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.

This problem is NP-hard in its most general from. For the general version, we present a solution framework based on integer linear programming and demonstrate its application in a fire-tower placement task. Next, we study the special case of bearing-only localization and present an approximation algorithm with a constant factor performance guarantee.

Note to Practitioners

Sensor networks can provide robust and scalable solutions to the localization problem which arises in numerous automation tasks. A common method for localization is triangulation in which measurements from two sensors are combined to obtain the location of a target.

In this work, we study the problem of finding the minimum number, and placement of sensors in such a way that the uncertainty in localization is bounded at every point in the workspace when triangulation is used for estimating the location of a target. We present an efficient geometric algorithm for bearing-only localization which can be used for the deployment of camera-networks. We also present a generic framework for arbitrary uncertainty metrics, and demonstrate its utility in an application where watchtowers are deployed to detect forest fires.

Index Terms

Sensor network deployment, localization, approximation algorithms.

SHORT PAPER

Corresponding Author is Onur Tekdas.
Email: tekdas@cs.umn.edu
Tel: +1-612-625-4002
Fax: +1-612-625-0572
Mailing Address:
University of Minnesota
Department of Computer Science and Engineering
4-192 EE/CS Building
200 Union Street SE
Minneapolis, MN 55455, USA

Volkan Isler
Email: isler@cs.umn.edu
Tel: +1-612-625-1067
Fax: +1-612-625-0572
Mailing Address:
University of Minnesota
Department of Computer Science and Engineering
4-192 EE/CS Building
200 Union Street SE
Minneapolis, MN 55455, USA

Both authors are with the Department of Computer Science and Engineering, University of Minnesota. Emails: {tekdas, isler}@cs.umn.edu. Corresponding author is Onur Tekdas. Mailing Address: 4-192 EE/CS Building, 200 Union Street SE, Minneapolis, MN 55455, USA. Tel: 612-625-4002 Fax: 612-625-0572.

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 IV). This work is supported in part by NSF grants 0907658, 0917676 and 0936710.

Sensor Placement for Triangulation Based Localization

***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.**

This problem is NP-hard in its most general form. For the general version, we present a solution framework based on integer linear programming and demonstrate its application in a fire-tower placement task. Next, we study the special case of bearing-only localization and present an approximation algorithm with a constant factor performance guarantee.

I. Introduction

The process of determining the location of an object from two sensor measurements is commonly referred to as triangulation. For example, a forest fire is localized by triangulating bearing measurements from two fire-towers whose relative locations are known. Similarly, images taken from two calibrated cameras can be used to localize an object.

As sensors are becoming inexpensive, deploying many sensors in a workspace to provide localization services is becoming feasible (see e.g. [3], [4]). We believe that this technology provides a valuable alternative to on-board localization for mobile robots for the following reasons. First, on-board localization requires a sensor and a processor on each robot. In scenarios where many inexpensive, specialized robots (e.g. Roombas) operate in the same workspace, it may be cheaper to install a small number of cameras in the environment and use them to localize all of the robots. Second, localization with external sensors can be more robust than on-board localization especially when there are not many distinct features in the environment. Finally, localization with external sensors may be the only alternative in scenarios such as localizing adversarial entities. We note that the additional burden of calibrating external sensors can be relieved by using fully automated techniques [5]–[7].

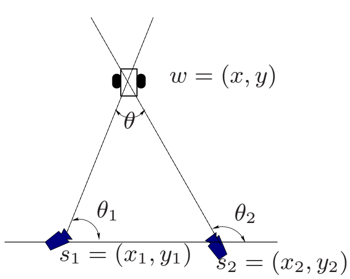

Fig. 1. The uncertainty in estimating the position of the target at w using bearing measurements θ_1 and θ_2 is given by: $U(s_1, s_2, w) = \frac{d(s_1,w)d(s_2,w)}{|\sin\theta|}$

The target-sensor geometry plays a significant role in the accuracy of an estimation obtained via triangulation. As an example, consider triangulation with bearing measurements. If the target and the two sensors are collinear, the sensors can not localize the target. In general, the uncertainty in bearing-only triangulation is proportional to the product of target-sensor distances and inversely proportional to the sine of the angle between the target and the sensors [8] (Figure 1). The environment also plays a role in localization with triangulation. For example, occlusions caused by the environment may prevent a sensor from participating in the triangulation process.

