Object Tracking in the Presence of Occlusions Using Multiple Cameras: A Sensor Network Approach

ALI O. ERCAN, Özyeğin University
ABBAS EL GAMAL and LEONIDAS J. GUIBAS, Stanford University

This article describes a sensor network approach to tracking a single object in the presence of static and moving occluders using a network of cameras. To conserve communication bandwidth and energy, we combine a task-driven approach with camera subset selection. In the task-driven approach, each camera first performs simple local processing to detect the horizontal position of the object in the image. This information is then sent to a cluster head to track the object. We assume the locations of the static occluders to be known, but only prior statistics on the positions of the moving occluders are available. A noisy perspective camera measurement model is introduced, where occlusions are captured through occlusion indicator functions. An auxiliary particle filter that incorporates the occluder information is used to track the object. The camera subset selection algorithm uses the minimum mean square error of the best linear estimate of the object position as a metric, and tracking is performed using only the selected subset of cameras.

Using simulations and preselected subsets of cameras, we investigate (i) the dependency of the tracker performance on the accuracy of the moving occluder priors, (ii) the trade-off between the number of cameras and the occluder prior accuracy required to achieve a prescribed tracker performance, and (iii) the importance of having occluder priors to the tracker performance as the number of occluders increases. We find that computing moving occluder priors may not be worthwhile, unless it can be obtained cheaply and to high accuracy. We also investigate the effect of dynamically selecting the subset of camera nodes used in tracking on the tracking performance. We show through simulations that a greedy selection algorithm performs close to the brute-force method and outperforms other heuristics, and the performance achieved by greedily selecting a small fraction of the cameras is close to that of using all the cameras.

Categories and Subject Descriptors: G.3 [**Mathematics of Computing**]: Probability and Statistics—*Probabilistic algorithms*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Auxiliary particle filter, camera sensor network, collaborative signal processing, noisy perspective camera model, occlusion, selection, sensor fusion, sensor tasking, tracking.

ACM Reference Format:
Ercan, A. O., El Gamal, A., Guibas, L. J. 2013. Object tracking in the presence of occlusions using multiple cameras: A sensor network approach. ACM Trans. Sensor Netw. 9, 2, Article 16 (March 2013), 36 pages.
DOI: http://dx.doi.org/10.1145/2422966.2422973

This work is partially supported by DARPA Microsystems Technology Office award no. N66001-02-1-8940, NSF grants CNS-0435111, CNS-0626151, ARO grant W911NF-06-1-0275, and DoD Multidisciplinary University Research Initiative (MURI) program administered by ONR under grant N00014-00-1-0637.
Preliminary versions of this work [Ercan et al. 2006, 2007] have been presented.

Authors' addresses: A. O. Ercan, Department of Electrical and Electronics Engineering, Özyeğin University, Istanbul, Turkey; email: ali.ercan@ozyegin.edu.tr; A. El Gamal, Electrical Engineering Department, Stanford University, Stanford, CA; email: abbas@ee.stanford.edu; L. J. Guibas, Computer Science Department, Stanford University, Stanford, CA; email: guibas@cs.stanford.edu.

DOI: http://dx.doi.org/10.1145/2422966.2422973

1. INTRODUCTION

There is a growing need to develop lowcost wireless networks of cameras with automated detection capabilities [Bhanu et al. 2011]. The main challenge in building such networks is the high data rate of video cameras. On the one hand, sending all the data (even after performing standard compression) is very costly in transmission energy; on the other hand, performing sophisticated vision processing at each node to substantially reduce transmission rate requires high processing energy.

To address these challenges, a *task-driven approach*, in which simple local processing is performed at each node to extract the essential information needed for the network to collaboratively perform the task, has been proposed and demonstrated [Zhao and Guibas 2004]. Only this essential information is communicated over the network, reducing the energy expended for communication without the need for complicated local processing. Further reduction in communication bandwidth and energy can be achieved by dynamically selecting the subset of camera nodes used [Zhao and Guibas 2004; Wang et al. 2004; Niu et al. 2011]. Thus, only a small subset of the nodes actively sense, process, and send data, while the rest are in sleep mode. In this article, we combine a task-driven approach with subset selection for tracking of a single object, for example, a suspect, in a structured environment, like an airport or a mall, in the presence of static and moving occluders via a network of cameras. Preliminary versions of this work have been presented in [Ercan et al. 2007, 2006].

Most previous work on tracking with multiple cameras has focused on tracking all the objects and does not deal directly with static occluders, which are often present in structured environments (see brief survey in Section 2). Tracking all the objects clearly provides a solution to our problem but may be infeasible to implement in a wireless camera network due to its high computational cost. Instead, our approach is to track only the target object, treating all other objects as occluders. We assume complete knowledge of the static occluder (e.g., partitions, large pieces of furniture) locations and some prior statistics on the positions of the moving occluders (e.g., people) which are updated in time. Each camera node performs local processing whereby each image is reduced to a single number indicating the horizontal position of the object in the image. If the camera sees the object, it provides a measurement of its position to the cluster head; otherwise, it reports that it cannot see the object. A noisy perspective camera measurement model is presented where occlusions are captured through occlusion indicator functions. Given the camera measurements and the occluder position priors, an auxiliary particle filter [Ristic et al. 2004] is used at the cluster head to track the object. The occluder information is incorporated into the measurement likelihood, which is used in the weighting of the particles.

