UC Berkeley

Title
The sensor selection problem for bounded uncertainty sensing models.

Permalink
https://escholarship.org/uc/item/7714j8bx

Journal
IEEE Transactions On Automation Science and Engineering, 3(4)

ISSN
1545-5955

Authors
Isler, V
Bajcsy, R

Publication Date
2006-10-01

Peer reviewed

The Sensor Selection Problem for Bounded Uncertainty Sensing Models

Volkan Isler
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY, 12180-3590
Email: isler@cs.rpi.edu

Ruzena Bajcsy
CITRIS and EECS
University of California, Berkeley, CA, 94720-1764
Email: bajcsy@eecs.berkeley.edu

Abstract

We address the problem of selecting sensors so as to minimize the error in estimating the position of a target. We consider a generic sensor model where the measurements can be interpreted as polygonal, convex subsets of the plane. In our model, the measurements are merged by intersecting corresponding subsets and the measurement uncertainty corresponds to the area of the intersection. This model applies to a large class of sensors including cameras. We present an approximation algorithm which guarantees that the resulting error in estimation is within factor 2 of the least possible error. In establishing this result, we formally prove that a constant number of sensors suffice for a good estimate – an observation made by many researchers. We demonstrate the utility of this result in an experiment where 19 cameras are used to estimate the position of a target on a known plane. In the second part of the paper, we study relaxations of the problem formulation. We consider (i) a scenario where we are given a set of possible locations of the target (instead of a single estimate); and (ii) relaxations of the sensing model.

Note to Practitioners

This paper addresses a problem which arises in applications where many sensors are used to estimate the position of a target. For most sensing models, the estimates get better as the number of sensors increases. On the other hand, energy and communication constraints may render it impossible to use the measurements from all sensors. In this case, we face the sensor selection problem: how to select a "good" subset of sensors so as to obtain "good" estimates. We show that under a fairly restricted sensing model, a constant number of sensors are always competitive with respect to all sensors and present an algorithm for selecting such sensors. In obtaining this result, we assume that the sensor locations are known. In future research, we will investigate methods that are robust with respect to errors in sensor localization/calibration.

Index Terms

Sensor selection, localization, target tracking, sensor fusion, estimation

The Sensor Selection Problem for Bounded Uncertainty Sensing Models

I. INTRODUCTION

This paper addresses the problem of estimating the position of a target using measurements from multiple sensors. This problem, which we broadly refer to as the localization problem, is a fundamental problem in distributed and mobile sensing research. In robotics and automation, it is a prerequisite for many applications such as navigation, mapping and surveillance. In sensor network deployment, when a new sensor is added to the network, it must localize itself with respect to already deployed nodes. The problem also arises when tracking a moving target with a sensor network, where the network obtains measurements in order to estimate the target's location.

The localization problem can be formulated as follows: Let $x(t)$ be the state (typically position) of the target at time t. Suppose we have a motion model, $x(t+1) = F(x(t))$ for the target's state in the next time step[1]. In practice, $x(t+1)$ is only an approximation to the true state at $t+1$. However, if we have sensors which can estimate the target's state, we can improve our estimation by incorporating measurements from these sensors and obtain a better localization for $x(t+1)$.

The state of the art in localization is the probabilistic approach where the state $x(t)$ is represented by a probability distribution $p(x, t)$. The localization problem is solved using Bayesian filtering techniques where $p(x, t+1)$ is estimated using the motion model and a sensing model. Usually the sensing model is also given by a probability distribution $p(z|x)$ for the probability of obtaining the measurement value z given the state x of the target. This approach turned out to be very effective in solving the localization problem and has found widespread applications in robotics [2]–[5] and sensor networks [6]–[9].

The quality of localization improves with an increasing number of measurements from different sensors. Therefore, from a localization perspective, it is desirable to have many sensors involved in localization. On the other hand, sensor-networks have energy limitations. Taking measurements from many sensors and transmitting these measurements reduce the lifetime of the network. Even for sensing systems with no energy constraints, the network bandwidth may force us to use a limited number of measurements. In an application where many devices use the same sensor-network for localization, addressing the tradeoff between localization quality and the number of measurements becomes very important. Consequently, many researchers in the sensor-network community focused on this issue.

The reader is referred to [5] for information theoretic principles in sensor management. In [7], an *information driven*

sensor query approach was proposed. In this approach, at any given time, only a single sensor (leader) is active. After obtaining a measurement, the leader selects the most informative node in the network and passes its measurement to this node which becomes the new leader. In subsequent work, researchers addressed leader election, state representation, and aggregation issues [6], [10]. A sensor selection method based on the mutual information principle is presented in [9]. Recently, an *entropy based heuristic approach* was proposed [8] which greedily selects the next sensor to reduce overall uncertainty. In these approaches, the performance of the sensor selection algorithm is verified experimentally. The present work distinguishes itself from previous work by presenting analytical bounds on the performance of the proposed sensor selection algorithm. We focus on a restricted class of sensors (where the measurement uncertainty can be represented as a convex and polygonal subset of the plane). For this class of sensors, we present a polynomial-time algorithm whose performance is guaranteed to be within factor two of the optimal performance.

Recently, other aspects of tracking in sensor networks have received significant attention as well. In [11], the problem of estimating target's location and velocity using minimal information has been addressed. The problem of assigning n disjoint pairs of sensors to n targets so as to minimize the overall error in the estimation has been studied in [12]. A related line of research is cooperative localization, where a group of robots or network-nodes localize themselves by collecting information over the network [13]–[17].

In our present work, we address the sensor selection problem for the bounded uncertainty sensing model. In this model, the exact probability distribution $p(z|x)$ is unknown but for a given x, the set of possible values for the measurement z is bounded. Such models are useful for modeling complex sensing devices such as cameras (where the observations are accurate up to the pixel resolution) as well as networks of heterogeneous sensors where it is difficult to obtain an exact sensing model for each sensor. To address the tradeoff between sensing cost and utility, we start by formulating the sensor selection problem (SSP) as a bicriteria optimization problem. After observing that the general version of SSP is computationally hard, we focus on a geometric version where the sensor measurements correspond to convex, polygonal (possibly unbounded) subsets of the plane and the cost is measured by the number of sensors. The convexity assumption is valid for many different types of sensors as discussed in Section II-A. For this version, we present an approximation algorithm which selects a given number of sensors and guarantees that the quality of the localization is within a factor two of the optimal choice. We also establish that a constant

---

[1]For notation simplicity, let us assume that time is discrete.

number of sensors can guarantee performance with respect to *all* sensors – formally proving the observation: *a small number of sensors is sufficient to obtain good estimates*. In Section III, we present an application of this theorem where the objective is to estimate the location of a target on a known plane using cameras.

