Sensor Placement for Triangulation Based Localization

Onur Tekdas, *Student Member, IEEE,* and Volkan Isler, *Member, IEEE*

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, and present approximation algorithms for two geometric instances. For the general problem, we present a solution framework based on integer linear programming and demonstrate its practical feasibility with simulations.

Index Terms

Sensor network deployment, localization, approximation algorithms.

Rensselaer Polytechnic Institute
Department of Computer Science
Technical Report 06-18

Corresponding Author is Onur Tekdas.
Email: tekdao@cs.rpi.edu
Tel: +1-518-276-2135
Fax: +1-518-276-2529
Mailing Address:
Department of Computer Science
110 8th Street, Lally 207
Troy, NY, 12180

Volkan Isler
Tel: +1-518-276-3275
Fax: +1-518-276-2529
Mailing Address:
Department of Computer Science
110 8th Street, Lally 207
Troy, NY, 12180

Both authors are with the Department of Computer Science, Rensselaer Polytechnic Institute. Emails: {tekdao, isler}@cs.rpi.edu. Corresponding author is Onur Tekdas. Mailing Address: 110 8th Street, Lally 207, Troy, NY, 12180, USA. Tel: 518-276-2135 Fax:518-276-2529.

Earlier versions of the results in this paper appeared in [1], [2]. In this full version, we improve the approximation ratio of the placement algorithm presented in [1] (Section III).

Sensor Placement for Triangulation Based Localization

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. As an example, consider the localization problem whose solution is a prerequisite for many robotics applications. Sensor network technology offers a scalable solution for localization of heterogeneous, independent robot teams operating in a large and complex environment: We can deploy a calibrated camera-network in such an environment and the robots can query these sensors for localization – instead of relying on on-board sensors and customized applications. Other robotics problems which can benefit from the existence of a sensor network include navigation, search and surveillance.

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 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 uncertainty threshold, what is the minimum number, and placement of cameras so that the uncertainty in localization is less than the threshold at every point in the workspace?*

In this paper, we make three main contributions:

First, we show, via a simple reduction, that the general sensor placement problem (where the uncertainty measure is arbitrary) is NP-Complete.

Second, we focus on geometric uncertainty measures and present two approximation algorithms for two different instances. In the first instance, the uncertainty in the estimation is proportional to $\frac{d(s_1,x)\times d(s_2,x)}{\sin \angle c_1 x c_2}$ where x is the position of the robot, s_1 and s_2 are the locations of the sensors and d denotes the Euclidean distance. We present 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 this instance, we do not address the issue of visual-occlusions in the workspace (Equivalently, it is assumed that there are no obstacles in the workspace and the cameras are omnidirectional.). In the second instance, we address visual occlusions and present a log-factor approximation algorithm for a more restricted uncertainty measure (where thresholds for allowable angle and sensor-target distances are given).

Finally, we present a general framework based on mathematical programming which, in practice, 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.

After an overview of the related work (Section I-A), we present a formal definition of the placement problem (Section II-B) and establish its hardness (Section II-C). We continue with the approximation algorithms (Sections III and IV) and the mathematical programming framework (Section V).

A. Related work

The most well-known placement problem that involves cameras is the Art Gallery Problem [3] 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 [4]. In this formulation, the sensors can individually estimate the positions of the targets. However, the quality of coverage decreases with increasing distance. Placement of sensors which jointly estimate the states of targets is an active research area.

In [5], the problem of controlling the configuration of a sensor team which employs triangulation for estimation is studied. The authors present a numerical, particle-filter based framework. In their approach, the optimal move at each time step is estimated by computing an n dimensional gradient numerically where n is the size of the joint configuration space of the team. In [6], 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. In [7], the authors study the problem of placing cameras in a polygonal workspace in such a way that for each point of interest, there are two cameras which can view the

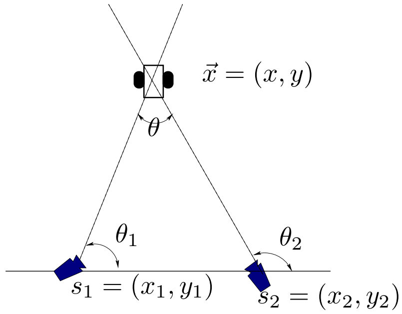


Fig. 1. The uncertainty in estimating the position of the target at x is given by: $U(s_1, s_2, x) = \frac{d(s_1,x) \times d(s_2,x)}{\sin\theta}$

point at a "good" angle. The authors present an approximation algorithm which guarantees that the number of sensors is within a logarithmic factor of the optimal value. The second approximation algorithm presented in this paper (Section IV) is based on this work. Our contribution is in extending this algorithm to handle distance constraints.

Finally, we note that the placement problem can be viewed as a clustering problem where the cluster centers correspond to the chosen sensor locations. For example, in the k-center problem, we are given a set of locations for centers and a set of targets. The objective is to minimize the maximum distance between any target and the center closest to it. In the placement problem, studied here, there are two centers associated with each location and the cost is much more involved than the Euclidean distance. We were unable to find any literature that addresses this type of clustering problems. Therefore, the placement algorithm presented in Section III may also be of independent interest.