Node subset selection is well suited for use in a camera network because measurements from close-by cameras can be highly correlated. Moreover, measurements from cameras that are far away from the target provide little useful information. Hence, by judiciously selecting the most appropriate subset of cameras, significant saving in energy can be achieved with little performance degradation relative to using all the cameras. Dynamic subset selection also helps avoid the occlusions in the scene and makes scaling of the network to a large number of nodes possible. The selection problem involves minimizing a utility metric over camera subsets of a predetermined size [Zhao and Guibas 2004]. We use the minimum mean square error (MSE) of the best linear estimator of the object location that incorporates the occlusions as the metric for selection. We describe the computation of the MSE metric using both the perspective camera model that we use in the tracking algorithm and a simpler weak perspective model which makes the computation of the MSE metric cheaper.

Using simulations with preselected subsets of cameras, we investigate the trade-off between the tracker performance, the moving occluder prior information, the number of cameras used, and the number of occluders present. Even if one wishes to track only one object, treating other moving objects as occluders, a certain amount of information about the positions of the occluders may be needed to achieve high tracking accuracy. Since obtaining more accurate occluder priors would require expending more processing and/or communication energy, it is important to understand the trade-off between the accuracy of the occluder information and that of tracking. Do we need any prior occluder information? If so, how much accuracy is sufficient?

We also investigate using simulations the effect of dynamically selecting the subsets of camera nodes used in tracking on the tracking performance. Every few time steps, the camera subset that minimizes the MSE is first selected and tracking is performed using this subset. The optimization needed to find the best camera subset of a certain size is in general combinatorial, and the complexity of brute-force search grows exponentially in the chosen subset size. This can be too costly in a wireless camera network setting. We show that a greedy selection algorithm performs close to the brute-force search and outperforms other heuristics, such as using a preselected or randomly selected subset of cameras. We also show that the performance of the greedy and brute-force heuristics are similar for both the perspective and weak perspective camera models and that the performance achieved by greedily selecting a small fraction of the cameras is close to that of using all the cameras.

The rest of the article is organized as follows. A brief survey of previous work on tracking using multiple cameras and node subset selection is provided in the next section. In Section 3, we describe our setup and introduce the camera measurement model used in tracking. The tracking algorithm is described in Section 4. Section 5 describes the computation of the MSE metric with the two camera models and the proposed selection algorithm. Simulations and experimental results are presented in Section 6 and 7, respectively.

2. PREVIOUS WORK

2.1. Tracking

Tracking has been a popular topic in sensor network research (e.g., [Li et al. 2002; Kim et al. 2005; Taylor et al. 2006; Shrivastava et al. 2006]). Most of this work, however, assumes low data rate range sensors, such as binary or acoustic sensors. By comparison, our work assumes cameras which are bearing sensors and have high data rate. Most of the previous work related to ours is by Pahalawatta et al. [2003], Funiak et al. [2006], del Blanco et al. [2008], and Sankaranarayanan et al. [2008, 2011]. Pahawalatta et al. [2003] use a camera network to track and classify multiple objects on the ground plane. This is done by detecting feature points on the objects and using a Kalman filter for tracking. By comparison, we use a particle filter, which is more suitable for nonlinear camera measurements, and track only a single object, treating others as occluders. Funiak et al. [2006] use a Gaussian model obtained by reparametrizing the camera coordinates together with a Kalman Filter. This method is fully distributed and requires less computational power than a particle filter. However, because the main goal of the system is camera calibration and not tracking, occlusions are not considered. In del Blanco et al. [2008], volumes occupied by moving objects in 3D are tracked using a particle filter. Although the setup and approach are similar to our work, it is assumed that measurements from different cameras are independent, which is not the case in general. An important contribution of our article is taking inter-camera measurement dependence into consideration while computing the likelihood. Another important difference between our work and that of del Blanco et al. is in the way occluders are considered. del Blanco et al. assume a fixed static occluder probability and do not consider moving occluders. As will be seen, we treat occluders in a more general way. Sankaranarayanan et al. [2008; 2011] focus on object detection, tracking, and recognition in visual sensor networks. They exploit planar world assumption and homography between cameras to come up with a computationally efficient tracker. However, this work does not consider occlusions, and our work does not rely on homography.