In the SSP formulation, we assume that an initial estimate of the target's location is given and that each measurement corresponds to a convex, polygonal subset of the plane. In the second part of the paper, we relax these assumptions. First, we study the *online SSP* problem where only a set of possible locations of the target is known (Section IV-A). In Section IV-B, we discuss relaxations to other sensing models.

After a brief overview of the notation used in the paper, we start with a formalization of the sensor selection problem.

Notation

Throughout the paper, 2^S denotes the set of all subsets of S. We call a set $S' \subseteq S$ as a k-subset of S if $|S'| = k$, i.e. the cardinality of S' is k.

The Euclidean distance between points x and y is denoted by $d(x, y)$. For a set S and a point x we define $d^+(x, S) = \max_{y \in S} d(x, y)$ and $d^-(x, S) = \min_{y \in S} d(x, y)$

II. THE SENSOR SELECTION PROBLEM

The general problem we study is as follows: We are given a set $S = \{s_1, \ldots, s_n\}$ of sensors and their locations which can estimate the position of the target. We are also given an estimate x of the target's location. Such an estimate is typically obtained by running a filter on the target's position. Our goal is to choose the "best" set of sensors to obtain a better estimate of the true location. Ideally, we would merge measurements from all sensors but this would be too costly. In order to model this trade-off, we need:

(i) A utility function, $Utility : 2^S \to \mathbb{R}$, which returns the utility of measurements obtained by each $S' \subseteq S$ when the robot is at x. This utility is typically related to the uncertainty of the measurements as a function of target/sensor geometry.

(ii) A cost function, $Cost : 2^S \to \mathbb{R}$, which returns the cost of taking measurements from each $S' \subseteq S$. This cost may incorporate, for example, the number of sensors used, the cost of transmitting the measurements, etc.

The Sensor Selection Problem (SSP) is to choose a subset $S' \subseteq S$ which maximizes utility and minimizes cost. To solve this bicriteria optimization problem we define a family of optimization problems

$$SSP(K) : \arg \max_{S' \subseteq S, Cost(S') \leq K} Utility(S') \qquad (1)$$

whose solution returns the best set of sensors for a given cost budget K.

In a typical sensor-network setting, the number of sensors is expected to be very large. Therefore, it is desirable for any algorithm for SSP to run in polynomial time in the number of sensors. For arbitrary cost and utility functions, it is not too difficult to see that SSP is a hard problem. As an example, consider perhaps the simplest scenario one can

imagine: Suppose we are given a utility value $u(s_i)$ and a cost value $c(s_i)$ for each sensor $s_i \in S$. Let $Cost(S') = \sum_{s_i \in S'} c(s_i)$ and $Utility(S') = \sum_{s_i \in S'} u(s_i)$. Even for these simple functions, SSP is NP-hard as it is equivalent to the well known NP-complete KNAPSACK problem [18].

In the next section, we study a geometric version of SSP, called k-SSP. For a given budget k, the cost $Cost(S')$ will be zero if $|S'| \leq k$ and infinite otherwise. This is motivated by the typical scenario where many robots/devices operate in the same workspace and localize themselves using the same sensor network. In this case, the network manager can put an upper bound on the number of sensors queried by each robot.

The utility model is formalized next.

A. Utility model

We consider the planar setting where the state of the target is given by its coordinates.

Our sensing model is as follows. The set of sensors is given by $S = \{s_1, ..., s_n\}$. Each sensor s_i returns an estimate $\mu_i(x) \subseteq \mathbb{R}^2$ for the position of x. We make two assumptions about μ_i:

(i) $\mu_i(x)$ is a convex, polygonal subset of the plane for all x. That is, $\mu_i(x)$ can be written as an intersection of a finite number of half-planes. Note that our model allows $\mu_i(x)$ to be unbounded, e.g. a half-plane.

(ii) $x \in \mu_i(x)$. The true location of the target is contained in the measurements.

Given a set $S' \subseteq S$ of sensors, the measurement obtained by all sensor in S' is given by $\mu(S', x) = \cap_{s_i \in S'} \mu_i(x)$. We will take $Utility(S')$ as inversely proportional to the error in estimating the position given by $Area(\mu(S', x))$ for a given x. Note that $\mu(S', x)$ is obtained by intersecting convex sets and hence convex.

As discussed below, Assumption (i) readily holds for many sensors. We will also revisit this assumption in Section IV-B and show how it can be relaxed for other types of sensors. The second assumption is merely for the area measure to make sense: consider three measurements (polygons) M_1, M_2, M_3 where M_1 and M_2 have a large intersection, M_2 and M_3 have a small intersection and M_1 and M_3 do not intersect. If such measurements can be obtained from the sensors, sensor selection becomes an ill-posed problem. Finally, it is worth noting that we make no assumptions about the uniformity of sensors. As long as the two assumptions are satisfied, each sensor may have a different model.

These two assumptions for the sensing model hold for a large class of sensors. A typical example is a spherical perspective camera commonly used in robotics applications.

The error model for such cameras is illustrated in Figure 1 where the cameras are represented by their projection centers c_1, c_2 and their imaging circles C_1, C_2. For a given world point x, its projection onto camera i is the intersection of the ray $c_i x$ with the imaging circle C_i. We can think of each camera as measuring the angle α_i corresponding to the projection of x_i. In the error-free case, we can compute the position of x by measuring α_1 and α_2 from two cameras[2] and

---

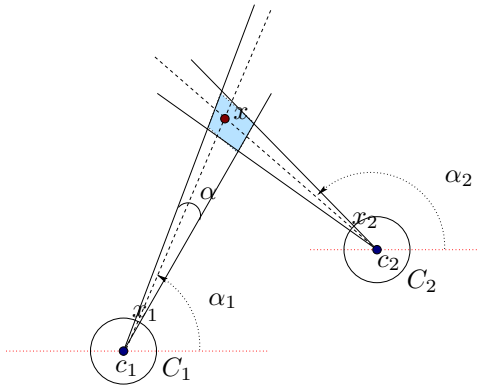[2]Excluding the degenerate case where the cameras and x are collinear.

Fig. 1. An error model for spherical perspective cameras.

intersecting the two rays. However, due to device limitations such as finite resolution, we can measure the angle α_i up to an additive error $\pm\alpha$ and each measurement can be interpreted as a cone (instead of a ray) which contains the target's location x. Using two cameras we can estimate the target's position by intersecting the two cones (shaded area in Figure 1).

For a spherical camera s_i on the plane, clearly $\mu_i(x)$ is a convex, polygonal subset of the plane. Further, if we assume that either the cameras have infinite range or we can detect cameras that do not see x and remove them from the set of available cameras, we also have $x \in \mu_i(x)$ for all $s_i \in S$. Throughout the paper, we will use this scenario to demonstrate our results: The workspace is the entire plane and we have n spherical perspective cameras with infinite range placed arbitrarily in the workspace.

We conclude this section with a summary of the assumptions in our model.