In this paper, motivated by these factors, we study the following problem: given an environment model, an estimation model and an uncertainty threshold, what is the minimum number and placement of sensors to guarantee that the uncertainty in estimating the location of a target in the workspace is not greater than the given threshold regardless of the target's location?

Contributions: After formalizing the sensor placement problem in Section II-A, we present a general solution framework based on integer linear programming (Section III). Since the general problem is NP-hard, we also focus on a common special case, bearing-only localization, and present an approximation algorithm that deviates from the optimal solution only by a constant factor both in the number of cameras used and the uncertainty in localization (Section IV).

In the next section, we start with the problem formulation.

II. The Sensor Placement Problem

A. Problem formulation

In this section, we formulate the Sensor Placement Problem for Triangulation Based Localization (SPP). In SPP, we are given a workspace $\mathcal{W}$ which consists of all possible locations of the target. Let s be a k-tuple representing related sensor parameters which can include, for example, location and orientation of a sensor. The second input to SPP is $\mathcal{Q}$, the domain of s. In other words, $\mathcal{Q}$ is the set of all possible placements of a single sensor. Finally, we are given a function $U : \mathcal{Q} \times \mathcal{Q} \times \mathcal{W} \rightarrow \mathbb{R}$, where $U(s_i, s_j, w)$ returns the uncertainty in localizing a target located at w using two sensors with parameters s_i and s_j. In general the function U is specific to particular environment and sensing models. For example, $U(s_i, s_j, w)$ can be infinite if the environment causes an occlusion between w and either s_i or s_j. Similarly, if the sensors have range or field-of-view constraints, the function U can be defined to incorporate them. To simplify the notation, we left the dependency of U on environment and sensing models implicit throughout the text.

Let $S = \{s_1, \ldots, s_n\} \subseteq \mathcal{Q}^n$ be a placement of sensors where the i^{th} sensor has parameters s_i. The quality of a placement in a workspace is determined as follows:

$$U(S, \mathcal{W}) = \max_{w \in \mathcal{W}} \min_{s_i, s_j \in S} U(s_i, s_j, w)$$

In other words, to establish the quality of a placement in a workspace, we take the largest uncertainty value over the entire workspace. To compute the uncertainty value for a specific location in the workspace, we find the best pair of sensors for that location.

We can now define the sensor placement problem: Given a workspace $\mathcal{W}$, candidate sensor locations $\mathcal{Q}$, an uncertainty function U and an uncertainty threshold U^*, find a placement S with minimum cardinality such that $U(S, \mathcal{W}) \leq U^*$.

It is easy to see that SPP is a hard problem 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]. The converse problem can be easily seen to be a special case of the SPP where the uncertainty function is chosen as $U(s_i, s_j, w) = \min\{d(s_i, w), d(s_j, w)\}$. Hence, SPP is at least as hard as the converse k-center problem.

B. Related work

Coverage and placement problems received a lot of attention recently. However, most of the existing work focuses on sensors that are capable of estimating the state of the target independently. In this section, we review work on the placement of sensors which jointly estimate the states of targets.

In [11], 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. This work solves a local placement problem (around the target) for a small number of robots at each time step. Instead, we focus on the global placement problem across the entire work space. In [12], 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 the present work, we study more general environments on the plane. In [13], 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 point at a "good" angle. The authors present a $\log$-approximation for this problem. Their work can address occlusions. Our approximation algorithm has a better performance ratio (it is a constant factor approximation algorithm). The uncertainty metric is also more sophisticated as it incorporates distance and angle. However, we do not address the issue of occlusions in the workspace. The work in [14] studies the problem of assigning disjoint pairs of sensors to targets. The authors introduce the notion of *Universal Placement* where "good" assignments exist regardless of the targets' locations. In this work, the main focus is on multi-target tracking, and there is an explicit requirement that the sensor pairs be disjoint. The issue of minimizing the number of sensors, which is the main focus of our current work, is not addressed.