II. THE PLACEMENT PROBLEM

Before we formalize the placement problem and establish its hardness, we start with an overview of uncertainty models for triangulation based state estimation.

A. Uncertainty in triangulation

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 locations 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

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

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:

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

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 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. For concreteness, throughout the paper we assume that $\mathcal{W}$ is discretized and given by a set of points on the plane. Similarly, let $\mathcal{S}$ be the set of candidate sensor locations[1]. 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

[1]Here, it is implicitly assumed that the only relevant sensor parameter is location. If there are additional sensor parameters such as orientation, $\mathcal{S}$ corresponds to the entire parameter set.

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.

In the next sections, we present polynomial time approximation algorithms for two geometric versions of the placement problem.

III. A CONSTANT FACTOR PLACEMENT ALGORITHM

In this section, we present a placement algorithm to minimize the error metric given by Equation 2, repeated here for convenience:

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

Let U^* be a desired uncertainty threshold and OPT be an optimal placement. We will first present an algorithm to compute a placement S with $|S| \leq 3|OPT|$ and $U(S, \mathcal{W}) \leq 6U^*$ where U^* is the uncertainty threshold. Let us call such a placement as a *competitive placement*. We assume that $\mathcal{W} \subseteq \mathbb{R}^2$ and cameras can be placed on the entire plane. However, as we will see shortly, no competitive placement can afford to place cameras too far from the workspace.

Let $R = \sqrt{U^*}$. The proposed placement algorithm consists of two phases. In the first phase, we choose a set of centers which will be used to determine the location of the cameras. In the second phase, we place cameras on circles whose centers coincide with the chosen centers and whose radii are at most $2R$. We will show that this placement is a competitive one.

The centers are chosen by the following algorithm:

Algorithm selectCenters(workspace $\mathcal{W}$):
- $C = \emptyset$, $W \leftarrow \mathcal{W}$
- while $W \neq \emptyset$
  - $w \leftarrow$ an arbitrary point in W
  - $C \leftarrow C \cup \{w\}$
  - $W \leftarrow W \setminus \{x : d(x, w) < 2R, x \in W\}$

The following lemma shows that the number of centers is small with respect to $|OPT|$.

Lemma 1: Let C be the set of centers chosen by selectCenters and OPT be an optimal placement. $|OPT| \geq |C|$.

Proof: For each center $c \in C$, let us define D_c to be a disk centered at c with radius R. Since the distance between the centers is at least $2R$, the disks D_c are pairwise disjoint. We claim that each disk D_c contains at least one camera in OPT, which proves the lemma.

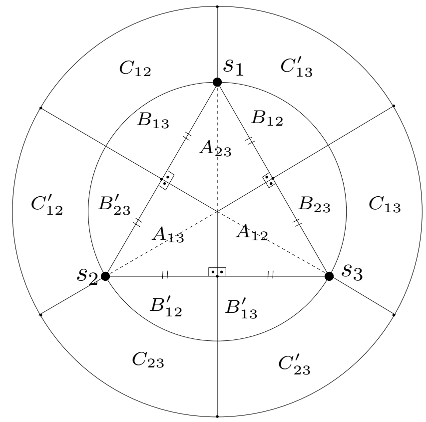


Fig. 2. This figure shows the partitions A, B and C, and their divided partitions. Three sensors (s_1, s_2, s_3) are placed on the circumference of $circle\left(c, \sqrt[3]{\frac{1}{4}}r\right)$. The area X_{ij} represents the region in the partition X, where *placeSensors* assigns s_i and s_j for all $w \in X_{ij}$.

Suppose the claim is not true and let c be a center such that OPT has no cameras in D_c. But then, for any $s_i, s_j \in OPT$, the error in observing the center of D_c will be:

$$U(s_i, s_j, c) = \frac{d(s_i, c) d(s_j, c)}{|\sin \angle s_i c s_j|} \geq d(s_i, c) d(s_j, c) > R^2 = U^*$$

However, this means that OPT exceeds the error threshold on c. A contradiction! ■

In the second phase, we use the set of centers to determine the placement of cameras.

Algorithm placeSensors(centers C):
- for each $c_i \in C$
  - $W_i \leftarrow \{w : d(c_i, w) < 2R, w \in W\}$
  - $r_i = \max_{w \in W_i} d(c_i, w)$
  - Place three cameras s_{i1}, s_{i2}, s_{i3} on $circle(c_i, r'_i)$ with $r'_i = \sqrt[3]{\frac{1}{4}} r_i$, at angles $\frac{\pi}{2}$, $\frac{7\pi}{6}$ and $\frac{11\pi}{6}$(See Figure 2).

Here $circle(c, r)$ denotes the circle centered at c with radius r. The angles are with respect to a coordinate frame whose origin is at the center of the circle and orientation is arbitrary.