Tracking has also been a popular topic in computer vision [Khan et al. 2001; Yilmaz et al. 2004; Zajdel et al. 2004]. Most of the work, however, has focused on tracking objects in a single camera video sequence [Yilmaz et al. 2004]. Tracking using multiple camera video streams has also been considered [Khan et al. 2001; Zajdel et al. 2004]. Individual tracking is performed for each video stream and the objects appearing in the different streams are associated. More recently, there has been work on tracking multiple objects in world coordinates using multiple cameras [Utsumi et al. 1998; Otsuka and Mukawa 2004; Dockstander and Tekalp 2001]. Utsumi et al. [1998] extract feature points on the objects and use a Kalman filter to track the objects. They perform camera selection to avoid occlusions. By comparison, in our work, occlusion information is treated as part of the tracker and the selection metric. Otsuka and Mukawa [2004] describe a double loop filter to track multiple objects, where objects can occlude each other. One of the loops is a particle filter (PF) that updates the states of the objects in time using the object dynamics, the likelihood of the measurements, and the occlusion hypotheses. The other loop is responsible for generating these hypotheses and testing them using the object states generated by the first loop, the measurements, and a number of geometric constraints. Although this method also performs a single object tracking in the presence of moving occluders, the hypothesis generation and testing is computationally prohibitive for a sensor network implementation. The work also does not consider static occlusions that could be present in structured environments. Dockstader and Tekalp [2001] describe a method for tracking multiple people using multiple cameras. Feature points are extracted from images locally and corrected using the 3D estimates of the feature point positions that are fed back from the central processor to the local processor. These corrected features are sent to the central processor where a Bayesian network is employed to deduce a first estimate of the 3D positions of these features. A Kalman filter follows the Bayesian network to maintain temporal continuity. This approach requires that each object is seen by some cameras at all times. This is not required in our approach. Also, performing motion vector computation at each node is computationally costly in a wireless sensor network.

We would like to emphasize that our work is focused on tracking a single object in the presence of static and moving occluders in a wireless sensor network setting. When there are no occluders, one could adopt a less computationally intensive approach, similar to that of Funiak et al. [2006] or Sankaranarayanan et al. [2008, 2011]. When all the objects need to be tracked simultaneously, the previously mentioned methods [Otsuka and Mukawa 2004; Dockstander and Tekalp 2001] or a filter with joint-state for all the objects [Doucet et al. 2002; Vihola 2007] can be used.

2.2. Selection

Selection has been studied in wireless sensor networks with the goal of decreasing energy cost and increasing scalability. Chu et al. [2002] develop a technique referred to as “information driven sensor querying” 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. Ertin et al. [2003] use the mutual information metric to select sensors. This is shown to be equivalent to minimizing the expected posterior uncertainty but with significantly less computation. Wang et al. [2004] expands on Ertin et al. [2003] 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. In comparison to these works, we use the minimum mean square object localization error as the selection metric. Also, we consider the occlusion phenomenon, which is unique to camera sensors compared to other sensing modalities.

Sensor selection has also been studied for camera sensors. Wong et al. [1999] and Vazquez et al. [2001] define a metric for the next best view based on most faces seen (given a 3D 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. Yang et al. [2004] and Isler and Bajcsy [2005] deal with sensing modalities where the measurements can be interpreted as polygonal subsets of the ground plane, such as the visual hull, and use geometric quantities such as the area of these subsets as the selection metric. Yang et al. [2004] present a greedy search to minimize the selection metric. Isler and Bajcsy [2005] prove that, for their setting, an exhaustive search for at most six sensors yields a performance within a factor two of the optimal selection. In Tessens et al. [2008], the visual hull metric is augmented with measures such as the ability to detect faces and the object speed and visibility. These works use numerical techniques or heuristics to compute the viewpoint scores. We investigate a simpler problem and use the mean square error (MSE) of the object location as our metric for selection.

In principle, Niu et al. [2011] is the closest work to our article. The authors first describe in Zuo et al. [2011] the computation of the conditional posterior Cramér-Rao Lower Bound (CRLB) in a nonlinear sequential Bayesian estimation framework. The CRLB is a lower bound on the variance of any unbiased estimator. The Conditional Posterior CRLB (CPCRLB) is the lower bound on the variance of the recursive estimator that updates the estimate with a new measurement, given the past measurements. Niu et al. use the CPCRLB as a metric for sensor selection in a particle filter tracker. Similar to our work, the selection algorithm uses the output of the particle filter tracker (i.e., the particle distributions) to compute the selection metric; however; our work extends the sensing modality to camera sensors. The camera model we use is similar to the bearing model in Niu et al. [2011]; however, we model occlusions and take inter-camera measurement dependence into consideration. We provide extensive simulations to explore the trade-offs previously mentioned. Also, we use the minimum mean square error (MSE) of the best linear estimator of the object location as the camera selection metric instead of the CPCRLB.

3. SETUP, MODEL, AND ASSUMPTIONS

We consider a cluster of camera nodes, each capable of locally processing their captured images and communicating the locally processed data wirelessly to a cluster head, where tracking and selection are performed. The cluster head can be selected among the regular camera nodes or it can be a more powerful pre-assigned node. Cluster formation and cluster head selection [Heinzelman et al. 2002] are not the focus of this article and are assumed to be performed a priori.