III. A General Discretized Solution

In this section, we present a general solution, based on integer linear programming, to SPP. The formulation requires discretizing the workspace $\mathcal{W}$ and the domain of sensor parameters $\mathcal{Q}$.

The Integer Linear Program (ILP) uses the following variables. For each possible sensor parameter $j \in \mathcal{Q}$, we define a binary variable y_j. If $y_j = 1$ after solving the ILP, a sensor with parameters j will be placed. Other binary variables are z_i^u and x_{ij}^u. The index u varies over all possible target locations in $\mathcal{W}$, whereas i and j vary over $\mathcal{Q}$. In the solution, variables z_i^u become 1 when the sensor with parameters i is assigned to target location u. Variables x_{ij}^u are set to 1 if the target location u is monitored by sensors with parameters i and j.

The complete ILP is given by:

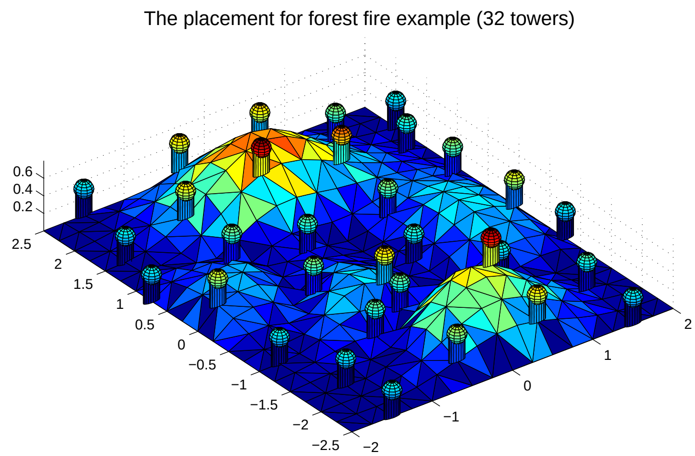

Fig. 2. A terrain model of a forest, and a placement of 32 fire-towers achieving small localization uncertainty.

minimize
$$\sum_j y_j \tag{1}$$
subject to
$$y_j \geq x_{ij}^u \qquad \forall i, j, u \tag{2}$$
$$x_{ij}^u = 0 \qquad \forall i, j, u \text{ with } U(i, j, u) > U^* \tag{3}$$
$$\sum_i z_i^u = 2 \qquad \forall u \tag{4}$$
$$\sum_j x_{ij}^u = z_i^u \qquad \forall i, u \tag{5}$$
$$\sum_i x_{ij}^u = z_j^u \qquad \forall j, u \tag{6}$$

Equation 1 is the objective function, i.e. the total number of placed sensors. The constraints on the placement are imposed by Equations 2 – 6. The first constraint (Equation 2) enforces that if a sensor with parameters j will be assigned to a target u, then such a sensor must be placed in the solution. Equation 3 enforces sensing and quality constraints: it prevents sensor pairs which do not satisfy the constraints from being assigned to a target location. Equation 4 guarantees that two sensors are placed to monitor the target u. Equations 5 and 6 maintain consistency between 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 parameters of the two sensors which are assigned to monitor the target at u. All the other x_{ij}^u variables with same u but different i and j parameters will be 0 (due to Equation 4). Therefore, if i' and j' are the parameters of the sensors to be assigned to the target at u', the sum of $x_{i'j}^{u'}$'s over all j's ($x_{ij'}^{u'}$'s over all i's) will be equal to $z_{i'}^{u'}$ ($z_{j'}^{u'}$).

In the next section, we present an example where the ILP formulation is used to solve a practical fire-tower placement problem.

A. An example

In this section, we demonstrate the utility of the ILP framework with an example. Imagine the task of placing fire-towers in a forest which are used for localizing events such as forest fires. Alternatively, one can imagine the task of deploying beacons which are used by firefighting robots for localization.