Clearly, algorithm *placeSensors* places at most $3 \cdot |OPT|$ cameras (Lemma 1). All we need to show is that for any point w in the workspace, we can find two cameras s_j and s_k such that $U(w, s_j, s_k) \leq 6U^*$. The next lemma shows the existence of such camera pairs.

Lemma 2: For each center $c_i \in C$, let W_i be the set of points defined in algorithm *placeSensors*. Let $S_i = \{s_{i1}, s_{i2}, s_{i3}\}$ be the set of three cameras placed by *placeSensors* inside $circle(c_i, r_i)$, along the circumference of $circle(c_i, r'_i = \sqrt[3]{\frac{1}{4}} r_i)$, at angles $\frac{\pi}{2}$, $\frac{7\pi}{6}$ and $\frac{11\pi}{6}$. For any point $w \in W_i$, there exists an assignment of two sensors (s_{ij}, s_{ik}), such that $U(w, s_{ij}, s_{ik}) \leq 6U^*$ where $s_{ij}, s_{ik} \in S_i$.

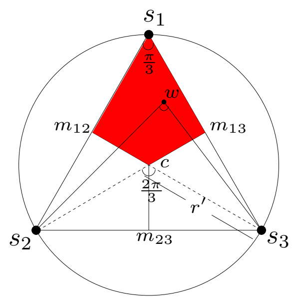


Fig. 3. We divide $\triangle s_1s_2s_3$ into three parts using the bisectors of the triangle. The shaded area shows the possible set of locations for w such that the assignment (s_2, s_3) satisfies $U(w, s_2, s_3) < \frac{3}{2}r^2$.

Proof: For each point w inside $circle(c, r)$, we will show that one can pick two sensors such that the uncertainty in observing w from these two sensors is less than $\frac{3}{2}r^2$. The lemma follows since $r^2 \leq 4U^*$.

We divide the set of points inside $circle(c, r)$ into three parts: $A = \triangle s_1s_2s_3$, $B = circle(c, r') \setminus A$ and $C = circle(c, r) \setminus circle(c, r')$. Here, we abused the notation to avoid additional notation: $circle(c, r)$ refers to all points inside the circle. Note that $\triangle s_1s_2s_3$ is an equilateral triangle. This fact is used throughout our proof. Let w be any point inside $circle(c, r)$. We consider three cases $w \in A$, $w \in B$ and $w \in C$, and show that the uncertainty at w in all cases is less then $\frac{3}{2}r^2$.

Case ($w \in A$):

We partition A into three regions using angular bisectors as shown in Figure 2. In present the proof for the case $w \in A_{23}$. The proof for both cases A_{12} and A_{13} is symmetric.

Let w be a point inside A_{23} (Figure 3). First, we establish a bound on the angle $\angle s_2ws_3$, which we use to bound distances $dist(s_2, w)$ and $dist(s_3, w)$. Finally, we obtain a bound on the uncertainty using these separate bounds on the numerator and the denominator in the uncertainty formula.

The angle $\angle s_2ws_3$ is always between the angles, $\angle s_2s_1s_3$ and $\angle s_2cs_3$. Hence, the angle between the sensors and the target is bounded by $\frac{\pi}{3} \leq \angle s_2ws_3 \leq \frac{2\pi}{3}$. It is easy to verify that $dist(s_2, w) \leq dist(s_2, s_1)$ and $dist(s_3, w) \leq dist(s_3, s_1)$, i.e. $dist(s_2, w), dist(s_3, w) \leq \sqrt{3}r'$. Finally, the uncertainty at $w \in A$ is bounded by $U(w, s_2, s_3) \leq \frac{3r'^2}{\sin \pi/3} < 1.38r^2$.

Case ($w \in B$):

Let us partition B into 6 equal parts using the bisectors of triangle $s_1s_2s_3$ as shown in Figure 2. The indices ij in B_{ij} correspond to the sensors assigned to all points inside B_{ij}. Note that, B_{ij} and B'_{ij} are symmetric with respect to the bisector of the line segment $seg(s_i, s_j)$.

Suppose that w lies in the region between the arc of a circle $arc(m'_{13}, s_3)$ and the line segments $seg(s_3, m_{13})$ and $seg(m_{13}, m'_{13})$ (See Figure 4 and also B_{23} in Figure 2). Let l be the tangent line to $circle(c, r)$ at the point s_3. For any point $p \in B$, p' represents the intersection point between circumference $circum(c, r)$ and the ray $ray(s_2, p)$. For clarity, let's relabel the following angles: $\alpha = \angle s_2ws_3$, $\beta = \angle s_2w's_3$, $\gamma = \angle ws_3w'$ and $\delta = \angle s_1s_3w'$. Since angle β is an inscribed angle, $\beta = \frac{\pi}{3}$ holds. The angle γ is lower bounded by 0 and upper bounded by δ (otherwise,