Our problem setup is illustrated in Figure 1, in which N cameras are aimed roughly horizontally around a room. Although an overhead camera (i.e., mounted on the ceiling and aimed directly downward) would have a less occluded view than a horizontally placed one, it generally has a more limited view of the scene and may be impractical to deploy. Additionally, targets may be easier to identify in a horizontal view. The camera network's task is to track an object on the ground plane in the presence of static occluders and other moving objects.

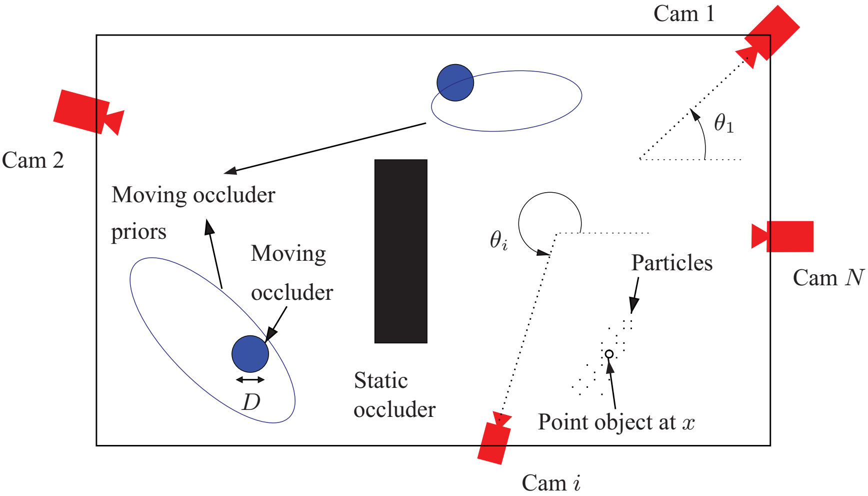

Fig. 1. Illustration of the problem setup.

The cameras are assumed to be fixed and their calibration information is assumed to be known to some accuracy to the cluster head. We assume that the roll angle (rotation around the optical axis) of the cameras is zero. As previously mentioned, we also assume that the cameras are aimed roughly horizontally; therefore, the pitch angle (vertical tilt) is also close to zero. For example, in the experimental setup (see Section 7), the cameras are mounted on the walls slightly above an average human's height and are tilted slightly downward. If these assumptions are not valid, one can use a more complete camera model (in Section 3.1) assuming nonzero roll and pitch angles. We did not consider this extension for simplicity. Given these assumptions and since the object is tracked on the 2D ground plane, the calibration information for camera i consists of its coordinates on the ground plane and its yaw angle (i.e., rotation around the vertical axis. In Figure 1, θ_i is π plus the yaw angle of camera i.).

We assume the object to track to be a point object, since it may be detected by some specific point features [Shi and Tomasi 1994]. We assume there are M other moving objects, each modeled as a cylinder of diameter D. The position of each object is assumed to be the center of its cylinder. From now on, we shall refer to the object to track as the "object" and the other moving objects as "moving occluders." We further assume that the camera nodes can distinguish between the object and the occluders. This can be done again using feature detection (e.g., [Shi and Tomasi 1994]).

For each image frame captured, background subtraction is performed locally at each camera node to detect the moving objects. Then if the object to track is in the field of view of the camera and not occluded by a static or moving occluder, its horizontal position in the image plane is estimated and sent to the cluster head. Note that the amount of data transferred per image is very small compared to the image data size.

We acknowledge the fact that point object assumption is not completely realistic, since it cannot model self occlusions of the point features by the object. Also, recognizing the object and distinguishing it from the moving occluders is a very challenging task [Sankaranarayanan et al. 2008]. On the one hand, with some additions to our work, this task can be simplified. For example, the estimated location of the object from the previous time step can be fed back to the cameras to limit the region where feature detection is performed. However, recognizing the object or performing more

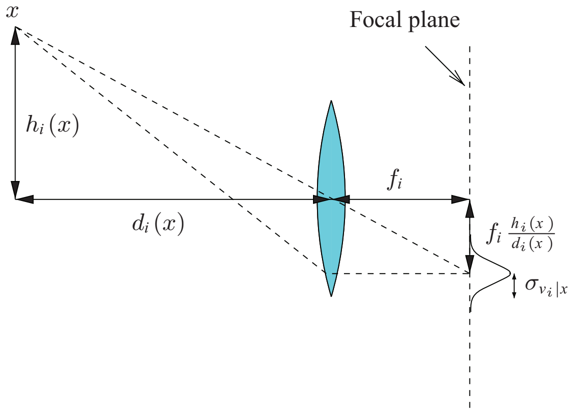

Fig. 2. Noisy perspective camera measurement model with unoccluded object at x.

complex local image processing is not the focus of this article. These topics have been the subject of extensive research [Yang et al. 2002; Zhao et al. 2003] and the references therein. In this article, we focus on a sensor network approach to the data fusion problem in camera networks.

We also acknowledge the fact that perfect object detection is not a realistic assumption and that mis-detections and false positives will occur. However, these can be handled within our probabilistic tracking framework, as explained in the last paragraph of Section 4.3.