- We assume that the locations of the sensors are known precisely. In addition, an estimate of the target's location is available.
- Each sensor measurement corresponds to a convex, polygonal subset of the plane which contains the target's true location.
- For a given k, we assume that the cost of querying up to k sensors is zero. The cost of querying more than k sensors is assumed to be infinite.

B. Importance of good sensor selection algorithms

Before setting out to design a sensor selection algorithm, one must ask whether the choice of sensors is a crucial factor in obtaining a good estimate. As an example, consider the scenario in Figure 2 where 25 spherical cameras with $\alpha = 2$ degrees estimation error are placed uniformly on a planar area. Now suppose, we would like to select two cameras to obtain the position of the target, shown as a blue dot. In the figure, the estimates obtained by the best and worst pairs of cameras are shown. The area of the worst estimate is more than a thousand times larger than the area of the best estimate!

As illustrated in the above example, choosing the right set of sensors is clearly a crucial factor for the quality of the estimate. On the other hand, it has been observed many times that a small number of sensors suffices for a good estimate.

This is illustrated in the simulation shown in Figure 2. In this simulation, we used the sensors shown on the left in Figure 2. For 100 random target locations, we selected $k = \{2, 3, 4\}$ best sensors. Next, for each target location, we computed the area of the estimate from all 25 sensors and normalized the estimate from k sensors by dividing to this area. Each histogram in the figure corresponds to a different k.

As shown in the last histogram, for these uniformly distributed samples, the estimates obtained by four best sensors are as good as the estimates obtained from *all* sensors! Hence, the lifetime of the network can be significantly increased by restricting the number of active sensors to a small number without losing too much from the quality of the estimates. In the next section, we formally prove that this intuition is always correct.

C. A 2-approximation for k-SSP

In this section, we present a 2-approximation algorithm for the k-SSP problem. We will use the following standard definitions for approximation algorithms.

Definition 1: An α-approximation algorithm for the k-SSP problem is an algorithm whose running time is polynomial in the number of sensors and which chooses k sensors such that the error in estimating the position of the target is within a factor α of the error resulting from the optimal choice of k sensors.

Definition 2: An (α, β)-approximation algorithm for the k-SSP problem is an algorithm whose running time is polynomial in the number of sensors and which chooses βk sensors such that the error in estimating the position of the target is within a factor α of the error resulting from the optimal choice of k sensors.

Recall that in the k-SSP problem, we are given a set S of n sensors, and we would like to select k of them. Let $\mathcal{R} = \{R : R \subseteq S, |R| \leq k\}$ be the set of all possible choices. Note that $|\mathcal{R}|$ is exponential in k and as k grows large, enumerating $\mathcal{R}$ to pick the best set becomes infeasible. Let S^* be the optimal choice given by $\arg\min_{R \in \mathcal{R}} Area(\mu(R))$. We will show how to pick a set S' from $\mathcal{R}$ in polynomial time such that $Area(\mu(S'))/Area(\mu(S^*)) \leq 2$.

To obtain the approximation algorithm, we will utilize the notion of a Minimum Enclosing Parallelogram (MEP). Given a polygon X, MEP of X is a parallelogram which has the smallest area among all parallelograms that contain X. Let C be a convex polygon and let MEP be the minimum enclosing parallelogram of C whose vertices are $v_1, \ldots, v_4$ in counterclockwise order. Let $e_i = (v_i, v_{i+1})$, $i = 1, \ldots, 4$ be the sides of MEP (see Figure 3)[3]. The following properties of MEP are adapted from [19] (see also [20]).

Property 1 ([19]): Either e_1 or e_3 contains a side of C. Similarly, either e_2 or e_4 contains a side of C.

Property 2 ([19]): Let $f_i = C \cap e_i$, $i = 1, \ldots, 4$. There exists a line l parallel to e_1 and e_3 such that $f_2 \cap l \neq \emptyset$ and $f_4 \cap l \neq \emptyset$. Similarly, there exists a line m parallel to e_2 and e_4 such that $f_1 \cap m \neq \emptyset$ and $f_3 \cap m \neq \emptyset$.

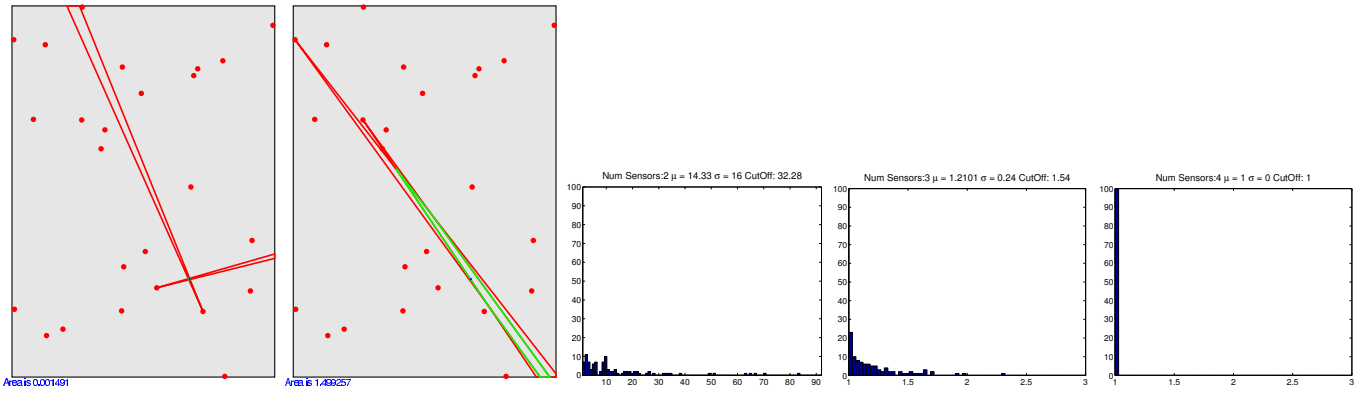[3]Throughout the paper we will use an acyclic ordering of vertices and let $v_{n+1} = v_1$

Fig. 2. **Left two figures:** The best (left) and the worst (right) sensor choices and the corresponding estimate for an arbitrary target location. The sensor locations were selected uniformly at random. **Right:** Histograms of the estimates obtained by best $k = \{2, 3, 4\}$ sensors. The numbers in the horizontal axis are obtained by dividing the area of the estimation from best k sensors by the area of the estimation from all sensors. The cut-off values correspond to the x-values where the cumulative density reaches 90%.
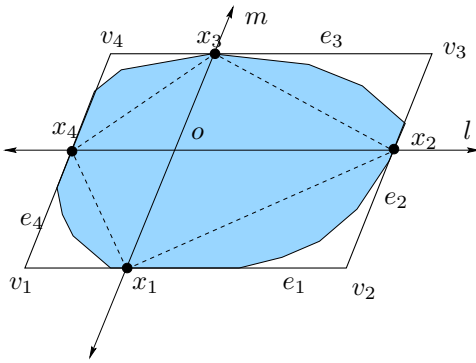

Fig. 3. The minimum enclosing parallelogram (MEP) of a convex polygon. One can always find a MEP such that each parallel side of MEP contains at least one side of the polygon. For example, the parallel sides $v_1 v_4$ and $v_2 v_3$ each contain a side of the polygon. However it is possible to have one side ($v_1 v_2$) contain a side and the other one ($v_3 v_4$) contain only a vertex.

