Optimal Placement and Selection of Camera Network Nodes for Target Localization

Ali O. Ercan[1], Danny B. Yang[2], Abbas El Gamal[1], and Leonidas J. Guibas[2]

[1] Dept. of Electrical Engineering, Stanford University, Stanford, CA 94305, USA.
aliercan@stanford.edu, abbas@ee.stanford.edu
[2] Dept. of Computer Science, Stanford University, Stanford, CA 94305, USA.
danny@riya.com, guibas@cs.stanford.edu

Abstract. The paper studies the optimal placement of multiple cameras and the selection of the best subset of cameras for single target localization in the framework of sensor networks. The cameras are assumed to be aimed horizontally around a room. To conserve both computation and communication energy, each camera reduces its image to a binary "scan-line" by performing simple background subtraction followed by vertical summing and thresholding, and communicates only the center of the detected foreground object. Assuming noisy camera measurements and an object prior, the minimum mean squared error of the best linear estimate of the object location in 2-D is used as a metric for placement and selection. The placement problem is shown to be equivalent to a classical inverse kinematics robotics problem, which can be solved efficiently using gradient descent techniques. The selection problem on the other hand is a combinatorial optimization problem and finding the optimal solution can be too costly to implement in an energy-constrained wireless camera network. A semi-definite programming approximation for the problem is shown to achieve close to optimal solutions with much lower computational burden. Simulation and experimental results are presented.

1 Introduction

A wireless sensor network (WSN) comprises a collection of many low cost, low-power nodes each with sensing, processing and communication capabilities. WSNs have many advantages over traditional sensing modalities including wide coverage, robustness, scalability, and the ability to observe large scale phenomena distributed over space and time [1, 2].

The scarcest resource in a WSN is energy, as typically nodes operate untethered. The limited battery life of a node imposes severe constraints on its communication and computation capabilities. Consequently, recent work on WSNs has focused mainly on very low data rate sensor nodes [3]. In many applications, however, high data rate sensors are needed to perform the desired tasks. The most notable example is video cameras, which are widely used for surveillance and monitoring. Current surveillance camera installations are expensive and use outdated infrastructure. All captured video data is shipped to a central station for human operators to watch which makes the system non-scalable. As a result, there is a growing need to develop less costly wireless networks of cameras with automated task-driven capabilities. Such development faces many challenges. First, current video cameras are expensive and have high power consumption. Second, video cameras are high data rate devices, so transmitting all the data is costly in terms of energy. Third, video processing algorithms are in general computationally expensive, require floating point arithmetic, and are costly to implement locally.

The camera cost and power problems can be addressed by recent advances in CMOS technology, which enable the integration of sensing, processing and communication [4]. It is currently feasible to design very low cost and power camera systems suitable for deployment in a wireless network. To address the communication and computation challenges facing the development of wireless camera networks, simple local processing algorithms that produce only the essential information needed for the network to collaboratively perform a task or answer a query are needed (e.g., see [5]).

Energy consumption can also be minimized by reducing the number of cameras used to answer a query. This can be achieved by judicious placement of the cameras with respect to the objects and selection of the best subset of cameras to collaboratively answer the query. A proper placement of cameras increases the accuracy of sensing, while selecting a good subset allows for efficient sensing with little performance degradation relative to using all the cameras. Selection also allows the network to scale to large numbers of nodes because of the savings in communication, computation, and sensing.

In this paper, we investigate the problems of placement and selection of camera nodes in order to minimize the localization error for a single object. Localizing an object is important in many applications such as tracking, surveillance, and human computer interaction. For example, if we could localize an object at every time step, the tracking and correspondence problems become trivial. Very accurate localization is also important in several robotics applications, such as navigation through a complex environment or controlling an end effector to perform a delicate task. Specifically, we focus on 2-D object localization, i.e., location on the ground plane, because this is the most relevant information for many real world applications. We assume that the cameras are placed horizontally around a room. The local processing framework in [5] is used to reduce the image to a scan-line and only the center of the detected object from each camera is communicated to the central processor. Given these noisy measurements and the object prior distribution, the minimum mean squared error (MSE) of the best linear estimate of the object location in 2-D is used as a metric for placement and selection. To find the best camera placement, we optimize this metric with respect to the camera positions. For a circularly symmetric object prior distribution and sensors with equal noise, we show that a uniform sensor arrangement is optimal. Somewhat surprisingly, we establish that the general problem is equivalent to solving the inverse kinematics of a planar robotic arm which can be solved efficiently using gradient descent techniques. We then devise a semi-definite programming approximation of the optimal solution for the selection problem. We show that this method performs close to optimal and outperforms naive heuristics (e.g., picking greedily or the closest or uniform sensors).

The rest of the paper is organized as follows: In Sect. 2, we review related sensor networks, computer graphics and computer vision work. In Sect. 3, we introduce the camera model and define the placement and selection problems. In Sect. 4, we derive the optimal solution for the placement problem. In Sect. 5, we develop an approximate solution to the selection problem and compare our approximation method to other heuristics and to the optimal solution, both in simulation and experimentally. Section 6 discusses how to handle some non-idealities. Finally, in Sect. 7, we conclude.