The positions and the shapes of the static occluders in the room are assumed to be completely known in advance. This is not unreasonable since this information can be easily provided to the cluster head. On the other hand, only some prior statistics of the moving occluder positions are known at each time step. How to obtain these priors is not the focus of this article, however, in Section 4.4, we discuss some preliminary ideas on how these priors may be obtained. Note that the moving occluder priors are not required for the tracking algorithm; however, they increase the tracking accuracy. In Section 6.1, we describe how the tracker formulation is modified for the case of no moving occluder information and we explore the trade-off between the moving occluder prior accuracy and that of the tracker. The selection algorithm also can be implemented without the moving occluder priors, using the same modification.

We assume that the number of moving occluders in the room (i.e., M) is known. One can obtain this number from the number of moving occluder priors. If the priors are not available, one can utilize algorithms such as those described in Yang et al. [2003] in parallel to tracking to estimate this number.

We assume that the camera nodes are synchronized. The tightness of the synchronization depends on the speed of motion of the tracked object. For example, in the experimental setup in Section 7, synchronization was achieved by starting the cameras at the same time, and no further synchronization constraint was imposed. This means the synchronization was of the order time elapsed between two consecutive frames (1/7.5 sec^{-1}), which is reasonable for average human walking speeds.

3.1. Camera Measurement Model

If camera $i \in \{1, 2, \ldots, N\}$, "sees" the object, we assume a noisy perspective camera measurement model (see Figure 2), and the horizontal position of the object in the image plane is given by

$$z_i = f_i \frac{h_i(x)}{d_i(x)} + v_i, \tag{1}$$

where x is the location of the object, f_i is the focal length for camera i, and $h_i(x)$ and $d_i(x)$ are the distances defined in the figure. The camera measurement noise v_i is due to its readout noise and calibration inaccuracies. Assuming the readout noise and the camera position and orientation (yaw angle) calibration inaccuracies to be zero mean with variances σ^2_{read}, σ^2_{pos}, and σ^2_θ, respectively, it can be shown that given x, v_i has zero mean and conditional variance (see Appendix B for details):

$$\sigma^2_{v_i|x} = f_i^2 \left(1 + \frac{h_i^2(x)}{d_i^2(x)}\right)^2 \sigma^2_\theta + f_i^2 \left(\frac{h_i^2(x) + d_i^2(x)}{d_i^4(x)}\right) \sigma^2_{\text{pos}} + \sigma^2_{\text{read}}. \tag{2}$$

We assume that $v_1, v_2, \ldots, v_N$ are conditionally independent Gaussian random variables, given x. If the camera cannot see the object because of occlusions or limited field of view, it reports a "not-a-number" (NaN, using MATLAB syntax) to the cluster head.

We can combine the preceding two camera measurement models into a single model by introducing, for each camera i, the following *occlusion indicator function*:

$$\eta_i := \begin{cases} 1, & \text{if camera } i \text{ sees the object,} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Note that the η_i random variables are not in general mutually independent. Using these indicator functions, the camera measurement model including occlusions can be expressed as

$$z_i = \begin{cases} f_i \frac{h_i(x)}{d_i(x)} + v_i, & \text{if } \eta_i = 1, \\ \text{NaN}, & \text{if } \eta_i = 0. \end{cases} \tag{4}$$

4. THE TRACKER

Since the measurements from the cameras of Equation (4) are nonlinear in the object position, using a linear filter (such as a Kalman filter (KF)) for tracking would yield poor results. As discussed in Bar-Shalom et al. [2001], using an extended kalman filter (EKF) with measurements from bearing sensors, which are similar to cameras with the aforementioned local processing, is not very successful. Although the use of an unscented Kalman filter (UKF) is more promising, its performance degrades quickly when the static occluders and limited field of view constraints are considered. Because of the discreteness of the occlusions and fields of views and the fact that UKF uses only a few points from the prior of the object state, most of these points may get discarded. We also experimented with a maximum a-posteriori (MAP) estimator combined with a Kalman filter, which is similar to the approach in Taylor et al. [2006]. This approach, however, failed at the optimization stage of the MAP estimator, as the feasible set is highly disconnected due to the static occluders. Given these considerations, we decided to use a particle filter (PF) tracker [Ristic et al. 2004; Caron et al. 2007].