Fig. 4. Notation for Lemma 4.

Using these two properties we can bound the area of MEP.

Lemma 3: $Area(MEP) \leq 2 \cdot Area(C)$.

Proof: Let $x_1 = f_1 \cap m$, $x_2 = f_2 \cap l$, $x_3 = f_3 \cap m$, $x_4 = f_4 \cap l$. By Property 2, all x_i exist and they belong to both C and MEP. Let $o = l \cap m$, which by convexity of C also belongs to C. The lines l and m partition MEP into four parallelograms $P_1, \ldots, P_4$. They also partition C into four convex polygons $C_1, \ldots, C_4$ such that $C_i \subseteq P_i$. In Figure 3, $P_1 = v_1 x_1 o x_4$, $P_2 = v_2 x_2 o x_1$, and so on. Consider C_1 and P_1. Since all of x_1, o, x_4 belong to C_1 we have $\Delta_1 \subseteq C_1$ where Δ_1 is the triangle whose vertices are x_1, o and x_4. Hence we have $\frac{Area(P_1)}{2} = Area(\Delta_1) \leq Area(C_1)$. A symmetric argument holds for each C_i, P_i pair. Therefore $Area(MEP)/2 = \sum_i Area(\Delta_i) \leq Area(C)$. ∎

The following result will be central to our sensor selection algorithm.

Lemma 4: Let x be a given target position and $S = \{s_1, \ldots, s_n\}$ be the set of sensors. Let $C = \mu(S)$ be the measurement obtained by all sensors. If C is bounded, then there exists a set $S' \subseteq S$ with $|S'| \leq 6$ such that $\mu(S') \leq 2 \cdot \mu(S)$.

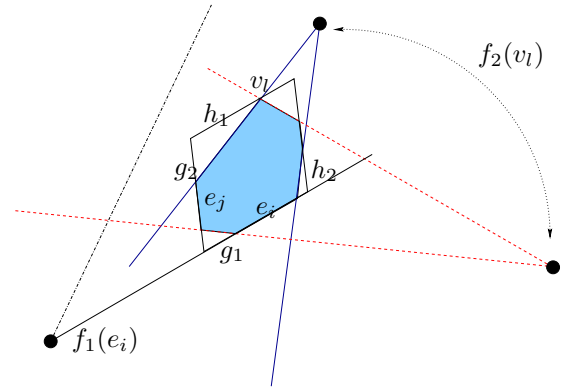*Proof:* We prove the lemma constructively.

First, we compute C by intersecting $\mu(s_i)$ in an arbitrary order. Let $V = \{v_i\}$ be the set of vertices and $E = \{e_i\}$ be the set of edges of C. Each edge e_i is contained in a unique line denoted $l(e_i)$. Let $f_1 : E \to S$ be a function that associates each edge e_i of C to the sensor that constrains x to be on one side of $l(e_i)$ (see Figure 4). Using f_1 we can also obtain a function $f_2 : V \to S \times S$ which maps each vertex $v_i = (e_i, e_{i+1})$ to $(f(e_i), f(e_{i+1}))$ such that $l(e_i) \cap l(e_{i+1}) = v_i$.

Next, we construct the minimum enclosing parallelogram MEP of C using the algorithm in [19]. By Property 1, two sides of C, say e_i and e_j are contained in the two sides of MEP, say g_1 and g_2 respectively. We will start with $S' = \emptyset$ and add $f_1(e_i)$ and $f_1(e_j)$ to S'. Let h_1 (resp. h_2) be the side of MEP parallel to g_1 (resp. g_2). The intersection of h_1 with C is either an edge or a vertex of C. If it is an edge, say e_k we add $f_1(e_k)$ to S'. Otherwise, if it is a vertex, say v_l we add $f_2(v_l)$. Similarly, we find one or two sensors for h_2 and add them to S'. At the end of this process, at most 6 edges will be added to S'. Note that at least two sides of the MEP contain an edge of C by Property 1.

If $|S'| = 4$, then $\mu(S') = Area(MEP)$. In all other cases, $\mu(S') < Area(MEP)$, which, together with Lemma 3 proves the lemma. ∎

We are now ready to present the sensor selection algorithm.

Theorem 5: There exists a polynomial time 2-

approximation algorithm for k-SSP.

Proof: Let $Area(X)$ denote $Area(\mu(X))$ for each subset X of S. First we observe that the error has a monotonicity property: for two sets $S_1 \subseteq S_2$, $\mu(S_2) \subseteq \mu(S_1)$. This is because the uncertainty region is computed by taking intersections of convex uncertainty regions of sensors. Hence $Area(S_2) \subseteq Area(S_1)$ and therefore, for a budget of k, there is an optimal solution which chooses exactly k sensors.

Let $l = \min\{k, 6\}$. We enumerate all l-subsets of S and pick the best among them. Clearly, the number of subsets considered is $O(n^6)$. Next, we show that this will yield a 2-approximation.

Let S^* be the optimal solution for choosing k sensors. If $k \leq 6$, since we exhaust all possible selections, the algorithm is optimal. If not, let S' be the 6-subset chosen by the algorithm. We have $Area(S^*) \geq Area(S)$ by monotonicity and $Area(S) \geq Area(S')/2$ by Lemma 4.

Running Time: Intersection of m hyperplanes can be found in $O(m \log m)$ time [21] and the area of intersection can be found in $O(m)$ time – by triangulating the (convex) polygon and computing the area of each triangle. Therefore the running time of the algorithm is $O(n^l \times ml \log(ml)) = O(n^l)$ where m is the maximum number of hyperplanes to define the sensor's estimate (e.g. $m = 2$ for cameras). Note that l is a constant. In fact, if $k \geq 6$ we can solve the problem in $O(mn \log(mn))$ time, by incorporating information from all n sensors, computing the MEP in linear time and choosing 6 sensors from MEP in constant time. ∎

Corollary 6: For any given set S of sensors, only 6 sensors are sufficient to approximate the utility of S within a factor of 2.

D. A distributed implementation

In a network of n sensors, by Corollary 6 we can restrict our attention to a small (polynomial in n) number of subsets of sensors and obtain reasonably good estimates. For stationary sensor-networks (such as a typical camera-network), this allows us to preprocess the workspace and generate a look-up table that can be used for a distributed implementation, as follows:

Followed by network deployment and localization, we build a table which stores the best choice of sensors for every position in the workspace. Such a table is shown in Figure 5 where the choice of pairs of sensors is color-coded. If possible, this computation can be performed off-site and the table can be uploaded to the nodes. During target tracking, the choice of sensors is obtained through lookup queries.