2 Related Work

Sensor placement and selection have been addressed in the sensor networks, computer vision, and computer graphics literature. Selection has been studied in wireless sensor networks with the goal of decreasing energy cost and increasing scalability. Viewpoint selection, or the next best view, has been studied in computer graphics and vision for picking the most informative views of a scene. We summarize the work related to this paper in this section.

Sensor placement: Camera placement has been studied in computer vision and graphics. In photogrammetry [6–8], the goal is to place the cameras so as to minimize the 3D measurement error. The error propagation is analyzed to derive an error metric that is used to rank camera placements. The best camera placement is then solved numerically. The computational complexity of this approach only allows solutions involving a few cameras. In our approach we simplify the camera model, derive the localization error analytically as a function of camera places and minimize it to find the best placement. This is computationally lighter compared to above numerical methods. In [9] the problem of how to position (general) sensors with 2-D measurement noise to minimize the overall error is investigated. The paper also presents an algorithm to compute the optimal sensor placement. Our measurements are 1-D after local processing and we pose the placement problem as a special case of classical inverse kinematics problem. The emphasis of our work is also on selection, which due to its combinatorial nature is different from placement.

Sensor Selection: In [10] a technique referred to as IDSQ is developed to select the next best sensor node to query in a sensor network. The technique is distributed and uses a utility measure based on the expected posterior distribution. However, expected posterior distribution is expensive to compute because it involves integrating over all possible measurements. In [11] the mutual information metric is used to select sensors. This is shown to be equivalent to minimizing the expected posterior uncertainty, but with significantly less computation. The work in [12] expands on [11] and shows how to select the sensor with the highest information gain. An entropy-based heuristic that approximates the mutual information and is computationally cheaper is used. All these methods greedily select the next best sensor based on an entropy metric. In our work, we show how to select the next best *group* of sensors via combinatorial optimization. Also, in our approach the utility function is an analytical expression, which makes it much faster to evaluate than methods requiring numerical integration.

Camera Selection: Sensor selection has also been studied for camera sensors. In [13–15], a metric is defined for the next best view based on most faces seen (given a 3-D geometric model of the scene), most voxels seen, or overall coverage. The solution requires searching through all camera positions to find the highest scoring viewpoints.

In [5], a subset of horizontal camera sensors are selected to minimize the visual hull of all objects in the scene. This problem is solved using heuristics. These works use numerical techniques or heuristics to compute the viewpoint scores. We investigate a simpler problem, devise an analytical metric for it and find the optimal solution using combinatorial optimization techniques.

3 Problem Formulation

Given a number of noisy camera sensors, our goal is to localize an object as accurately as possible in the ground plane. We first describe the camera model and then formulate the utility metric used in determining the best placement and selection.

We assume that the cameras are aimed roughly horizontally. An overhead camera may have a less occluded view, and may allow better localization. But overhead cameras are often impractical to deploy and can only observe a small area limited by the field of view. Horizontal cameras are often more practical to install. They can also observe a larger area. Additionally targets may be easier to identify in a horizontal view.

As discussed earlier, the camera nodes in a WSN must perform cheap local image processing to reduce the video data. Following a similar approach to [5], we limit this processing to background subtraction. The background subtracted images are vertically summed and thresholded, as the horizontal location of the object in the camera plane is most relevant to the 2-D localization (see Fig. 1). We refer to the resulting linear bitmap as a “scan-line”. For the localization method that we use, only the center of the detected foreground object in the scan-line is communicated to the central processor. This reduces the data per frame from a 2-D array of pixel values to a single integer. This approach requires very little computation and communication and is thus compatible with a resource constrained WSN framework.

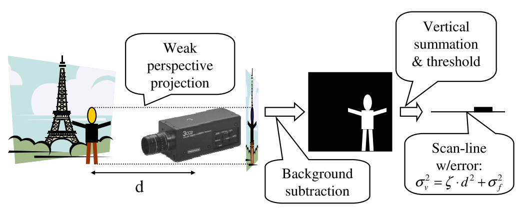


Fig. 1. Local processing at each camera.

We assume that the cameras are far enough from the object that they can be modeled by weak perspective projections. We assume that the measurement error variance is of the form $\sigma_v^2 = \zeta d^2 + \sigma_{\mathrm{f}}^2$, where d is the distance from the camera to the object (see Fig. 1). It can be shown that making camera noise variance dependent on d effectively models the weak perspective projection while allowing the usage of projective model in the equations. Our noise model also accounts for errors in the calibration of the cameras. Errors in the 2-D camera locations can be accounted for in σ_{f} and errors in the orientation can be accounted for in ζ.