We denote by $u(t)$ the state of the object at time t, which includes its position $x(t)$ and other relevant information, such as its intended destination and its speed (see Section 4.1 for details). The positions of the moving occluders $j \in \{1, \ldots, M\}$, $x_j(t)$ are assumed to be Gaussian with mean $\mu_j(t)$ and covariance matrix $\Sigma_j(t)$. These priors are

ALGORITHM 1: ASIR – One Step of the Auxiliary Sampling Importance Resampling Filter

```
Input: Particle - weight tuples: $\{u_\ell(t-1), w_\ell(t-1)\}_{\ell=1}^{L}$; moving occluder priors: $\{\mu_j(t), \Sigma_j(t)\}_{j=1}^{M}$;
        measurements: $Z(t) = [z_1(t), \ldots, z_N(t)]^T$; struct room (camera fields of views, positions
        and orientations; room's shape and sizes; static occluder information).
Output: Particle - weight tuples: $\{u_\ell(t), w_\ell(t)\}_{\ell=1}^{L}$.
for ($\ell = 1, \ldots, L$) do
    $\kappa_\ell := \mathrm{E}(u(t)|u_\ell(t-1))$;
    $\tilde{w}_\ell(t) \propto w_\ell(t-1) f(Z(t)|\kappa_\ell)$; /*Section 4.2*/
end
$\{w_\ell(t)\}_{\ell=1}^{L} = \mathrm{normalize}(\{\tilde{w}_\ell(t)\}_{\ell=1}^{L})$;
$\{\breve{\kappa}_\ell, \breve{w}_\ell, \pi^\ell\}_{\ell=1}^{L} = \mathrm{resample}(\{\kappa_\ell, w_\ell(t)\}_{\ell=1}^{L})$;
for ($\ell = 1, \ldots, L$) do
    $u_\ell(t) \sim f(u(t)|u_{\pi^\ell}(t-1))$; /*Section 4.1*/
    $\tilde{w}_\ell(t) = \frac{f(Z(t)|u_\ell(t))}{f(Z(t)|\kappa_{\pi^\ell})}$;
end
$\{w_\ell(t)\}_{\ell=1}^{L} = \mathrm{normalize}(\{\tilde{w}_\ell(t)\}_{\ell=1}^{L})$;
```

available to the tracker. The state of the object and positions of moving occluders are assumed to be mutually independent. If the objects move in groups, one can still apply the following tracker formulation by defining a "super-object" for each group and assuming that the super-objects move independently. Also, in reality, people avoid colliding with each other, which violates the independence assumption when two objects come closer. We ignored this fact in the design of the tracker for simplicity. However, the objects do avoid collisions in the simulations (Section 6) and in the experiments (Section 7).

The tracker maintains the probability density function (pdf) of the object state $u(t)$ and updates it at each time step using the new measurements. Given the measurements from all cameras up to time $t-1$, $\{Z(t')\}_{t'=1}^{t-1}$, the particle filter approximates the pdf of $u(t-1)$ by a set of L weighted particles as follows:

$$f(u(t-1)|\{Z(t')\}_{t'=1}^{t-1}) \approx \sum_{\ell=1}^{L} w_\ell(t-1)\delta\left(u(t-1) - u_\ell(t-1)\right),$$

where $\delta(\cdot)$ is the Dirac delta function, $u_\ell(t)$ and $w_\ell(t)$ are the state and weight of particle ℓ at time t, respectively. At each time step, given these L weighted particles, the camera measurements $Z(t) = [z_1(t), \ldots, z_N(t)]^T$, the moving occluder priors $\{\mu_j(t), \Sigma_j(t)\}$, $j \in \{1, \ldots, M\}$, information about the static occluder positions, and the camera fields of view, the tracker incorporates the new information obtained from the measurements at time t to update the particles and their associated weights.

We use the Auxiliary Sampling Importance Resampling (ASIR) filter [Ristic et al. 2004]. The outline of one step of our implementation of this filter is given in Algorithm 1. In the algorithm, $\mathrm{E}(\cdot)$ is the expectation operator, and the procedure `normalize` scales the weights so that they sum to one. The procedure `resample` takes L particle-weight pairs and produces L equally weighted particles ($w_\ell = 1/L$), while preserving the original distribution [Ristic et al. 2004]. The third output of the procedure (π^ℓ) refers to the index of particle ℓ's parent before resampling, while the first two outputs are not used in the rest of the algorithm. The ASIR algorithm approximates the optimal importance density function $f(u(t)|u_\ell(t-1), Z(t))$, which is not feasible to compute in general [Ristic et al. 2004].

In the following, we explain the implementation of the importance density function $f(u(t)|u_\ell(t-1))$ and the likelihood $f(Z(t)|u_\ell(t))$.

4.1. Importance Density Function

The particles are advanced in time by drawing new samples $u_\ell(t)$ from the *importance density function*: $u_\ell(t) \sim f(u(t)|u_\ell(t-1)), \ell \in \{1, \dots, L\}$. This is similar to the time update step in a Kalman filter. After all L new particles are drawn, the distribution of the state is forwarded one time step. Therefore, the dynamics of the system should be reflected as accurately as possible in the importance density function. In a Kalman filter, a constant velocity assumption with a large variance on the velocity is assumed to account for direction changes. Although assuming that objects moving at constant velocity is not a realistic assumption, especially when an object changes its direction of motion drastically (e.g., when the object changes its intended destination), the linearity constraint of the Kalman filter forces this choice. In the PF implementation, we do not have to choose linear dynamics. We use the random waypoints model [Bettstetter et al. 2002], in which the objects choose a destination that they want to reach and try to move toward it with constant speed plus noise until they reach the destination. When they reach it, they choose a new destination.