When the terrain is non-planar, such as the one shown in Figure 2, visual occlusions must be addressed. For this purpose, we use the following uncertainty model: Let s_1 and s_2 be two (candidate) fire-tower locations and w be a target location. If both s_1 and s_2 can see w, we compute the uncertainty using the formula $U(s_1, s_2, w) = \frac{d(s_1,w)d(s_2,w)}{|\sin \angle s_1 w s_2|}$ on the plane defined by s_1, s_2 and w. Otherwise, $U(s_1, s_2, w)$ is infinite. Note that it is possible to introduce a more sophisticated uncertainty measure for a particular sensor.

Suppose we are given the environment model shown in Figure 2, the error model mentioned above and an error threshold of 0.5. To setup the ILP, we first compute two sets of candidate sensor and target locations. We place a uniform grid on the x-y plane with resolution ζ and project the grid points onto the forest surface to obtain candidate sensor locations. These sensor locations correspond to the vertices of the triangles of the surface triangulation. To obtain candidate target locations, we shift the original grid by $\zeta/2$ in both x and y directions. The projection of this new grid onto the forest surface gives us candidate target locations. In terms of running time, the bottleneck is the number of x^u_{ij} variables. Therefore, we focus on them. In this example, there are 357 candidate sensor locations and 320 target locations which resulted in $40,783,680$ x^u_{ij} variables. Two simple optimizations are to disregard values with $x^u_{ij} = 0$ and to use only one of the duplicate pairs: x_{ij} and x_{ji}. This reduces the number of x^u_{ij} variables to $113,136$.

The resulting ILP was solved with $CPLEX 9.1.0$ solver on a PC equipped with a 2 GB RAM and 3.20 GHz CPU. Solvers like CPLEX, use branch-and-bound technique to solve ILPs. Since the number of variables is very large, the number of branch-and-bound operations is increased proportionally. Hence, rather than waiting until the optimal solution is reached, we found the best solution after $10,000$ branch-and-bound operations which took 35.54 CPU hours. The resulting placement has 32 fire-towers and is shown in Figure 2.

The generality of the ILP-based solution comes at the expense of computation time. When the number of variables is large, the ILP takes too long. It is possible to design efficient algorithms with provable performance guarantees for special cases of SPP. In the next section, we present such an algorithm for bearing-only localization on the plane.

IV. Placement for Bearing-only Triangulation

In this section, we address the problem of placing bearing-only sensors on the open plane to monitor the workspace given by an arbitrary subset of the plane. A common uncertainty function associated with this model is

$$U(s_1, s_2, w) = \frac{d(s_1, w)d(s_2, w)}{|\sin \angle s_1 w s_2|} \tag{7}$$

where $d(x, y)$ denotes the Euclidean distance between x and y and $\theta = \angle s_1 w s_2$ is the angle between the sensors and the target (Figure 1). The details of this derivation can be found in [8]. In general, Equation 7 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.

Even though we ignore occlusions caused by the environment, this version of the problem has applications in the placement of fire towers to monitor a (flat) forest and placement of omnidirectional cameras in convex environments.

Let U^* be a desired uncertainty threshold and OPT be an optimal placement which uses the minimum number of sensors such that $U(OPT, \mathcal{W}) \leq U^*$. We will present an algorithm to compute a placement S with $|S| \leq 3|OPT|$ and $U(S, \mathcal{W}) < 5.5U^*$ where $|S|$ and $|OPT|$ be the cardinalities of the corresponding placements. Our algorithm works for any workspace $\mathcal{W} \subseteq \mathbb{R}^2$. We assume that sensors can be placed anywhere in the plane. As we will see shortly, the optimal placement can not 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 C 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 achieves bounded deviation from the optimal one in terms of both the number of sensors and the performance guarantee.

The centers are chosen by the following algorithm:

Algorithm selectCenters(workspace $\mathcal{W}$):

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

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 $D(c, R)$ to be a disk centered at c with radius R. Since the distance between the centers is at least $2R$, the disks $D(c, R)$ are pairwise disjoint. We claim that each disk $D(c, R)$ contains at least one camera in OPT, which proves the lemma.