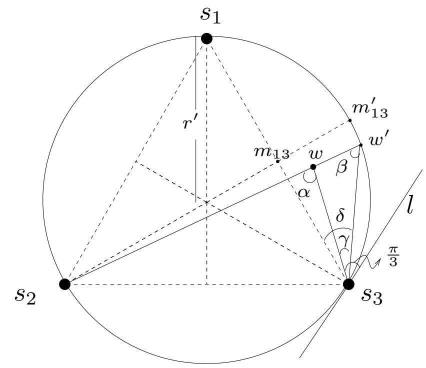


Fig. 4. Inside the region bounded by $arc(m'_{13}, s_3)$, $seg(s_3, m_{13})$ and $seg(m_{13}, m'_{13})$, any w sees s_2 and s_3 with properties: $\frac{\pi}{3} \leq \angle s_2ws_3 \leq \frac{2\pi}{3}$ and $U(w, s_2, s_3) < \frac{3}{2}r^2$.

w would lie outside of B). Further, δ is restricted by the tangent line l and $seg(s_3m_{13})$, which shows that $0 \leq \delta \leq \frac{\pi}{3}$. Finaly, using the fact $\alpha = \beta + \gamma$ and the bounds established earlier, we can bound $\angle s_2ws_3$: $\frac{\pi}{3} \leq \angle s_2ws_3 \leq \frac{2\pi}{3}$. The distances $dist(s_2, w)$ and $dist(s_3, w)$ reach their maximum value when w is on m'_{13}, i.e. $dist(s_2, w) \leq dist(s_2, m'_{13})$ and $dist(s_3, w) \leq d(s_3, m'_{13})$. Hence, the bound on uncertainty of $w \in B$ is $U(w, s_2, s_3) \leq \frac{2r'^2}{\sin \pi/3} < 0.92r^2$.

Case ($w \in C$):

We partition C into 6 equal pieces using bisectors of $\triangle s_1s_2s_3$ as shown in Figure 2. In what follows, we establish a bound for an arbitrary point w inside the region C_{13}. The proof for the region C'_{13} is symmetric with respect to the bisector of $seg(s_1, s_3)$. The generalization of this proof to other regions is obtained by rotating the $circle(c, r)$ in the counterclockwise direction around its origin by angles: $2\pi/3$ and $4\pi/3$.

For any point w inside C_{13}, we assign sensors s_1 and s_3 (See Figure 5). To obtain a bound on the uncertainty, we first establish a lower bound on $\sin(\angle s_1ws_3)$, followed by an upper bound on the product $dist(s_1, w) \times dist(s_3, w)$. Finally, we show that both bounds are reached at the same point.

For any point p inside C, p' denotes the intersection of $line(c, p)$ with $circum(c, r')$, and p'' denotes the intersection of $line(c, p)$ with $circum(c, r)$. Let w be a point inside C. The angle $\angle s_1ws_3$ is always between $\angle s_1w's_3$ and $\angle s_1w''s_3$. Note that, $\angle s_1w's_3 = \pi/3$ as it is an inscribed angle of $circle(c, r')$. Therefore, $sin(\angle s_1ws_3)$ is bounded from below by $min(sin(\angle s_1w''s_3), sin(2\pi/3))$.

For the remaining part of the proof, we will represent w in polar coordinates as shown in Figure 5. Let us define three functions: $Mult(\rho, \theta) = dist(s_1, w) \times dist(s_3, w)$, $Angle(\rho, \theta) = \angle s_1ws_3$, and $Uncert(\rho, \theta)$ is the uncertainty of the target. These three functions are given by:

$$
\begin{aligned}
Uncert(\rho,\theta) &= \frac{Mult(\rho,\theta)}{sin(Angle(\rho,\theta))} \\
Mult(\rho,\theta) &= \sqrt{(\rho^2-2\rho r'\sin\theta+r'^2)\left(\rho^2-2\rho r'\cos\left(\theta+\frac{\pi}{6}\right)+r'^2\right)} \\
Angle(\rho,\theta) &= \frac{\pi}{3}+\arctan\left(\frac{r'-\rho\sin\theta}{\rho\cos\theta}\right)+\arctan\left(\frac{r'-\rho\cos\left(\theta+\frac{\pi}{6}\right)}{\rho\sin\left(\theta+\frac{\pi}{6}\right)}\right)
\end{aligned}
$$

Note that, $-\pi/6 \leq \theta \leq \pi/6$ and $r'_i \leq \rho \leq r_i$.

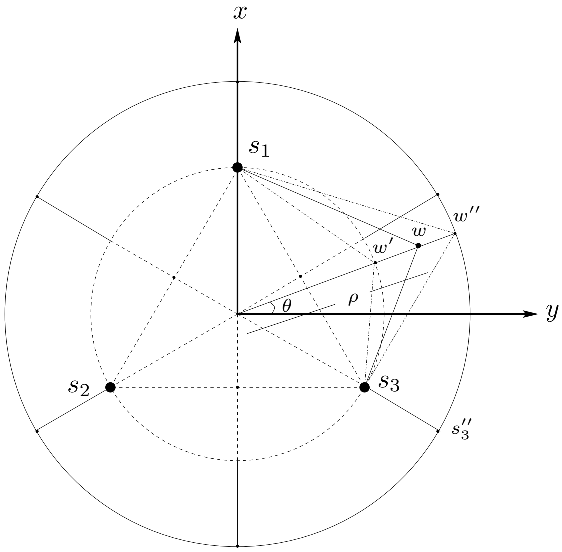


Fig. 5. In this figure, we represent w with ρ and θ parameters in polar coordinate system. For all $w \in C_{13}$, $\sin(\angle s_1 w s_3) \leq \sin(\angle s_1 s_3'' s_3)$ and $dist(s_1, w) \times dist(s_3, w) \leq dist(s_1, s_3'') \times dist(s_3, s_3'')$