We implemented a modified version of this model in which the state $u_\ell(t)$ of particle ℓ consists of its current position $x_\ell(t)$, destination $\tau_\ell(t)$, speed $s_\ell(t)$, and regime $r_\ell(t)$. Note that the time step here is 1, and thus $s_\ell(t)$ represents the distance traveled in a unit time. The regime can be one of the following.

(1) *Move Toward Destination (MTD)*. A particle in this regime tries to move toward its destination with constant speed plus noise,

$$x_\ell(t) = x_\ell(t-1) + s_\ell(t-1)\frac{\tau_\ell(t-1) - x_\ell(t-1)}{\|\tau_\ell(t-1) - x_\ell(t-1)\|_2} + \nu(t), \tag{5}$$

where $\nu(t)$ is zero mean Gaussian white noise with covariance $\Sigma_\nu = \sigma_\nu^2 I$, I denotes the identity matrix, and σ_ν is assumed to be known. The interpretation of Equation (5) is as follows: a unit vector toward the particle's destination is multiplied by its speed and added to the previous position, together with a zero mean Gaussian noise vector. The speed of the particle is also updated according to $s_\ell(t) = (1-\phi)s_\ell(t-1) + \phi\|x_\ell(t) - x_\ell(t-1)\|_2$. Updating the speed this way smooths out the variations due to added noise. We arbitrarily chose $\phi = 0.7$ for our implementation. The destination $\tau_\ell(t)$ is left unchanged.

(2) *Change Destination (CD)*. A particle in this regime first chooses a new destination uniformly randomly in the room and performs an MTD step.

(3) *Wait (W)*. A particle in this regime does nothing.

Drawing a new particle from the importance density function involves the following. First, each particle chooses a regime according to its current position and destination. If a particle does not reach its destination, it chooses the regime according to

$$r_\ell(t) = \begin{cases} \text{MTD}, & \text{with probability } \beta_1, \\ \text{CD}, & \text{with probability } \lambda_1, \\ \text{W}, & \text{with probability } (1 - \beta_1 - \lambda_1). \end{cases}$$

If a particle reaches its destination, the probabilities β_1 and λ_1 are replaced by β_2 and λ_2, respectively. The destination is assumed reached when the distance to it is less than the particle's speed. We arbitrarily chose $\beta_1 = 0.9, \lambda_1 = 0.05, \beta_2 = 0.05, \lambda_2 = 0.9$.

4.2. Likelihood

Updating the weights in the ASIR algorithm requires the computation of the likelihood of a given set of measurements obtained from the cameras, that is, $f(Z(t)|u_\ell(t))$. Here we implicitly assume that the measurement noise $v_i(t)$ in Equation (4) is white Gaussian noise given $x_\ell(t)$. Although the readout noise component in Equation (2) can be modeled as white Gaussian noise, the calibration inaccuracies are fixed over time. Therefore, $v_i(t)$ given $x_\ell(t)$ is not independent of previous noise samples. This can be dealt with by including the camera calibration errors in the state $u(t)$. For such an approach to be scalable, a fully distributed implementation whereby each camera estimates its own calibration errors is essential, since otherwise the tracker state would grow with the number of cameras. An alternative approach is to estimate the calibration errors of the cameras at an initialization phase. Both approaches result in having just the readout noise component, which is white. To simplify the analysis, however, we assume that $v_i(t)$ given $x_\ell(t)$ is white Gaussian noise.

For brevity, we shall drop the time index from now on. We can use the chain rule for probabilities to decompose the likelihood and obtain

$$f(Z|u_\ell) = f(Z, \eta|u_\ell) = p(\eta|u_\ell) f(Z|\eta, u_\ell), \tag{6}$$

where $\eta := [\eta_1, \ldots, \eta_N]^T$ are the occlusion indicator variables. In the first equality, we use the fact that η can be derived from a given Z, since $z_i = \mathrm{NaN}$ if and only if $\eta_i = 0$.

Given x_ℓ, which is part of u_ℓ, and η, $z_1, \ldots, z_N$ are independent Gaussian random variables and the second term in Equation (6) is

$$f(Z|\eta, u_\ell) = \prod_{i \in \{j:\, \eta_j = 1\}} \mathcal{N}\left\{z_i; f_i \frac{h_i(x_\ell)}{d_i(x_\ell)}, \sigma^2_{v_i|x_\ell}\right\},$$

where $\mathcal{N}\{r; \xi, \rho^2\}$ denotes a univariate Gaussian function of r with mean ξ and variance ρ^2, $\sigma^2_{v_i|x_\ell}$ is given in Equation (2) and $d_i(x)$ and $h_i(x)$ are defined in Figure 2. This term by itself does not exploit any information from the occlusions, while there might be valuable information there. To see this, consider the following example. Suppose that there are no dynamic occluders and just one static occluder (e.g., a pillar) in the room. Also assume there is only one camera and it reports NaN. This means the object must be behind the pillar. So even with one occlusion measurement, one can deduce quite a lot about the object's whereabouts. Such information is incorporated by the first term (i.e., $p(\eta|u_\ell)$) in the likelihood. However unlike the second term, $p(\eta|u_\ell)$ cannot simply be expressed as a product, as the occlusions are not independent given u_ℓ. This can be explained via the following simple example. Suppose two cameras are close to each other. Once we know that one of these cameras cannot see the object, it is more likely that the other one also cannot see it. Hence, the two ηs are dependent given u_ℓ. Luckily, we can approximate this term using recursion, which is described next.