Suppose the claim is not true and let c be a center such that OPT has no cameras in $D(c, R)$. But then, for any $s_i, s_j \in OPT$, the uncertainty in observing 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 uncertainty 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 $c \in C$
  - $W_c \leftarrow \{w : d(c, w) < 2R, w \in W\}$
  - Let T be any equilateral triangle whose circumcircle is $D(c, 2R')$ where $R' = \sqrt[3]{\frac{1}{4}}R$
  - Place three sensors s_1, s_2, s_3 on the vertices of T (See Figure 4).

In Figure 3 we illustrate the two phases of the algorithm. We only show the first two selected disk centers and the placement of sensors inside their disks.

Clearly, algorithm *placeSensors* places at most $3|OPT|$ cameras (Lemma 1). All we need to show is that for any point w in the workspace, we can find two sensors s_i and s_j such that $U(s_i, s_j, w) < 5.5U^*$. The next lemma shows the existence of such camera pairs.

Lemma 2: Let $S = \{s_1, s_2, s_3\}$ be the set of three cameras placed by *placeSensors* inside $D(c, 2R)$. For any point $w \in W_c$, there exists an assignment of two cameras s_i and s_j, such that $U(s_i, s_j, w) < 5.5U^*$ where $1 \leq i, j \leq 3$.

Proof:

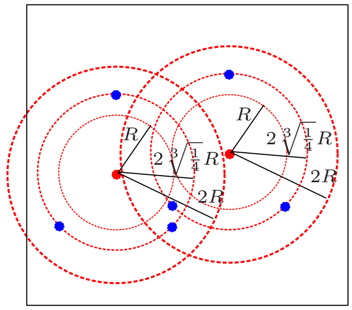

Fig. 3. This figure shows the two phases of the placement algorithm. In the first phase, we choose disk centers (for clarity we show only the first two selected centers which are represented with red circles whose radii are $2R$). In the second phase, we place three sensors (blue circles) for each center as described in algorithm *placeSensors*.

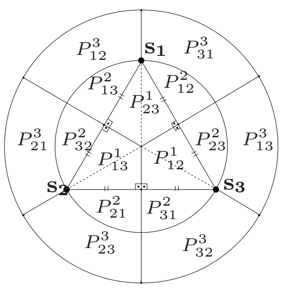

Fig. 4. This figure shows the partitioning of $D(c, 2R)$ into three partitions P^1, P^2 and P^3. Three sensors (s_1, s_2, s_3) are placed on the equilateral triangle's vertices which is determined by its circumcircle $D(c, 2R')$ where $R' = \sqrt[3]{\frac{1}{4}}R$. Each partition P^k divided into subregions P^k_{ij} where s_i and s_j are the sensors assigned for points inside P^k_{ij}

We divide the set of points inside $D(c, 2R)$ into three partitions: $P^1 = \triangle s_1s_2s_3$, $P^2 = D(c, 2R') \setminus P^1$ and $P^3 = D(c, 2R) \setminus D(c, 2R')$ (See Figure 4).

Let w be any point inside $D(c, 2R)$. Throughout the proof, we use various geometric properties for the three cases: $w \in P^1$, $w \in P^2$ and $w \in P^3$. For each P^k, we present a proof for a subregion P^k_{ij} such that $U(s_i, s_j, w) < 5.5U^*$ is satisfied for all $w \in P^k_{ij}$. The proof generalizes to all P^k_{ij} (hence to P^k) by symmetry.

Case ($w \in P^1$):

We divide P^1 into three regions using the bisectors of the triangle T and label them as shown in Figure 4. Let w be a point inside P^1_{23}. It can be easily verified that the following three inequalities hold, $\frac{\pi}{3} \leq \angle s_2ws_3 \leq \frac{2\pi}{3}$, $d(s_2, w) \leq 2\sqrt{3}R'$, $d(s_3, w) \leq 2\sqrt{3}R'$. Hence, $U(s_2, s_3, w) \leq \frac{12R'^2}{\sin \pi/3} = 5.4989U^* < 5.5U^*$ (See Figure 5).