We showed before that $\sin(\angle s_1 w s_3)$ is bounded by $min(sin(\angle s_1 w'' s_3), sin(2\pi/3))$ where w'' is a point on circumference of $circle(c, r)$. Notice that $\sin(\angle s_1 s_3'' s_3) < sin(2\pi/3)$, consequently $\sin(\angle s_1 w s_3)$ is bounded by $min_{-\pi/6 \leq \theta \leq \pi/6}(sin(Angle(r, \theta)))$. The function $Angle(r, \theta)$ has its local maxima and minima at $\theta = \pi/6$ and $\theta = -\pi/6$, respectively and it is increasing in its domain. This can be shown by investigating the boundary of domain and roots of the first derivative of $Angle(r, \theta)$.

Both $\sin(Angle(\pi/6, \theta))$ and $\sin(Angle(-\pi/6, \theta))$ are less than 1, accordingly $\sin(Angle(r, \theta))$ gets its minimum value at $\theta = -\pi/6$.

By Euclid's exterior angle theorem (in any triangle the angle opposite the greater side is greater), we have $dist(s_1, w) \leq dist(s_1, w'')$ and $dist(s_3, w) \leq dist(s_3, w'')$. Therefore, $Mult(\rho, \theta) \leq Mult(r, \theta)$ holds. By the extreme value theorem, $Mult(r, \pi/6) \leq Mult(r, \theta) \leq Mult(r, -\pi/6)$.

We showed that both the maximum value of product $dist(s_1, w)$ and $dist(s_3, w)$, and the minimum value of $\sin(\angle s_1 w s_3)$ appears at the same point $w = s_3''$. As a result, the uncertainty of $w \in C$ holds the following:

$U(w, s_2, s_3) \leq \frac{Mult(r, -\pi/6)}{\sin(Angle(r, -\pi/6)} < 1.38 r^2$.

Hence, in all cases we have an uncertainty value which is less than $\frac{3}{2} r^2$. Finally, since *placeSensors* guarantees that $r^2 \leq 4U^*$ for all $w \in \mathcal{W}$, $U(\mathcal{W}, \mathcal{S}) < \frac{3}{2} r^2 \leq 6U^*$. ■

In Figure 2, we present numerical results for the optimal partitioning of the disk and the corresponding uncertainty values (for the placement given by *placeSensors*). The maximum uncertainty matches the bound obtained in Lemma 2. However, the optimal assignment scheme is slightly different than the one used in the proof (cf. Figure 6).

Is it possible to obtain a better uncertainty guarantee? In general, let us define an (α, β)-approximation algorithm for sensor placement be an algorithm which places at most β times the number of cameras used in an optimal placement and guarantees a deviation of factor α in uncertainty. From the results presented above, we have $(6, 3)$ approximation algorithm. Clearly, there is a trade-off between α and β. Using algorithm *placeSensors* as a subroutine, we can obtain a class of approximation algorithms by covering each disk of radius $2R$ (used by *placeSensors*) with k disks of smaller radius. This guarantees a smaller deviation from U^*. The problem now becomes a disk-covering problem: Given a disk of radius $2R$, find the smallest radius $r(k) < 1$ required for k equal disks to completely cover the original disk. Clearly, this would guarantee a reduction of $r(k)^2$ in the performance guarantee of placeSensors, at the expense of increasing the number of cameras by a factor k. The interested reader can find different values of $r(k)$ in [12].

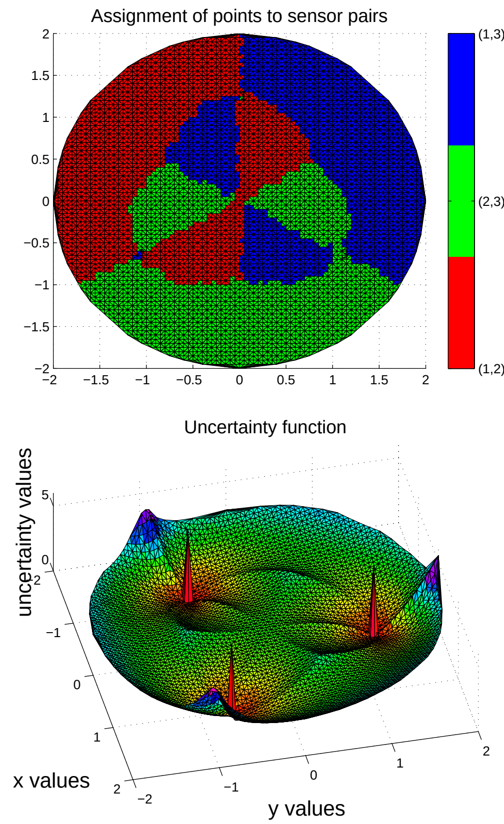


Fig. 6. Top figure shows the optimum assignment in $circle(c, r)$ and bottom figure shows the uncertainty in $circle(c, r)$.