Our specific problem is to localize one point object in a room with N cameras placed around its perimeter (See Fig. 2). As there is only one object to localize, we do not need to consider occlusions from other objects. For now, we also assume that there is no static object that occludes the view of the cameras. We discuss how static occlusions can be handled later in Sect. 6. The orientations of the cameras with respect to the abscissa are given by θ_i, $i = 1, 2, \ldots, N$. For simplicity, it is assumed that the point to be localized is in the FOV of all cameras. The consequences of limited FOVs is also addressed in Sect. 6. We assume that the mean μ_x and covariance Σ_x of the prior distribution of the object location are known.

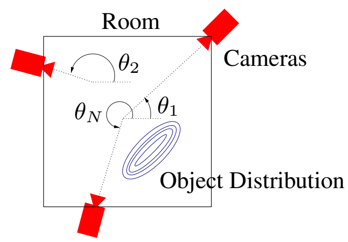


Fig. 2. Illustration of the problem definition.

We formulate the problem of camera placement for target localization in the framework of linear estimation. Given the first and second order statistics of the object prior and the camera noise parameters, we use the minimum MSE of the best linear estimate of the object location, which is a function of the camera orientations θ_i, as a measure for localization error. The best placement is then obtained by finding the camera orientations that minimize this metric. As explained in Sect. 4, the best orientations give the best positions of the cameras uniquely based on our assumptions for the placement problem. The same formulation is then used to investigate the camera selection problem, with some of the simplifying assumptions removed.

The measurement model is illustrated in Fig. 3. As explained before, we use the projective model in our equations. The weak perspective dependence of the measurements is hidden in the error model for the cameras. Let the object location be $x = [x_1, x_2]^{\mathrm{T}}$, and the measurements from the N cameras be $z = [z_1 \ldots z_N]^{\mathrm{T}}$. The relationship between the measurements and the object location is then given by

$$z = A\,x + v \ ,$$

where A is given by

$$A = \begin{pmatrix} -\sin\theta_1 & \cos\theta_1 \\ -\sin\theta_2 & \cos\theta_2 \\ \vdots & \vdots \\ -\sin\theta_N & \cos\theta_N \end{pmatrix} \ ,$$

and v is the measurement error.

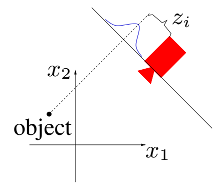


Fig. 3. The measurement model.

The best linear unbiased estimator for x is given by

$$\hat{x} = \mu_x + \Sigma_x A^{\mathrm{T}}(A\Sigma_x A^{\mathrm{T}} + \Sigma_v)^{-1}(z - A\mu_x) \ , \tag{1}$$

where μ_x is the mean and Σ_x is the covariance of the object location prior, and Σ_v is the covariance of the measurement noise.

Assuming the measurement noise is independent for different cameras, its covariance is

$$\Sigma_v = \mathrm{diag}(\sigma_{v_1}^2, \cdots, \sigma_{v_N}^2) \ .$$

The object prior can be assumed to be diagonal with a horizontal major axis without loss of generality because one can rotate everything, solve the problem and rotate everything back, so

$$\Sigma_x = \sigma_x^2 \mathrm{diag}(1, 1/\alpha) \ ,$$

where $\alpha \geq 1$. It can be shown that the MSE of the best linear estimate in (1) is

$$\mathrm{MSE} = \frac{4\left(\frac{\alpha+1}{\sigma_x^2} + \sum_{i=1}^{N} \frac{1}{\sigma_{v_i}^2}\right)}{\left(\frac{\alpha+1}{\sigma_x^2} + \sum_{i=1}^{N} \frac{1}{\sigma_{v_i}^2}\right)^2 - \left(\frac{\alpha-1}{\sigma_x^2} + \sum_{i=1}^{N} \frac{\cos 2\theta_i}{\sigma_{v_i}^2}\right)^2 - \left(\sum_{i=1}^{N} \frac{\sin 2\theta_i}{\sigma_{v_i}^2}\right)^2} \ . \tag{2}$$

Note that only the latter two terms in the denominator of (2) are functions of the camera orientations. Given fixed σ_{v_i}s, the MSE is minimized by setting these two squared terms as close to zero as possible. If these two terms are set to zero, we obtain a lower bound for MSE:

$$\mathrm{MSE} \geq \frac{4}{\frac{\alpha+1}{\sigma_x^2} + \sum_i \frac{1}{\sigma_{v_i}^2}} \ . \tag{3}$$

The cameras with lower noise achieve a smaller lower bound. In our noise model, closer cameras have less noise, so this means closer cameras achieve a smaller lower bound.

The two additional terms in the denominator of MSE (2) balance this low noise criterion with the direction criterion. Given σ_{v_i}s, (3) can be utilized to estimate the number of cameras that must be used to achieve a certain allowable error.

Given the above formulation, we define the following two problems:

Placement: Minimize (2) over $\theta_1, \theta_2, \ldots, \theta_N$. The resulting θ_is provide the locations of the cameras that result in the best localization error.