Case ($w \in P^2$):

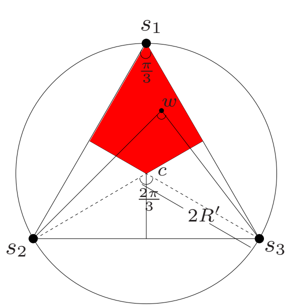

Fig. 5. We divide $P^1 = \triangle s_1s_2s_3$ into three regions using the bisectors of the triangle T. The shaded area shows the possible set of locations for w such that the assignment of s_2 and s_3 satisfies $U(s_2, s_3, w) =< 5.5U^*$.

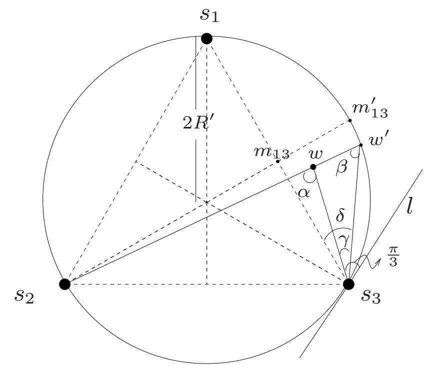

Fig. 6. We cut $P^2 = D(c, 2R') \setminus P^1$ into 6 regions by bisectors of triangle T. In this figure we consider the area P^2_{23} shown in Figure 4. For any $w \in P^2_{23}$, the assignment of s_2 and s_3 satisfies $U(s_2, s_3, w) < 5.5U^*$.

We partition P^2 into 6 equal parts using the bisectors of triangle T (See Figure 4). Suppose w lies in region: P^2_{23}. Let m_{13} be the middle point of $[s_1\ s_3]$, and w' and m'_{13} be the intersection points of the boundary of $D(c, 2R')$ with $\overrightarrow{s_2\ w}$ and $\overrightarrow{s_2\ m_{13}}$, respectively. For clarity, we use the same notation as in Figure 6, i.e. $\alpha = \angle s_2ws_3$, $\beta = \angle s_2w's_3$, $\gamma = \angle ws_3w'$ and $\delta = \angle s_1s_3w'$. For any $w \in P^2_{23}$, following inequalities hold: (i) $\beta = \frac{\pi}{3}$, (ii) $0 \leq \gamma \leq \delta$, (iii) $\frac{\pi}{6} \leq \delta \leq \frac{\pi}{3}$, (iv) $\alpha = \beta + \gamma$. Finally, using (i)–(iv), we can bound $\alpha = \angle s_2ws_3$: $\frac{\pi}{3} \leq \angle s_2ws_3 \leq \frac{2\pi}{3}$. The distances $d(s_2, w)$ and $d(s_3, w)$ are upper bounded by: $d(s_2, w) \leq d(s_2, m'_{13}) = 4R'$, $d(s_3, w) \leq d(s_3, m'_{13}) = 2R'$.

Hence, $U(s_2, s_3, w) \leq \frac{8R'^2}{\sin \pi/3} = 3.6659U^* < 5.5U^*$.

Case ($w \in P^3$):

Again, we partition P^3 into 6 equal regions using bisectors of triangle T (See Figure 4). For any point w inside P^3_{13}, we assign cameras s_1 and s_3. Let us say that w' is the intersection point of the boundary of $D(c, 2R')$ with $\overrightarrow{c\ w}$. Similarly, suppose that w'' and s''_3 are intersection points of the boundary of $D(c, 2R)$ with $\overrightarrow{c\ w}$ and $\overrightarrow{c\ s_3}$, respectively. To obtain a bound on the uncertainty, we first establish a lower bound on $\sin(\angle s_1ws_3) = sin\,(Angle\,(r, \theta))$, followed by an upper bound on the product $d(s_1, w)d(s_3, w) = Mult(r, \theta)$. Finally, we show that both bounds are reached at the same point: $w = s''_3$.