IV. A LOG FACTOR APPROXIMATION ALGORITHM FOR HANDLING 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.

When there are obstacles in the workspace, a sensor placement strategy which solely minimizes this uncertainty value may result in a placement with some properties which are undesirable in practice. For example, in an optimal placement, a target may make a very large angle with the cameras it is assigned to. This is because the optimal solution may compensate the decrease in the value of $|\sin\angle s_1 x s_2|$ by putting the cameras too close and still obtain a low uncertainty value. Or similarly, under this metric, two cameras can be assigned to a target in a way that one camera is very far and the other is very close. We note that the placement algorithm presented in the previous section avoids these degeneracies by design in the case where there are no obstacles.

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 $s_1 x$ 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 [7], 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 [7]: 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 [7] 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 3 ([7]): 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 [7] 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 4: 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 [13].

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 [14].

(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 [15].

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 [16], [17]. 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

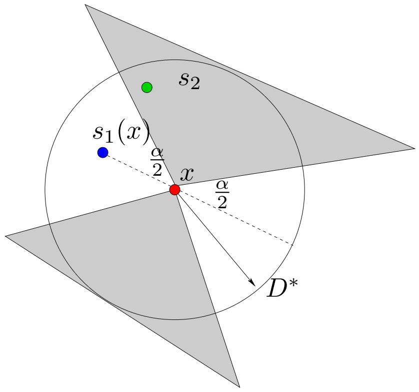


Fig. 7. The sensor s_2 covers x because it satisfies the distance constraint and, together with $s_1(x)$, it satisfies the angle constraint as well.

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 folklore 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 [18] 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

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

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 [16], [17]. 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 $d(x, s_1(x)) \leq D^*$.

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 3 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 7. 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.

V. 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

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

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

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

$$\sum_i x_{ij}^u = z_i^u \qquad \forall u, j \tag{7}$$

$$\sum_j x_{ij}^u = z_j^u \qquad \forall u, i \tag{8}$$

We define a binary variable y_j for every location j [2]. If $y_j = 1$, a sensor will be placed at location j. Other binary variables are z_i^u and x_{ij}^u. The index u varies over all possible target locations whereas i and j vary over candidate sensor locations. Variables z_i^u become 1 when the sensor at location i is assigned to target location u. Variables $x_{i,j}^u$ 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.

[2]If the sensors have more parameters, y_j is obtained by discretizing the entire parameter space. For example, for limited field-of-view cameras, one would define a y_j variable for each (position, orientation) pair.

The first constraint (Equation 4) forces 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 enforces 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_{ij}^u and z_i^u. The variable x_{ij}^u 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_{ij}^u 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 $x_{i'j}^{u'}$ will be equal to $z_{i'}^{u'}$ and $z_{j'}^{u'}$.

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 the next section, we demonstrate the practical feasibility of this approach in simulations.

VI. Simulations

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

We computed optimal placements which satisfy all three constraints (visibility, angle and distance) given in Section IV for two environments. The top rows in Figures 8 and 9 correspond to the solutions obtained by the ILP solver. For these simulations, we used the Cbc ILP solver on the NEOS server [19]. Both environments occupied the unit square. The angle constraint was chosen to be $\pi/2$ and the distance constraint was 0.4 units. The first environment contained 68 target locations and 84 sensor locations, whereas the second environment had 64 target locations and 70 sensor locations. The number of x_{ij}^u variables in the ILP introduced in Section V were mn^2 where m is the number of target locations and n is the number of candidate sensor locations. However, most of these variables were redundant. For example, if $U(u,i,j) > U^*$, we could remove the variable x_{ij}^u. This alone reduced 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 was applied to remove other redundant binary variables. For example, a variable z_i^u were removed when u was not visible from i or the distance between them was 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 bottom rows in Figures 8 and 9 are obtained using the approximation algorithm presented in Section IV. 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.

VII. 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.

After showing that the general problem for arbitrary uncertainty models is computationally hard, we focused on two geometric instances and presented approximation algorithms with provable performance guarantees. We also presented a framework based on integer linear programming which can be used to solve general placement problems in practice. We demonstrated the practical feasibility of this approach with simulations.

Our future work includes the deployment of a real camera network in our building and to address placement (calibration) uncertainties. Future research also includes extending our results to three dimensions.

Acknowledgment

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. [Online]. Available: http://www.cs.rpi.edu/ isler/new/pub/pubs/isler06icra.pdf

[2] O. Tekdas and V. Isler, “Sensor placement for triangulation based localization,” in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2007, to appear.