Selection: N cameras are previously placed. This means that the θ_is and σ_{v_i}s are fixed. We need to select the best subset of k cameras. As we are using a subset of the cameras, the summations in (2) also need to run over the selected set of cameras. The metric to minimize then becomes

$$\frac{4\left(\frac{\alpha+1}{\sigma_x^2}+\sum_{i\in S}\frac{1}{\sigma_{v_i}^2}\right)}{\left(\frac{\alpha+1}{\sigma_x^2}+\sum_{i\in S}\frac{1}{\sigma_{v_i}^2}\right)^2-\left(\frac{\alpha-1}{\sigma_x^2}+\sum_{i\in S}\frac{\cos 2\theta_i}{\sigma_{v_i}^2}\right)^2-\left(\sum_{i\in S}\frac{\sin 2\theta_i}{\sigma_{v_i}^2}\right)^2} \, . \tag{4}$$

where S is the set of selected cameras. Formally defined, the selection problem becomes: Minimize (4) over the set S such that $S \subseteq \{1, \ldots, N\}$ and $|S| = k$. The resulting set S gives the best selection.

In the following section we show that the optimal placement problem can be solved efficiently. An approximate solution to the selection problem is presented in Sect. 5.

4 Placement

We wish to find θ_is that minimize (2), given the object location prior statistics and the σ_{v_i}s for each camera.

The placement of the cameras is usually done before there is actually any object in the room. Therefore, it is natural to assume in this problem that the object prior is centered in the room. However, it does not have to be circularly symmetric, as people might tend to walk along certain directions more than others. For example, in a hallway, the major axis of the prior distribution would be aligned with the hallway.

Cameras are usually placed on the walls of the room, so the distance of the camera to the object prior cannot vary by much and can be approximated by a constant. Therefore, for the placement problem specifically, we are trying to come up with the best *orientations* of the cameras, and we assume a circular room for simplicity. As the cameras are fixed to the periphery and oriented towards the center, the orientations of the cameras (θ_is) uniquely determine their placement.

Note that all of the above assumptions are specific to the placement problem. In the selection problem (see Sect. 5), the circular room assumption and the centered object prior assumption are removed and our approach is applicable to any room shape, camera configuration and object prior. In Sect. 6 we give a recipe on how to handle general case for the placement problem with non-circular room and non-centered prior.

Given these assumptions, the camera noise parameters are constant. From (2) we note that only the last two terms of the denominator depend on the θ_is. Thus minimizing

(2) is the same as minimizing

$$\left(\frac{\alpha - 1}{\sigma_x^2} + \sum_{i=1}^{N} \frac{\cos 2\theta_i}{\sigma_{v_i}^2} \right)^2 + \left(\sum_{i=1}^{N} \frac{\sin 2\theta_i}{\sigma_{v_i}^2} \right)^2 \quad . \tag{5}$$

It is clear that (5) is bounded below by 0. The following subsections show when this can be achieved for the optimal camera orientations.

4.1 Symmetric Case

When the cameras have the same error variance, $\sigma_{v_i} = \sigma_v$ for all $1 \leq i \leq N$, and the object prior is circularly symmetric ($\alpha = 1$), the problem of minimizing (5) reduces to minimizing

$$\left(\sum_{i=1}^{N} \cos 2\theta_i \right)^2 + \left(\sum_{i=1}^{N} \sin 2\theta_i \right)^2 \quad . \tag{6}$$

This is equivalent to the norm-squared of the sum of N unit vectors with angles $2\theta_i$. Thus (6) is equal to zero when the θ_is are chosen uniformly between 0 and π. This leads to the intuitive conclusion that when the object prior is circularly symmetric and the cameras have the same amount of noise, uniform placement of cameras is optimal. An illustration of this result for 6 cameras is depicted in Fig. 4(a). The angles of the vectors are twice the orientation angles of the cameras. For example, for two cameras, an orthogonal placement of the cameras is optimal, so that the unit vectors are 180 degrees apart.

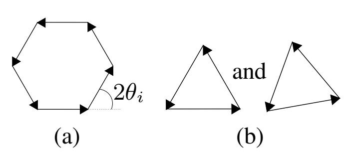


Fig. 4. (a) Uniform placement of 6 cameras that minimizes (6). (b) Locally optimal clusters are globally optimal. Relative orientations of clusters do not matter.

Uniform placement, however, is not the only optimal way to place the cameras. If we partition the vectors into subgroups and all subgroups of vectors sum to zero, then the combination of all the vectors also sums to zero. This means that we can cluster the cameras into (local) groups (with at least 2 cameras in each group) and solve the problem distributedly in each cluster. If each group finds a locally optimal solution, then the combined solution is globally optimal (see Fig. 4(b)). This is true no matter what the relative orientations between the groups of cameras are.

4.2 General Case

We now discuss the general placement problem, i.e., when $\alpha \neq 1$ and the σ_{v_i}s are not all equal. The problem corresponds to minimizing (5). Again this is the sum of N vectors, but the vectors can have different lengths $1/\sigma_{v_i}^2$. The MSE is minimized when the sum equals $-\frac{\alpha-1}{\sigma_x^2}$ (offset from zero) on the abscissa. Again, the resulting angles of the vectors are twice the optimal θ_i of the cameras (see Fig. 5).

