NSF CCF-0634823.

References

[1] V. Isler, "Placement and distributed deployment of sensor teams for triangulation based localization," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2006, to appear. [Online]. Available: http://www.cs.rpi.edu/ isler/new/pub/pubs/isler06icra.pdf

[2] J. O'Rourke, *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.

[3] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

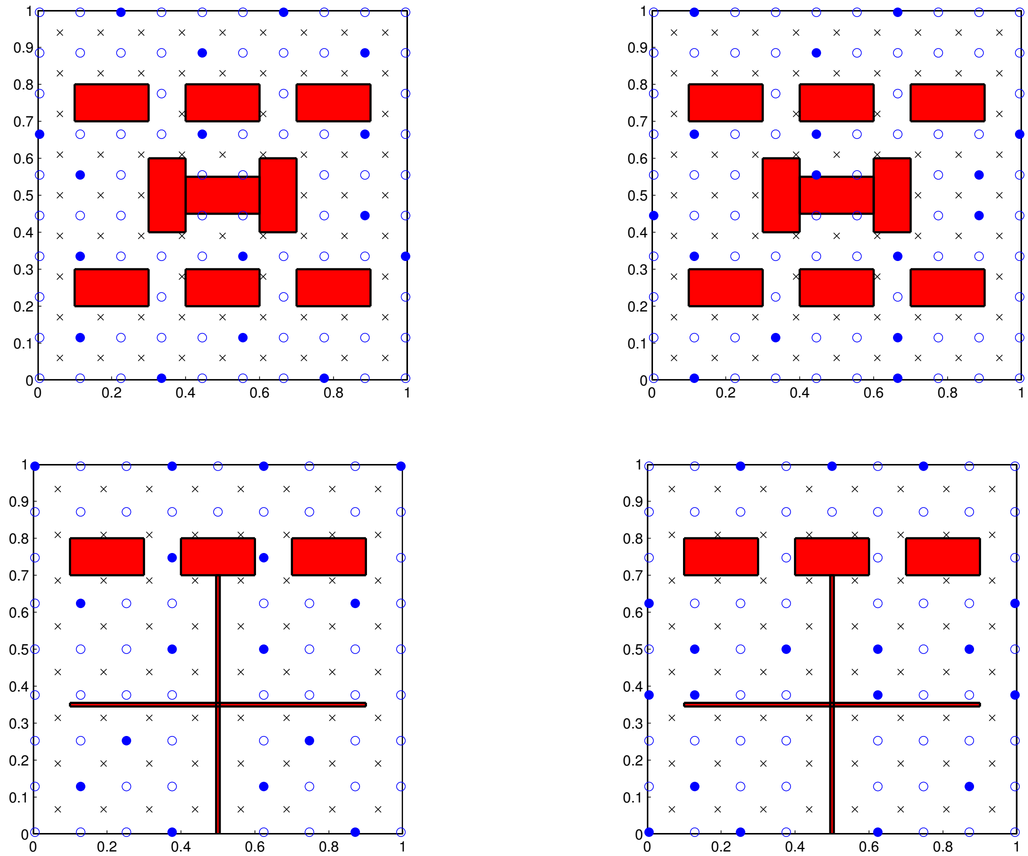

Fig. 3. Comparison of the ILP solution with the approximation algorithm in two different environments. Target points in the environment are represented by the symbol $\times$. Candidate sensor locations are represented by the symbol $\circ$. A filled circle indicates a sensor placed at that location. **TOP ROW:** The figure on the left shows the placement obtained using the ILP solution. The figure on the right shows the placement for the same environment obtained using the approximation algorithm. In this simulation, both algorithms place 16 sensors. **BOTTOM ROW:** The figure on the left (resp. right) shows the placement obtained using the ILP solution (resp. approximation algorithm). In this simulation, the ILP solution achieved the desired uncertainty with 16 sensors whereas the approximation algorithm achieved the same uncertainty with 18 sensors.

[4] A. Efrat, J. S. B. Mitchell, and S. Har-Peled, "Approximation algorithms for two optimal location problems in sensor networks," in *Broadnets*, 2005. [Online]. Available: http://valis.cs.uiuc.edu/ sariel/papers/04/sensors

[5] J. Spletzer and C. Taylor, "Dynamic sensor planning and control for optimally tracking targets," *International Journal of Robotics Research*, vol. 22, no. 1, pp. 7–20, 2003.

[6] V. Isler and R. Bajcsy, "The sensor selection problem for bounded uncertainty sensing models," *IEEE Tran. Automation Science and Engineering*, 2006, to appear.

[7] S. Aranda, S. Martínez, and F. Bullo, "On optimal sensor placement and motion coordination for target tracking," in *International Conference on Robotics and Automation*, Barcelona, Spain, Apr. 2005.

[8] A. Kelly, "Precision dilution in mobile robot position estimation," in *Intelligent Autonomous Systems*, Amsterdam, Holland, 2003.

[9] T. F. Gonzales, "Clustering to minimize the maximum intercluster distance," *Theoretical Comput. Sci.*, no. 38, pp. 293–306, 1985.

[10] J. Bar-Ilan and D. Peleg, "Approximation algorithms for selecting network centers," in *Proc. 2nd Workshop on Algorithms and Data Structures*, 1991, pp. 343–354.

[11] U. Feige, "A threshold of $\ln n$ for approximating set cover," *Journal of the ACM (JACM)*, vol. 45, pp. 634 – 652, 1998.

[12] V. N. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theory of Probability and its Applications*, vol. 16, no. 2, pp. 264–280, 1971.

[13] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, "Learnability and the vapnik-chervonenkis dimension," *J. ACM*, vol. 36, no. 4, pp. 929–965, 1989.

[14] B. Chazelle, *Handbook of Discrete and Computational Geometry*. CRC Press, 2004, ch. 10 (The Discrepancy Method in Computational Geometry), pp. 983–996.

[15] H. Brönnimann and M. Goodrich, "Almost optimal set covers in finite VC-dimension," *Discrete & Computational Geometry*, vol. 14, no. 4, pp. 463–479, 1995.

[16] G. Even, D. Rawitz, and S. Shahar, "Hitting sets when the vc-dimension is small," *Inf. Process. Lett.*, vol. 95, no. 2, pp. 358–362, 2005.

[17] P. Valtr, "Guarding galleries where no point sees a small area," *Israel Journal of Mathematics*, vol. 104, pp. 1–16, 1998.

[18] "Cbc – mixed integer linear programming solver." [Online]. Available: http://neos.mcs.anl.gov/neos/solvers/milp:Cbc/AMPL.html