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

[4] 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.

[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] 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.

[7] 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

[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] E. W. Weisstein, “Disk covering problem,” MathWorld–A Wolfram Web Resource, http://mathworld.wolfram.com/DiskCoveringProblem.html.

[13] 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.

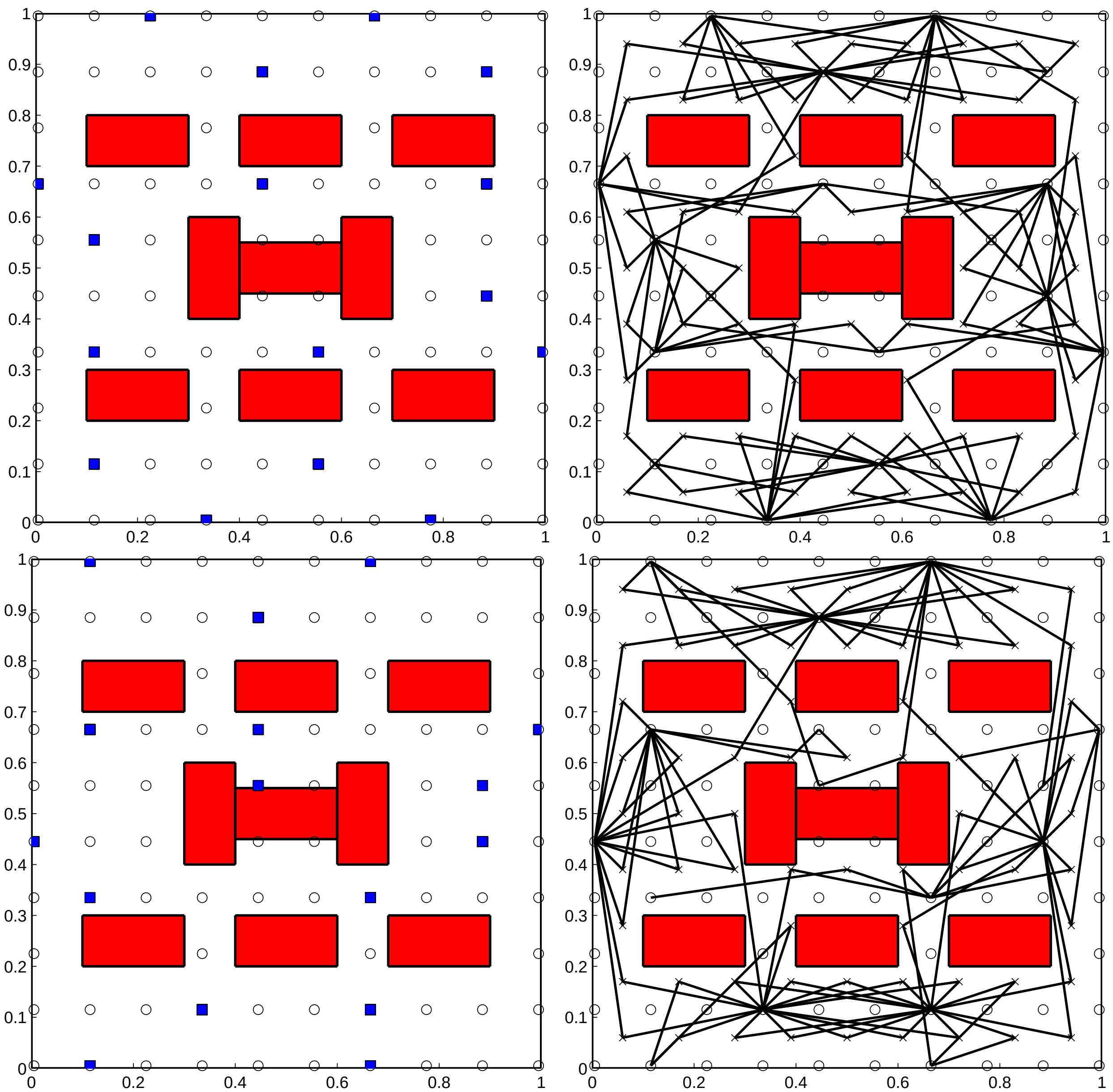


Fig. 8. **TOP ROW:** Placement obtained from the ILP solution. The figure on the left shows the selected cameras whereas the right figure shows the assignment of points (represented by $\times$ symbols) in the workspace to the placed cameras. **BOTTOM ROW:** Placement obtained from the approximation algorithm. In this simulation, both algorithms place 16 sensors.

[14] 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.

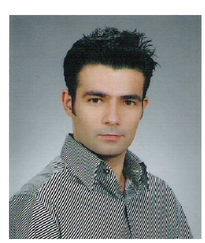
[15] B. Chazelle, *Handbook of Discrete and Computational Geometry*. CRC Press, 2004, ch. 10 (The Discrepancy Method in Computational Geometry), pp. 983–996.

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

[17] 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.