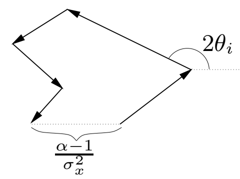


Fig. 5. An optimal solution.

This problem can be thought of as an inverse kinematics robotics problem. Our vectors describe a planar revolute robot arm with N linkages. The base of the robot arm is at the origin and it is trying to reach a point $-\frac{\alpha-1}{\sigma_x^2}$ on the abscissa with its end effector. If the σ_{v_i}s are ordered such that

$$\sigma_{v_N} \geq \sigma_{v_{N-1}} \geq \ldots \geq \sigma_{v_1} \quad ,$$

then any point in an annulus with inner and outer radii

$$r_{\text{out}} = \sum_i 1/\sigma_{v_i}^2 \quad ,$$

$$r_{\text{in}} = \max\left(0, 1/\sigma_{v_1}^2 - \sum_{i\neq 1} 1/\sigma_{v_i}^2\right)$$

is achievable. If the point the robot is trying to reach is inside the annulus, we use gradient descent algorithms to find an optimum solution that minimizes (5) by setting it to zero [16]. If the point is outside the annulus, we minimize the distance to the point the robot arm is trying to reach by lining up all the vectors along the abscissa such that the tip of the arm touches the outer or inner radius of the annulus. This configuration does not zero out (5) but gives the minimum achievable error. Figure 6 illustrates these two cases.

The gradient descent technique might give different solutions for different starting arm configurations. However, under the assumptions made earlier, all such solutions yield the same MSE on average. This leaves room for further relaxations of these assumptions. Some of these generalizations will be discussed in Sec. 6.

In Fig. 6(b), note that all the vectors point in the same direction. The best placement for this scenario is putting all cameras orthogonal to the object prior's major axis (such

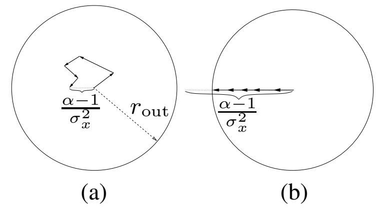


Fig. 6. Inverse kinematics can solve for the best θ_i: The case when the point to reach is (a) inside the annulus, and (b) outside of the annulus. Note that r_{in} is 0 for this example.

that twice the angles are 180°). This seems counterintuitive since we expect an orthogonal placement to be better for triangulation. However in this case, the prior uncertainty along the minor axis is small enough ($(\alpha-1)/\sigma_x^2 > \sum_i 1/\sigma_{v_i}$) that the optimal solution is to place all cameras to minimize the uncertainty along the major axis.

An example placement for $N = 4$, $\alpha = 5$, $\sigma_x = 4$ and $\sigma_{v_i}^2 = 5, 10, 15$, and 20 is given in Fig. 7(a). The room, object prior and resulting camera placements are shown. Note that the three higher noise cameras are placed close to each other, while the first camera is placed separately. The interpretation here is that the similar views from these bad cameras are averaged by the linear estimator to provide one good measurement. This is verified by the example in Fig. 7(b). Here, an optimal placement for two high quality cameras ($\sigma_{v_i}^2 = 5$ for both) is shown. The first camera is placed roughly at the same position as before, while the second camera is placed in the middle of the three bad camera positions. Note that the cameras are placed in $[0, \pi)$, as the corresponding vectors have angles in $[0, 2\pi)$ and they are twice the angles of the cameras. However, one can flip any camera to the opposite side of the room without changing its measurement.

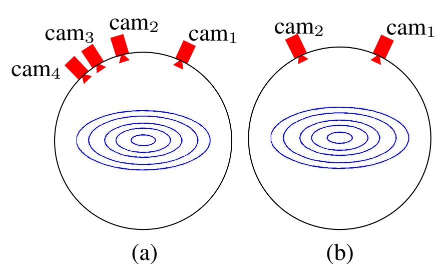


Fig. 7. Two optimal placements for the given object prior. (a) One good camera and 3 worse ones. (b) Two good cameras.

Note that for the general case, clustering the cameras into multiple groups can still achieve global optimality while solving the placement problem for each cluster as long as the clusters zero out their share of the offset. Suppose N cameras are clustered into c

groups. Then one algorithm might ask each cluster's "arm" to reach $-\frac{\alpha-1}{c\sigma_x^2}$. If this can be achieved by all the clusters, the solution is globally optimal.

5 Selection

For the selection problem defined in Sect. 3, the camera locations are fixed and we wish to select the best set S of size k that minimizes (4). In this problem, the object prior is not necessarily assumed to be at the center of the room. Also, the room does not have to be circular, and the cameras neither have to be placed at the periphery nor be oriented towards the center. This problem is difficult to solve because:

- All the terms in (4) change with different selections since the summation involving the σ_{v_i}s is a function of the selected set S.
- A naive search for the global optimum requires a combinatorial search among all possible sets S, which is $O(N^k)$. This is too costly.