E. Comparison with a greedy algorithm

The running time of the 2-approximation algorithm is not monotone in k. For choosing k sensors from a set of n, the running time is in the order of n^k for $k < 6$. For greater values of k though, the running time drops to $n \log n$ as discussed in the proof of Theorem 5.

In an application where less than six sensors must be allocated per target, the $O(n^5)$ running time may be prohibitive for a real time implementation. One possible solution is to prepare
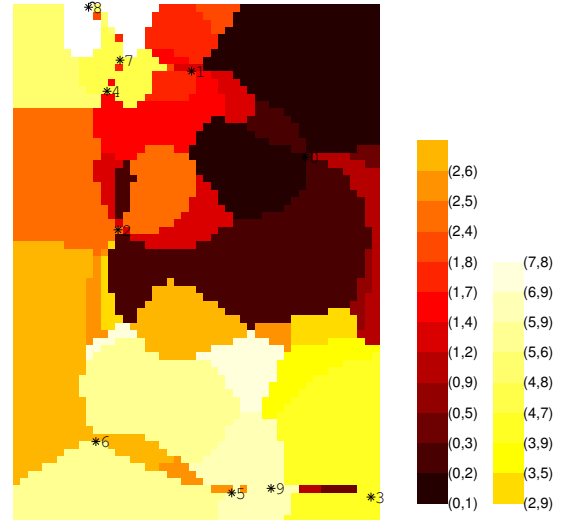

Fig. 5. A partition of the workspace. The color of each region indicates the best pair of cameras for that region as shown by the legend on the right.

a look-up table offline as described in the previous section. However, this may not always be possible. For such cases, we have investigated the performance of a greedy algorithm via simulations. At each iteration, the greedy algorithm selects the sensor which yields the greatest decrease in the estimation area. The results are shown in Figure 6 where the histograms of the ratio of the area of the estimation from the greedy algorithm to the best choice required by the 2-approximation algorithm are plotted. In this simulation, we used 25 randomly placed sensors and selected $k = \{2, 3, 4\}$ of them to estimate the position of 100 uniformly placed points.

Even though there are instances where the greedy algorithm performs poorly, on the average its performance is comparable to the best choice. Hence, in real-time applications where a small number of sensors need to be allocated, a greedy algorithm may also be used.

III. AN APPLICATION: ESTIMATING THE LOCATION OF A TARGET ON A KNOWN PLANE

In this section, we present an experiment which demonstrates our main result (Theorem 5). The goal of the experiment is to localize a point in 3D space from images. In general, the uncertainty that corresponds to a measurement from an image corresponds to a 3D cone and our result does not apply directly. However, there are many computer vision applications, such as tracking objects from satellites, where it is desirable to estimate the location of a target on a known plane (e.g. ground). If such additional information is available, the measurements become 2D polygons given by the intersection of 3D cones and the known plane. In this case, our result applies readily. To simulate such a scenario, we used a 19-camera setup located in our lab. The cameras are calibrated [22]. Hence, we know their locations and projection matrices with respect to a common world frame (Figure 7). For this experiment, we put a planar calibration pattern approximately 2-3 meters away from the cameras and took images with a resolution of 640×480 (Figure 9).
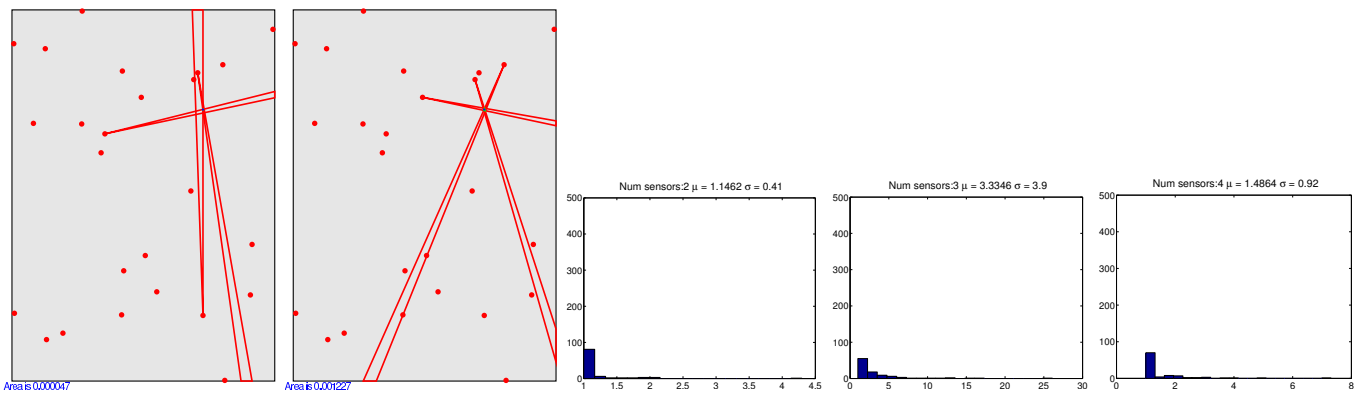
Fig. 6. **Left two figures:** An instance where the greedy choice (left) differs from the optimal choice (right). The area of the workspace is normalized to 100 units. **Right:** Comparison of the greedy choice of $k = \{2, 3, 4\}$ sensors with the best choice of k sensors. The horizontal axes indicate the ratio of the areas obtained by the greedy choice and the best choice. Even though the greedy algorithm occasionally performs poorly, in this simulation it has a good overall expected performance.

Our experiment started with obtaining the parameters of the plane. For this purpose, we computed corresponding pixels in different images by extracting corners from the images of calibration pattern. Afterwards, we computed 3D locations of these corners (which lie on the calibration plane) using stereo triangulation. To achieve robustness, we computed 3D points using a number of pairs of cameras and fitted a plane to these points using a robust, RANSAC based plane fitting method [23]. See Figure 7 for the estimated parameters and Figure 9 for sample images taken from the cameras.

To obtain the position of the target automatically, we used a corner estimator [22]. However, as it can be seen from Figure 7 (bottom-right), it is very difficult to estimate the precise location of the corner. Hence, we used an uncertainty parameter of $\alpha = \pm 4$ pixels[4]. These uncertainties correspond to 3D cones whose intersections with the known plane gives us the polygons shown in Figure 10.

Finally, to estimate the position of the target, we intersected the uncertainty polygons. The results are shown in Figure 11. The best choice for a pair of cameras is the pair (4,17) which gives an uncertainty of approximately $9mm^2$. The worst choice, on the other hand, is given by cameras (18, 19) with an error of $729mm^2$. See also Figure 9 for corresponding images. The best choice of three cameras is (4, 11, 17) with an area of $8mm^2$ and the worst choice is $(17, 18, 19)$ with an error of $670mm^2$. It is worth noting that the intersection of *all* polygons is the same as the intersection of the best three polygons (4, 11, 17) which is in agreement with Corollary 6.