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

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



Onur Tekdas is a PhD student in the Department of Computer Science at Rensselaer Polytechnic Institute. He received his BS degree in Computer Engineering from Middle East Technical University, Turkey. His research interests are motion planning and placement of sensor networks.

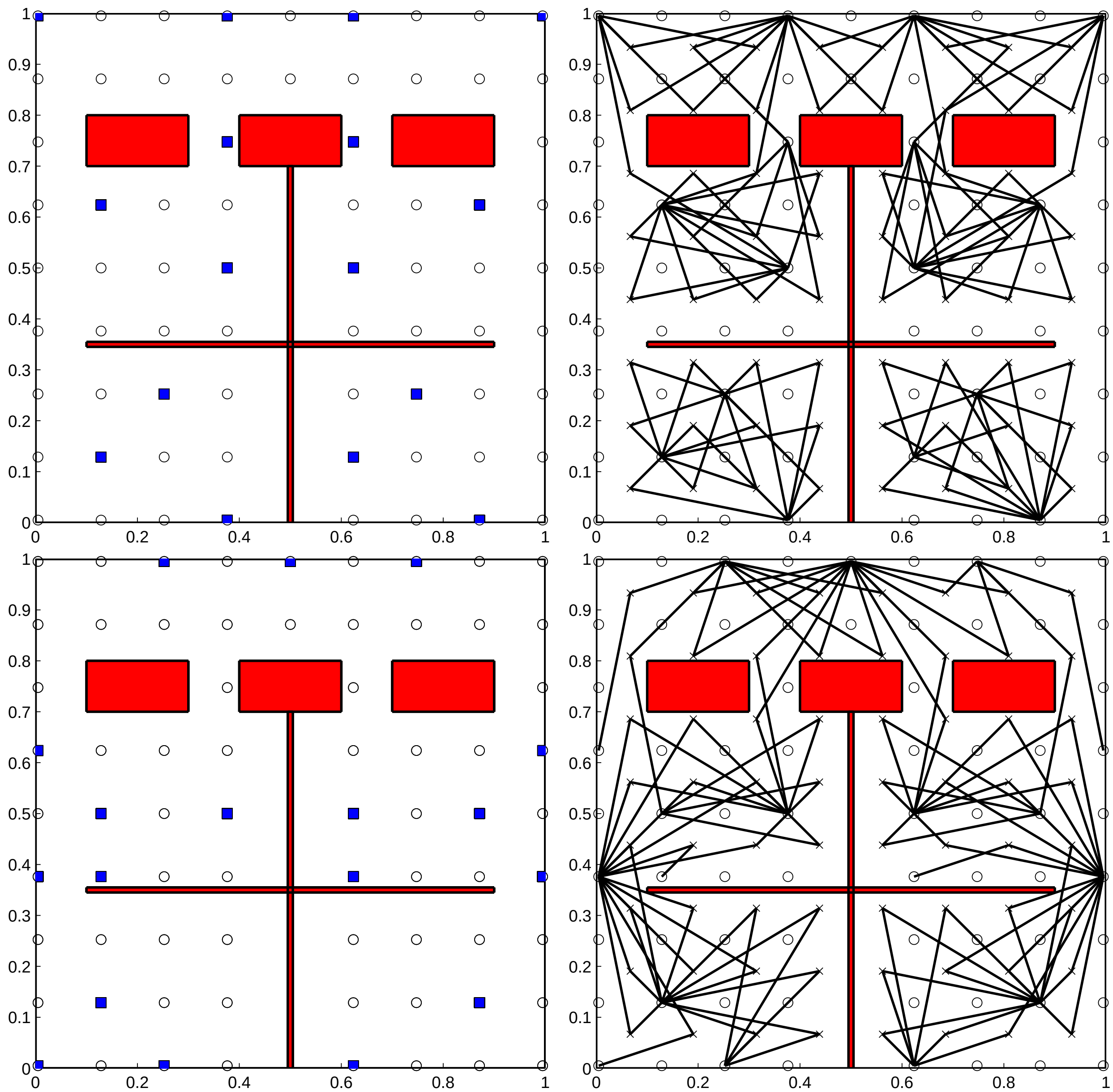


Fig. 9. **TOP ROW:** Placement obtained from the ILP solution. The figure on the left shows the selected cameras whereas the right figure shows the assignment of points (represented by $\times$ symbols) in the workspace to the placed cameras. **BOTTOM ROW:** Placement obtained from the approximation algorithm. In this simulation, the ILP solution is 16 sensors whereas the approximation algorithm places 18 sensors.

Volkan Isler is an Assistant Professor in the Department of Computer Science at Rensselaer Polytechnic Institute. Before joining RPI, he was a post-doctoral researcher at the Center for Information Technology Research in the Interest of Society (CITRIS) at the University of California, Berkeley. He received his MSE and PhD degrees in Computer and Information Science from the University of Pennsylvania and BS degree in Computer Engineering from Bogazici University in Istanbul, Turkey. His research interests are in robotics (pursuit-evasion, motion planning and human-robot interaction) and sensor-networks (deployment, target tracking and localization with camera networks).