To overcome these difficulties, we drop the numerator of (4) and focus only on optimizing the denominator. This is reasonable because it is equivalent to maximizing the mutual information between the measurements and the object location assuming Gaussian distributions, which is another good metric for camera selection ([11, 12]). Simulations also show that this modification does not introduce much performance degradation. We added the weights w_i inside the sums of (4), instead of running them over the subset S. The weights can be either 0 or 1. That is, they have to satisfy $w_i^2 - w_i = 0$. With the above, the problem can be formulated as follows:

$$\text{Maximize} \quad \left[\left(\tfrac{\alpha+1}{\sigma_x^2} + \sum_{i=1}^{N} \frac{w_i}{\sigma_{v_i}^2} \right)^2 - \left(\tfrac{\alpha-1}{\sigma_x^2} + \sum_{i=1}^{N} \frac{w_i \cos 2\theta_i}{\sigma_{v_i}^2} \right)^2 - \left(\sum_{i=1}^{N} \frac{w_i \sin 2\theta_i}{\sigma_{v_i}^2} \right)^2 \right]$$

$$\text{Subject to} \quad \sum_{i=1}^{N} w_i = k, \qquad w_i^2 - w_i = 0, \forall i \ .$$

This problem is not convex – neither the objective function nor the feasible set is necessarily convex. We use the following heuristic to find a good solution. We form the dual of the problem and use semi-definite programming (SDP) to find the dual optimal variables [17]. We then plug the dual optimal variables in the Lagrangian and solve for the w_is. We select the k cameras with the highest weights.

In practice, the selection technique we described might be performed distributedly as follows. The user asks for the location of an object with a desired accuracy. The query is passed to a cluster head near the object prior. The cluster head knows the locations of the other cameras near him. Using (3) with the lowest noise cameras, it computes a lower bound of the number of required cameras. It can also compute an upper bound assuming uniformly placed cameras and that the final selection will do better than the uniform selection. Using this lower and upper bounds, the cluster head decides on k. It then can compute the optimal selection of k cameras. If the predicted MSE from (4) for this selection is above the desired accuracy, k is incremented and the selection algorithm is repeated. Computing the optimal selection is feasible on a sensor node because it is computationally inexpensive. Finally, the selected cameras are queried for a measurement and the result is sent back to the cluster head to compute the measured localization. The relatively cheap processing is only done on the cluster head, and only the minimum number of camera nodes are queried. This limits the amount of networking and processing required of each node and is expected to greatly extend the overall network lifetime.

5.1 Simulation Results

We performed Monte-Carlo simulations to compare the performance of the above SDP approach to the optimal using brute-force enumeration as well as to the following other heuristics:

- *Uniform:* Pick uniformly placed cameras.
- *Closest:* Pick the closest cameras to the object mean.
- *Greedy:* Pick one camera at a time, using the *expected* posterior after each selected camera's measurement (w/o actually making a measurement) as the prior for the next camera.

In Fig. 8(a) we show a typical simulation run for $k = 3$ to 11 cameras out of 30 uniformly placed cameras on a circle of radius 100 units. The camera noise parameters used were

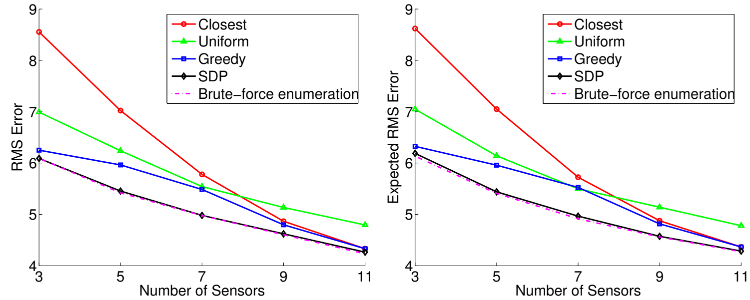
$$\sigma_{v_i}^2 = (0.1 \times d_i)^2 + 4 \ ,$$

where d_i is the distance from the ith camera to the object mean. For this run we chose $\alpha = 5$ and $\sigma_x^2 = 80$. As seen in the figure, the SDP approach achieves very close to optimal and clearly outperforms the other heuristic approaches. We can also predict the expected RMS of the localization using (2), for the selected cameras. Figure 8(b) shows the predicted RMS values, which are close to the RMS errors in Fig. 8(a).

5.2 Experimental Results

We tested our selection algorithm in an experimental setup consisting of 12 web cameras placed around a $22' \times 19'$ room. The horizontal FOV of the cameras used is $49\,^\circ$, and they all look toward the center of the room. The relative positions of the cameras in the room can be seen in Fig. 9(a). The cameras are hooked up to a PC via an IEEE 1394 (FireWire) interface and can provide 8-bit 3-channel (RGB) raw video at 15 Frames/s. The PC connected to a camera models a sensor node with processing and communication capabilities. Each PC is connected to 2 cameras, but the data from each camera is processed independently. The data is then sent to a central PC, where further processing is performed.