IV. EXTENSIONS

In this section, we present two extensions to the model used to establish Theorem 5. The first extension is regarding the knowledge about the target's position. In the previous section we studied the problem of selecting a set of sensors for a given target location x. In the next section, we address the problem of selecting the sensors when x is not given. Instead, we are given an uncertainty region U that contains the true location of the target.

[4]The value of α is determined empirically.

In Section IV-B, we address extensions to other sensing models. In particular, we discuss how to relax the assumption that the measurements correspond to convex and polygonal subsets of the plane. We start with the first extension.

A. SSP for Unknown Target Location

In this section, we address the problem of selecting the sensors when x is unknown. Let W be the workspace and suppose we are given an uncertainty region $U \subseteq W$ such that the true location $x \in U$. Such an uncertainty region is typically obtained by a filter which starts with an estimate (e.g. a ball of certain radius) of the target's initial position and propagates this uncertainty as the target moves (See Figure 12).

In this section, we consider an online version of k-SSP. In the online version, we are given a domain U where the target's true location lies. An online algorithm for SSP chooses the set of sensors which guarantee performance regardless of the true location of the target.

Formally, the Online k-SSP problem is defined as follows: We are given an uncertainty region U and a set of sensors. Let $\hat{x} \in U$ be the unknown true location of the target. Let $S^*(\hat{x})$ be the optimal choice of k sensors for $\hat{x}$. Let $S' \subseteq S$ be any choice of sensors with $|S'| = k$. We define the competitive-performance of S' as

$$c(S') = \max_{\hat{x} \in U} \frac{Area(\mu(S', \hat{x}))}{Area(\mu(S^*(\hat{x}), \hat{x}))} \qquad (2)$$

Establishing the competitive performance (or ratio) can be formulated as a game between two players. The first player is the sensor selection algorithm which picks the set of sensors S'. The second player is the adversary who tries to maximize the deviation from the optimal performance by choosing the true location $\hat{x}$. Our goal is then to establish the existence of a choice of sensors S' with small (preferably constant) competitive ratio. If there is such a choice, we could select a good set of sensors and obtain a good estimate of the true location $\hat{x}$ without an accurate guess of the true location.

A simple example shows that we can not establish a constant competitive ratio: suppose the uncertainty region U is given by the equilateral triangle ABC shown in Figure 13. Suppose
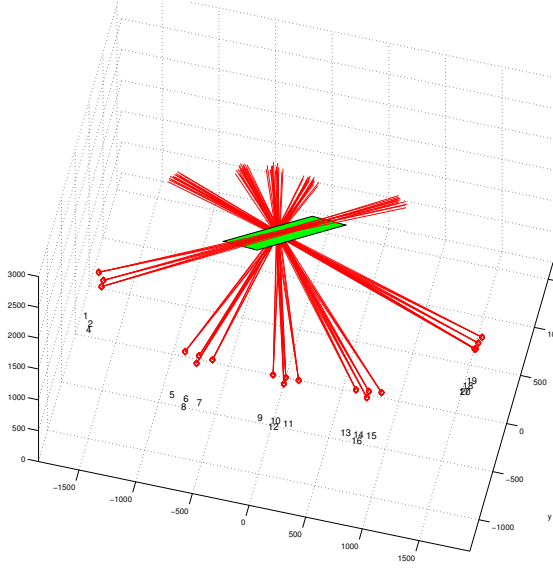
Fig. 7. The camera setup used for the experiments. A calibration pattern was used to estimate the parameters of a plane in a robust fashion (top). The projection centers of the cameras, location of the plane and uncertainty cones going through the target in 3D space. The units are in millimeters (bottom).

Fig. 8. A close-up image showing the location of the target (left). An 8×8 window showing the pixels around the projection of the target in an image. The precise location of the corner is difficult to obtain. (right)

Fig. 9. Images taken from cameras 4, 11, 17, 18, 19 (from top to bottom, left to right).

we have 6 sensors $S = \{a_1, a_2, b_1, b_2, c_1, c_2\}$ such that a_1 and a_2 are very close to the corner A and the angle $a_1 A a_2$ is $\frac{\pi}{2}$; b_1 and b_2 are very close to the corner B and so on. Consider the task of selecting two sensors from the sensor set S. There are two distinct choices: $\{a_1, a_2\}$ and $\{a_1, b_1\}$. All the remaining choices are symmetric. For each choice, there is a corner of the triangle, C, that is very far from either sensor. Hence, the adversary can choose $\hat{x}$ near C. For this value of $\hat{x}$, the optimal choice of sensors would be $S^* = \{c_1, c_2\}$. Let Z_{min} and Z_{max} be the closest and furthest distance between U and S. The competitive ratio (Equation 2) would be:

$$c(\{a_1, a_2\}) \approx c(\{a_1, b_2\}) \approx \frac{(Z_{max}\alpha)(Z_{max}\alpha)}{(Z_{min}\alpha)(Z_{min}\alpha)} = \left(\frac{Z_{max}}{Z_{min}}\right)^2$$

which can be made arbitrarily big by increasing the size of U. A symmetric argument can be made for any choice of sensors. Consequently, there exists no choice of sensors with a constant competitive ratio.

In fact, in the above example we observe that the term Z_{max}^2 is almost as big as the area of the uncertainty region U. Therefore, for any sensor selection algorithm to be effective,
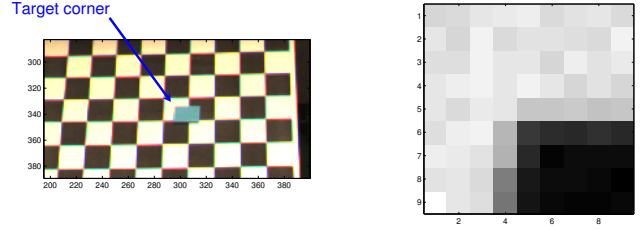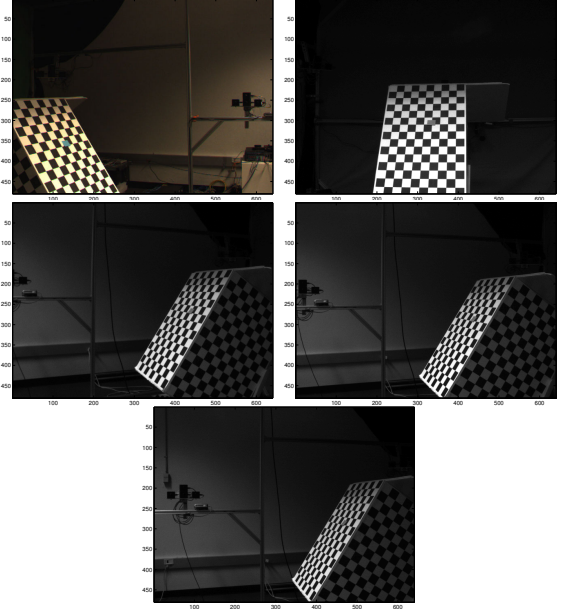
we need a fairly good estimate (small U) of where the target may be.