For the remaining part of the proof, we will use polar coordinates as shown in Figure 7. We define uncertainty function $Uncert(r, \theta)$ in polar coordinates as follows:

$$
\begin{aligned}
Uncert(r,\theta) &= \frac{Mult(r,\theta)}{sin(Angle(r,\theta))} \\
Mult(r,\theta) &= \sqrt{\left(r^2-4rR'\sin\theta+4R'^2\right)\left(r^2-4rR'\cos\left(\theta+\frac{\pi}{6}\right)+4R'^2\right)} \\
Angle(r,\theta) &= \frac{\pi}{3}+\arctan\left(\frac{2R'-r\sin\theta}{r\cos\theta}\right)+\arctan\left(\frac{2R'-r\cos\left(\theta+\frac{\pi}{6}\right)}{r\sin\left(\theta+\frac{\pi}{6}\right)}\right)
\end{aligned}
$$

where for all $w \in P^3_{13}$, $-\pi/6 \leq \theta \leq \pi/6$ and $2R' < r \leq 2R$.

First, we show that for any $w \in P^3_{13}$, $Mult(r, \theta) \leq Mult(2R, -\pi/6)$, then we show $\sin(Angle(r, \theta)) \geq \sin(Angle(2R, -\pi/6))$ where $(2R, -\pi/6)$ corresponds to s''_3.

By Euclid's exterior angle theorem (in any triangle the angle opposite the greater side is greater), we have $d(s_1, w) \leq d(s_1, w'')$ and $d(s_3, w) \leq d(s_3, w'')$. Therefore, $Mult(r, \theta) \leq Mult(2R, \theta)$. By the extreme value theorem: $Mult(2R, \pi/6) \leq Mult(2R, \theta) \leq Mult(2R, -\pi/6)$. Hence, for any $w \in P^3_{13}$, $Mult(r, \theta) \leq Mult(2R, -\pi/6)$ holds.

The angle $\angle s_1ws_3$ is always between $\angle s_1w's_3$ and $\angle s_1w''s_3$ where $\angle s_1w's_3 = 2\pi/3$. Therefore, $sin(\angle s_1ws_3)$ is lower bounded by $\min(\sin(\angle s_1w''s_3), sin(2\pi/3))$, i.e.

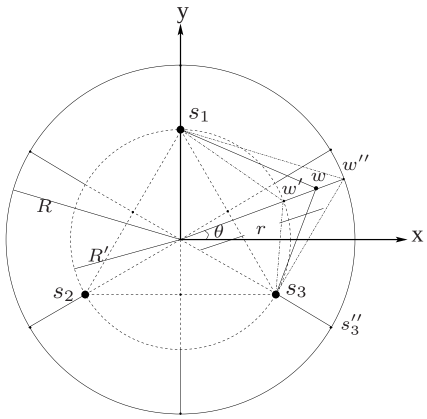

Fig. 7. In this figure, we represent w in polar coordinates (r, θ). For all $w \in P^2_{13}$ (See Figure 4), $\sin(\angle s_1 w s_3) \geq \sin(\angle s_1 s_3'' s_3)$ and $d(s_1, w)d(s_3, w) \leq d(s_1, s_3'')d(s_3, s_3'')$. Hence, $U(s_2, s_3, w) < 5.5U^*$.

$\sin(Angle(r,\theta)) \geq \min(\sin(Angle(2R,\theta)), \sin(2\pi/3))$. The function $Angle(2R,\theta)$ has its local minima and maxima at $\theta = -\pi/6$ and $\theta = \pi/6$, respectively and it is increasing in its domain. This can be shown by investigating the domain and roots of the first derivative of $Angle(2R,\theta)$. Therefore, $\sin(Angle(r,\theta)) \geq \min\left(\min\left(\sin\left(Angle(2R,-\frac{\pi}{6})\right), \sin\left(Angle(2R,\frac{\pi}{6})\right)\right), \sin(\frac{2\pi}{3})\right) = \sin\left(Angle(2R,-\frac{\pi}{6})\right)$.

Finally, $U(s_2, s_3, w) \leq \frac{Mult(2R,-\pi/6)}{\sin(Angle(2R,-\pi/6))} = 5.4989U^* < 5.5U^*$.