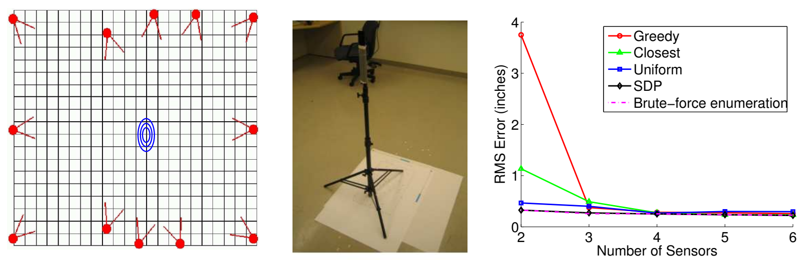
A process was run for each camera to perform background subtraction and generate the scan-lines (as described in Sect. 3). Only the selected cameras need to take a measurement and send the scan-line data over the network to an aggregating node where



(a) (b)

Fig. 8. (a) Localization performance for different selection heuristics. (b) Expected RMS localization error.

the localization is performed. The actual object that is localized is a point light source (see Fig. 9(b)).



(a) (b) (c)

Fig. 9. (a) The object prior and positions of the cameras in the real setup. Cones show FOV of cameras and grid spacing is 1'. (b) Object to be localized. (c) Experimental Results.

The object was randomly placed 100 times according to the prior:

$$\Sigma_x = (6'')^2 \times \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{4} \end{bmatrix} ,$$

and localized using the selection algorithm. The noise parameters for the cameras were measured separately. The selection algorithm was applied for 2 to 6 cameras using the object prior and noise statistics. Figure 9(c) shows the localization error of the selection heuristics. For $k = 2$ and 3, the SDP and brute-force enumeration heuristics perform the best. While uniform selection scheme is also much better compared to greedy and closest schemes, it performs about 40% worse than SDP or brute-force. For this specific localization problem, around 4 cameras is enough to localize the object to the accuracy of the ground truth, so for $k \geq 4$, the localization error of all the heuristics levels off to a similar value. From the figure we see that a good selection scheme can allow us to task a very small number of cameras (2 or 3 out of 12) and localize with an accuracy that is very close to optimum. On the other hand, if we have the luxury of tasking a large number of the cameras (e.g., 6 out of 12), then a simple scheme like uniform selection works just as well.

6 Discussion

In this section, we discuss how some of the non-idealities can be handled in our framework.

Static Occlusions: As we are localizing a single object, there is no occlusion from other moving objects. But there might be occlusions due to static objects such as partitions, tables, etc. For the case of selection, handling these is simple. If a camera cannot see a considerable portion of the object prior (if the prior probability of the object being in the area that a camera cannot see is bigger than a user defined threshold), we simply discard that camera from the feasible set of cameras and limit the search to the remaining set. For the placement, the following approach could be used: As mentioned in Sect. 4, flipping the camera to the other side of the room does not affect the result. So, the type of occlusions that do not have any occluding object at the other side of the room can easily be handled. If the solution obtained in Sect. 4 places the camera behind such an occluding object, one can still achieve the same result by flipping the cameras to the other side of the room. (See Fig. 10(a)).

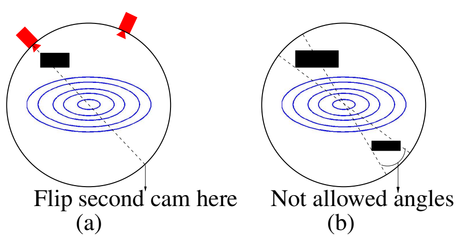


Fig. 10. (a) Static occlusions can be avoided if there is no other occluding object on the other side of the room. (b)We cannot place the cameras at some angles due to occlusion.

If it is the case that for some region of angles there are occluding objects at both sides of the room, then we cannot place the cameras at those angles (Fig. 10(b)). This places regions of angles that are not allowed (and regions that are allowed) for our inverse kinematics solution of Sect. 4. For the case that the number of allowed regions is one, a solution is given in [18]. Note that the situation illustrated in Fig. 10(b) is an example of one region of allowed angles (not two), as two flip sides of the room are equivalent. However, for the case of more than one such allowed regions, there is no general inverse kinematics solution. For this case, we can restrict each joint angle to be in a specific region and try all possible combinations until a feasible solution is found. Although the complexity of this search is exponential in the number of allowed regions, in practice we do not expect the number of occluding objects to be too high to make the computation infeasible.

Limited FOV: We can also handle limited FOV of cameras. For placement, one can place the cameras such that the object prior mean is mostly in the FOV of the camera. Of course there is still a possibility that the actual object position is too far away from the prior mean and it is out of the FOV of some cameras but this is very unlikely. For selection, again one can discard the cameras that cannot see a considerable portion of the object prior beforehand and restrict the selection only to the remaining cameras.