It is easy to see that the arguments above can be extended for the problem of selecting k sensors. We simply replace the triangle ABC with a uniform $(k + 1)$-gon. No matter which sensors are chosen, there will be one corner that is far from all the chosen sensors: the adversary chooses this corner as the true location for which the optimal choice includes the two sensors that are very close to this corner.

B. Extensions to other sensing models

Throughout the paper, we assumed that the sensor measurements are convex and polygonal. In this section, we show how our techniques can be extended to other sensing models. We investigate the following alternatives as shown in Figure 14.

Convex but non-polygonal measurements. In this case, the sensor measurements can be efficiently approximated by a convex polygon and our techniques apply.

Non-convex and non-polygonal measurements. In certain cases, such as range-and-bearing sensors, the measurement area can be efficiently approximated with a convex polygon – see [16] for a discussion. However, as in the case of range-only
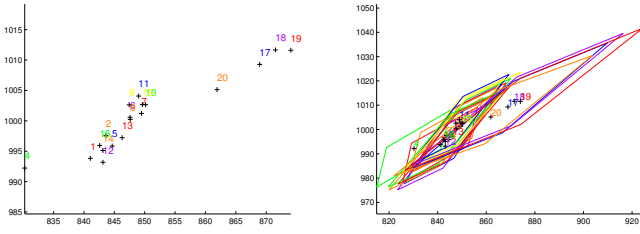
Fig. 10.   All units are in millimeters. **LEFT:** intersections of rays (originating from the camera centers and going through estimated target projections) with the plane. **RIGHT:** Intersections of uncertainty cones that correspond to an error of $\alpha = \pm 4$ pixels with the plane.

Fig. 12.   SSP for unknown target location: we are given the region U and we would like to choose sensors to guarantee performance regardless of the location of the target in U.
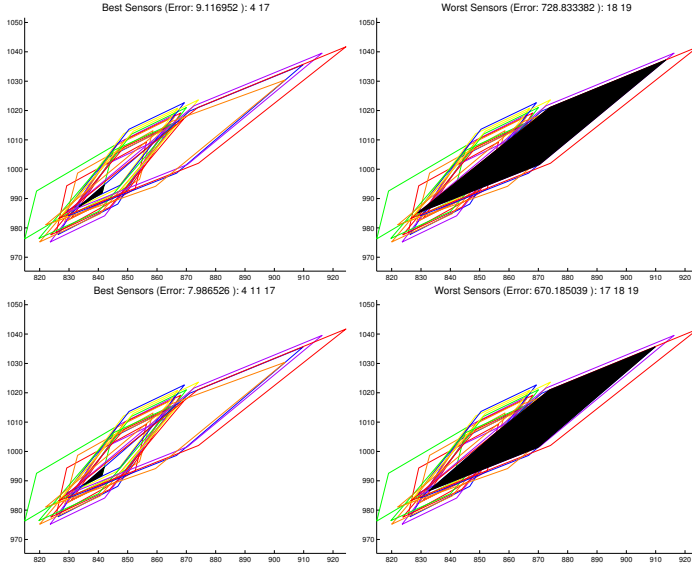

Fig. 11.   **TOP:** Best (left) and worst (right) choices for pairs of cameras. **BOTTOM:** Best (left) and worst (right) choices for triples of cameras. The shaded area corresponds to the intersection of the chosen cameras. See text for details.

Fig. 13.   For both choices of two sensors, $\{a_1, a_2\}$ and $\{a_1, b_1\}$, the adversary can select the true location near corner C. The shaded area corresponds to the error estimate from $\{a_1, b_1\}$. The optimal choice for this location is $\{c_1, c_2\}$ which results in a high competitive ratio.

of the plane. In our model, the measurements are merged by intersecting corresponding subsets and the measurement uncertainty corresponds to the area of the intersection. Under these assumptions, we presented a 2-approximation algorithm for choosing optimal sensors. In doing so, we formally proved an observation made by numerous researchers: *a small number of sensors is sufficient for a good estimate*. This is obtained by first bounding the area of the information from all sensors by its minimum enclosing parallelogram (MEP). Next, we showed that there are six sensors such that the intersection of the measurements from these sensors is contained in the MEP.

sensors, this may not be always possible. In Figure 14 (lower left), any convex polygon that contains the error annulus would fail to approximate the true error. One possible solution is to group m sensors (in the figure, a group of $m = 3$ sensors is shown) and to treat these groups of sensors as a single sensor whose error regions can be efficiently approximated by a convex polygon. If this is possible, we can use the 2-approximation algorithm defined in Section II-C to obtain a (2,m)-approximation algorithm for k-SSP (see definition 2).

V. CONCLUDING REMARKS

In this paper, we studied the problem of choosing k sensors so as to minimize the error in estimating the position of a target. We made two assumptions in our formulation:

(i) network localization has been performed (see e.g. [17]) and the locations of the nodes in the network are available; and,

(ii) an estimate of the target's location is available (possibly through a filter that estimates the position of the target as it moves).

We focused on a generic sensor model where the measurements can be interpreted as polygonal, convex subsets
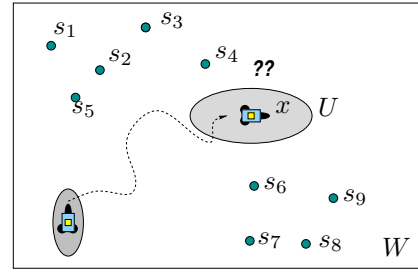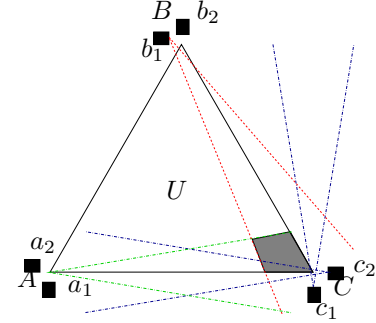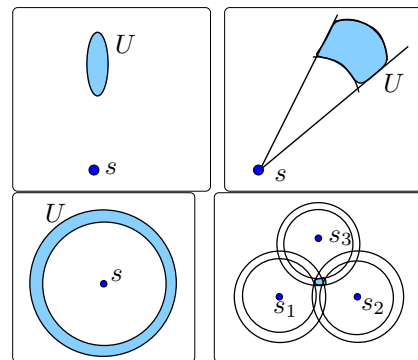

Fig. 14.   Common sensors where the convexity and/or linearity assumption fails. Top-left: convex but non-polygonal measurements Top-right: non-convex measurements which can be efficiently approximated (e.g a range-and-bearing sensor) Bottom-row: Non-polygonal and non-convex measurements which can not be approximated. The solution is to group the sensors (bottom-right).