In conclusion, *p*laceSensors guarantees an uncertainty for all cases that is not greater than $5.5U^*$, i.e. $U(\mathcal{W}, \mathcal{S}) < 5.5U^*$. ■

Remark 1: We believe that the placement of the three sensors given in Algorithm *placeSensors* is optimal. To verify this, we numerically computed the error in localization as a function of the radius of the circle circumscribing the equilateral triangle formed by the three sensors. The minimum value was achieved at radius 1.25 which is very close to the radius used in the algorithm ($2\sqrt[3]{1/4} = 1.2599$). The small difference is due to discretization. To prove optimality, it remains to be shown that an optimal placement is symmetric around the center. We believe that this claim is true. However we do not have a proof of this statement.

As a comparison, we ran the ILP program for the case where the environment is a disk of radius two. Recall that the ILP takes a threshold as input, and computes a placement with the smallest number of sensors to achieve this threshold. It turns out that the precise threshold value is critical. The value given by the argument in Lemma 2, used in the analysis of the approximation algorithm is 5.4989. The number of sensors in the placement changes from three to four when the threshold changes from 5.499 to 5.498. The placement of the three sensors was identical to the one used in Lemma 2.

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 $(5.5, 3)$ 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 [15].

V. Conclusion

In this paper, we introduced a novel sensor placement problem where pairs of sensors are used to obtain estimates about a target's location. We presented a general solution based on integer linear programming and an approximation algorithm for the special case of bearing-only localization in the absence of obstacles. We are currently working on extending the approximation guarantees to more complex environments. Preliminary results can be found in [16].

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.
[3] A. Smith, H. Balakrishnan, M. Goraczko, and N. Priyantha, "Tracking moving devices with the cricket location system," in *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*. New York, NY, USA: ACM Press, 2004, pp. 190–202.
[4] A. Nasipuri and R. el Najjar, "Experimental evaluation of an angle based indoor localization system," in *4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2006, pp. 1–9.
[5] I. Rekleitis, D. Meger, and G. Dudek, "Simultaneous planning, localization, and mapping in a camera sensor network," *Robotics and Autonomous Systems*, vol. 54, no. 11, pp. 921–932, November 2006.
[6] S. N. Sinha, M. Pollefeys, and L. McMillan, "Camera network calibration from dynamic silhouettes," *cvpr*, vol. 01, pp. 195–202, 2004.
[7] D. Devarajan, R. J. Radke, and H. Chung, "Distributed metric calibration of ad hoc camera networks," *ACM Trans. Sen. Netw.*, vol. 2, no. 3, pp. 380–403, 2006.
[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] 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.
[12] 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.
[13] 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
[14] V. Isler, S. Khanna, J. Spletzer, and C. Taylor, "Target tracking with distributed sensors: The focus of attention problem," *Computer Vision and Image Understanding Journal*, no. 1-2, pp. 225–247, October-November 2005, special Issue on Attention and Performance in Computer Vision. [Online]. Available: http://www.cs.rpi.edu/ isler/new/pub/pubs/foaCVIU.pdf
[15] E. W. Weisstein, "Disk covering problem," MathWorld–A Wolfram Web Resource, http://mathworld.wolfram.com/DiskCoveringProblem.html.
[16] O. Tekdas and V. Isler, "Sensor placement for triangulation based localization," Rensselaer Polytechnic Institute, Department of Computer Science, Tech. Rep. 07-01, 2007.

Onur Tekdas is a PhD student in the Department of Computer Science and Engineering at University of Minnesota-Twin Cities. He received his MS degree (2008) in Computer Science from Rensselaer Polytechnic Institute and BS (2006) degree in Computer Engineering from Middle East Technical University, Turkey. His research interests are in robotics and sensor networks.

Volkan Isler is an Assistant Professor in the Computer Science Department, and a resident fellow at the Institute on Environment at the University of Minnesota. He obtained his MSE (2000) and PhD (2004) degrees in Computer and Information Science from the University of Pennsylvania. He obtained his BS degree (1999) in Computer Engineering from Bogazici University, Istanbul, Turkey. In 2008, he received the National Science Foundation's Young Investigator Award (CAREER). He is currently co-chairing IEEE Society of Robotics and Automation's Technical Committee on Networked Robots. His research interests are in robotics and sensor networks.