General Case for Placement: In Sect. 4 we assumed the object prior is centered in a circular room. Any general room shape and non-centered prior can be also handled as follows. We discretize all possible locations that a camera can be placed and assume there exists a camera at all of the locations. Then using the SDP heuristic described above, we select best k cameras. The actual placement is then done at the locations of these k selected cameras. Note that the selection heuristic we used did not require a circular room or centered object prior. This placement method can also handle the non-idealities such as limited FOV of the cameras and static occlusions, using the extensions described above.

7 Conclusion

With the goal of accurately localizing an object, we have shown how to efficiently compute the best sensor placement and selection in an idealized setting. Our results are analytical, and yield algorithms that are more computationally efficient than the numerical utility maximization techniques for sensor placement/selection [10–12]. For the placement problem, the globally optimal solution is found. For selection, an efficient approach using SDP is proposed. We demonstrated using simulation and experimentally that our selection algorithm performs as well as the exhaustive search and outperforms other heuristics.

Our algorithm for selection are performed assuming a static object. It can be easily extended to a moving object using a Kalman filter approach. Initially, the prior can be assumed circular. Using the measured data, the posterior of the object location can be computed and used as a prior for the next iteration.

In order to make the analysis tractable, we made several simplifying assumptions that resulted in some performance degradation in experiments. However, as the optimization criterion tries to distribute the viewing directions of the cameras, the results were still promising. When the noisy weak perspective camera assumption is valid (when the object is far enough from the cameras), our method is expected to work even better, as the results of our simulations showed.

Acknowledgments

The work in this paper was supported by Stanford SNRC Consortium, Stanford Media-X Consortium, Max Planck Center for Visual Computing and under NSF NeTS NOSS grant 0535111. We wish to thank Prof. John T. Gill III for his help in setting up the experimental lab, and to Prof. Jack Wenstrand, Prof. Balaji Prabhakar, Helmy Eltouhkhy, Sam Kavusi and James Mammen for their comments.

References

1. Pottie, G.J., Kaiser, W.J., Clare, L., Marcy, H.: Wireless integrated network sensors. Communications of the ACM **43**(5) (2000) 51–58
2. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: A survey. Computer Networks **38** (2002) 393–422
3. Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D., Anderson, J.: Wireless sensor networks for habitat monitoring. In: Proceedings of First International Workshop on Sensor Networks and Applications. (2002)
4. Yazawa, Y., Oonishi, T., Watanabe, K., Nemoto, R., Kamahori, M., Hasebe, T., Akamatsu, Y.: A wireless biosensing chip for DNA detection. In: Proceedings of ISSCC'05. (2005)
5. Yang, D.B.R., Shin, J.W., Ercan, A.O., Guibas, L.J.: Sensor tasking for occupancy reasoning in a network of cameras. In: Proceedings of BASENETS'04. (2004)
6. Chen, X., Davis, J.: Camera placement considering occlusion for robust motion capture. Stanford University Computer Science Technical Report, CS-TR-2000-07 (2000)
7. Olague, G., Mohr, R.: Optimal camera placement for accurate reconstruction. Pattern Recognition **35**(4) (2002) 927–944
8. Wu, J., Sharma, R., Huang, T.: Analysis of uncertainty bounds due to quantization for three-dimensional position estimation using multiple cameras. Optical Engineering **37**(1) (1998) 280–292
9. Zhang, H.: Two-dimensional optimal sensor placement. IEEE Transactions on Systems, Man, and Cybernetics **25**(5) (1995)
10. Chu, M., Haussecker, H., Zhao, F.: Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. The International Journal of High Performance Computing Applications **16**(3) (2002) 293–313
11. Ertin, E., Fisher III, J.W., Potter, L.C.: Maximum mutual information principle for dynamic sensor query problems. In: Proceedings of IPSN '03. (2003)
12. Wang, H., Yao, K., Pottie, G., Estrin, D.: Entropy-based sensor selection heuristic for localization. In: Proceedings of IPSN '04. (2004)
13. Vazquez, P.P., Feixas, M., Sbert, M., Heidrich, W.: Viewpoint selection using viewpoint entropy. In: Proceedings of the Vision Modeling and Visualization'01. (2001)
14. Wong, L., Dumont, C., Abidi, M.: Next best view system in a 3d object modeling task. In: Proceedings of Computational Intelligence in Robotics and Automation. (1999)
15. Roberts, D., Marshall, A.: Viewpoint selection for complete surface coverage of three dimensional objects. In: Proceedings of the British Machine Vision Conference. (1998)
16. Welman, C.: Inverse kinematics and geometric constraints for articulated figuremanipulation. Master's Thesis, Simon Fraser University (1993)
17. Poljak, S., Rendl, F., Wolkowicz, H.: A recipe for semidefinite relaxation for (0,1)-quadratic programming. Journal of Global Optimization **7** (1995) 51–73
18. Goldenberg, A.A., Benhabib, B., Fenton, R.G.: A complete generalized solution to the inverse kinematics of robots. IEEE Journal of Robotics and Automation **RA-1**(1) (1985) 14–20