We presented an application of this theorem for the task of obtaining the position of a target (on a known plane) using cameras.

In the second part of the paper, we discussed two extensions of the previous scenario. In the first extension, we considered the case where we are given a bounded area where the target may be located – instead of a single estimate of the target's location. For this case, we showed that the performance of any sensor selection algorithm is a function of the area of the uncertainty region. We also addressed relaxations of the sensing model and presented arguments for the existence of similar theorems for other sensing models.

In all of the algorithms above, we assumed that the sensor locations are known precisely. We plan to investigate the effect of localization errors over the entire network in our future work. Another research direction to extend the six-sensor result (Corollary 6) to higher dimensions. Future work also includes the problem of assigning *disjoint* sets of sensors to track multiple targets. The techniques presented in this paper may be easily extended to obtain an algorithm for this problem whose running time is $O(n^t)$ where n is the number of sensors and t is the number of targets – we simply try all possible ways of assigning up to 6 sensors to each target. Though applicable for a small number of targets, such a running time becomes prohibitive as the number of targets grow large and become comparable to the number of sensors. Unfortunately, the problem of assigning disjoint sets of sensors to many targets is NP-hard. Approximation algorithms for constrained error metrics and geometries can be found in [12]. We are currently working on extending these results to more general deployment scenarios.

ACKNOWLEDGMENT

REFERENCES

[1] V. Isler and R. Bajcsy, "The sensor selection problem for bounded uncertainty sensing models," in *The Fourth International Symposium on Information Processing in Sensor Networks (IPSN'05)*, UCLA, Los Angeles, California, USA, April 2005, pp. 151 – 158.

[2] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millenium*, G. Lakemeyer and B. Nebel, Eds. Morgan Kaufmann, 2002.

[3] ——, "A probabilistic online mapping algorithm for teams of mobile robots," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 335–363, 2001.

[4] 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.

[5] J. Manyika and H. Durrant-Whyte, *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach*. Ellis Horwood Series in Electrical and Electronic Engineering, 1994.

[6] F. Zhao, J. Liu, J. Liu, L. Guibas, and J. Reich, "Collaborative signal and information processing: An information directed approach," *Proceedings of the IEEE*, vol. 91, no. 8, 2003. [Online]. Available: citeseer.ist.psu.edu/article/zhao03collaborative.html

[7] J. Liu, J. Reich, and F. Zhao, "Collaborative in-network processing for target tracking," *EURASIP JASP*, vol. 4, no. 378-391, 2003.

[8] H. Wang, K. Yao, G. Pottie, and D. Estrin, "Entropy-based sensor selection heuristic for target localization," in *Proceedings of the third international symposium on Information processing in sensor networks*. ACM Press, 2004, pp. 36–45.

[9] E. Ertin, J. W. F. III, and L. C. Potter, "Maximum mutual information principle for dynamic sensor query problems," in *2nd International Workshop on Information Processing in Sensor Networks*, 2003.

[10] J. Liu, M. Chu, J. Liu, J. Reich, and F. Zhao, "Distributed state representation for tracking problems in sensor networks," in *Proceedings of the third international symposium on Information processing in sensor networks*. ACM Press, 2004, pp. 234–242.

[11] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus, "Tracking a moving object with a binary sensor network," in *Proceedings of the first international conference on Embedded networked sensor systems*. ACM Press, 2003, pp. 150–161.

[12] V. Isler, J. Spletzer, S. Khanna, and C. Taylor, "Target tracking in sensor networks: the focus of attention problem," *Computer Vision and Image Understanding Journal*, vol. 100, pp. 225–246, October 2005.

[13] L. Doherty, K. S. J. Pister, and L. E. Ghaoui, "Convex position estimation in wireless sensor networks," in *Proc. IEEE Infocom*, 2001.

[14] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Cooperative relative localization for mobile robot teams: An ego-centric approach," in *Proceedings of the Naval Research Laboratory Workshop on Multi-Robot Systems*, Washington, D.C., Mar 2003.

[15] J. Spletzer, A. Das, R. Fierro, C. Taylor, V. Kumar, and J. Ostrowski, "Cooperative localization and control for multi-robot manipulation," in *Proceedings of the Conference on Intelligent Robots and Systems (IROS01)*, 2001.

[16] J. Spletzer and C. Taylor, "A bounded uncertainty approach to multi-robot localization," in *Proceedings of the Conference on Intelligent Robots and Systems (IROS 2003)*, 2003.

[17] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys '04)*, Baltimore, MD, November 2004.

[18] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA, 1979.

[19] C. Schwarz, J. Teich, E. Welzl, and B. Evans, "On finding a minimal enclosing parallelgram," International Computer Science Institute, Berkeley, CA, Tech. Rep. tr-94-036, 1994. [Online]. Available: http://www.icsi.berkeley.edu/ftp/pub/techreports/1994/tr-94-036.ps.gz

[20] C. Schwarz, J. Teich, A. Vainshtein, E. Welzl, and B. L. Evans, "Minimal enclosing parallelogram with application," in *SCG '95: Proceedings of the eleventh annual symposium on Computational geometry*. ACM Press, 1995, pp. 434–435. [Online]. Available: http://doi.acm.org/10.1145/220279.220338

[21] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*. Springer, 1997.

[22] J.-Y. Bouguet, "Camera calibration toolbox for matlab."

[23] P. Kovesi, "Matlab functions for computer vision and image analysis."

Volkan Isler 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, exploration, motion planning), sensor-networks (deployment, target tracking and localization) and computer vision (tele-immersion, model reconstruction and segmentation).

Ruzena Bajcsy Ruzena Bajcsy is a Professor of Electrical Engineering and Computer Science at the University of California, Berkeley. Dr. Ruzena Bajcsy was appointed Director of CITRIS at the University of California, Berkeley on November 1, 2001 and stepped down in August 2005. Prior to coming to Berkeley, she was Assistant Director of the Computer Information Science and Engineering Directorate (CISE) between December 1, 1998 and September 1, 2001. She came to the NSF from the University of Pennsylvania where she was a professor of computer science and engineering.

Dr. Bajcsy received her masters and Ph.D. degrees in electrical engineering from Slovak Technical University in 1957 and 1967, respectively. She received a Ph.D. in computer science in 1972 from Stanford University.

Dr. Bajcsy has done seminal research in the areas of human-centered computer control, cognitive science, robotics, computerized radiological/medical image processing and artificial vision. She is a member of the National Academy of Engineering, as well as the Institute of Medicine. She has served as advisor to more than 50 Ph.D. recipients. In 2001 she received an honorary doctorate from University of Ljubljana in Slovenia. In 2001 she became a recipient of the ACM A. Newell award.