First, we ignore the static occluders and the limited field of view constraints and only consider the effect of the moving occluders. The effects of static occluders and limited field of view will be incorporated in Section 4.3. Define the indicator functions $\eta_{i,j}$ for $i = 1, \ldots, N$ and $j = 1, \ldots, M$ such that $\eta_{i,j} = 1$ if occluder j does not occlude camera i, and 0, otherwise. Thus

$$\{\eta_i = 1\} = \bigcap_{j=1}^{M} \{\eta_{i,j} = 1\}.$$

Define $q^{\mathrm{mv}}_{i,j}(x)$ to be the probability that occluder j occludes camera i given u, where the superscript "mv" signifies that only moving occluders are taken into account.

$$\begin{aligned} q^{\mathrm{mv}}_{i,j}(x) &:= \mathrm{P}\left\{\eta_{i,j} = 0|u\right\} \\ &= \int_{\mathbb{R}^2} f(x_j|u)\mathrm{P}\left\{\eta_{i,j} = 0|u, x_j\right\}\, dx_j \overset{(a)}{=} \int_{\mathbb{R}^2} f(x_j)\mathrm{P}\left\{\eta_{i,j} = 0|x, x_j\right\}\, dx_j, \end{aligned}$$

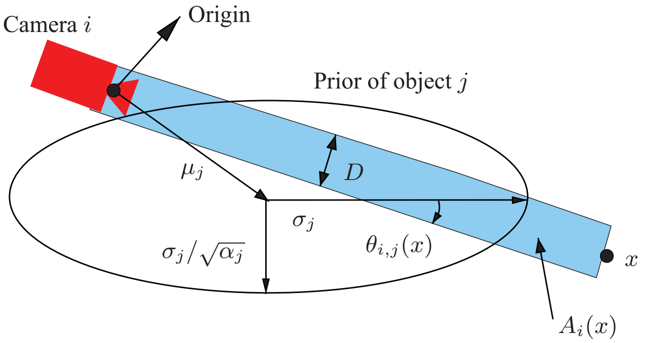

Fig. 3. Computing $q^{\mathrm{mv}}_{i,j}(x)$. Without loss of generality, the camera is assumed at the origin and everything is rotated such that the major axis of occluder j's prior is horizontal. Occluder j occludes point x at camera i if its center is inside the rectangle $A_i(x)$.

where x is the position part of the state vector u, $\mathbb{R}^2$ denotes the 2D real plane, and step (a) uses the fact that x_j is independent of u and $\eta_{i,j}$ is a deterministic function of x and x_j.

To compute $q^{\mathrm{mv}}_{i,j}(x)$, refer to Figure 3. Without loss of generality, we assume that camera i is placed at the origin and everything is rotated such that the major axis of occluder j's prior is horizontal. Occluder j occludes point x at camera i if its center is inside the rectangle $A_i(x)$. This means $\mathrm{P}\left\{\eta_{i,j} = 0 | x, x_j\right\} = 1$ if $x_j \in A_i(x)$, and it is zero everywhere else.

$$\begin{aligned} q^{\mathrm{mv}}_{i,j}(x) &= \int_{A_i(x)} \frac{1}{2\pi\sqrt{|\Sigma_j|}} e^{-\frac{1}{2}(x_j-\mu_j)^T\Sigma_j^{-1}(x_j-\mu_j)}\,dx_j \\ &\overset{(b)}{\approx} \frac{1}{4}\left[\mathrm{erf}\left(\frac{\sqrt{\alpha_j}}{\|g_1'\|}\left(\frac{D}{2}-\varphi\right)\right) + \mathrm{erf}\left(\frac{\sqrt{\alpha_j}}{\|g_1'\|}\left(\frac{D}{2}+\varphi\right)\right)\right] \\ &\quad \left[\mathrm{erf}\left(\frac{\|x\|\|g_1\|^2 - {\mu_j}^T o_1}{\|g_1'\|}\right) + \mathrm{erf}\left(\frac{{\mu_j}^T o_1}{\|g_1'\|}\right)\right], \end{aligned} \tag{7}$$

where $o_1^T = [\cos(\theta_{i,j}(x))\ \alpha_j \sin(\theta_{i,j}(x))]$, $g_1^T = [\cos(\theta_{i,j}(x))\ \sqrt{\alpha_j}\sin(\theta_{i,j}(x))]$, $\theta_{i,j}(x)$ is defined in the figure, $g_1' = \sqrt{2}\sigma_j g_1$, $\varphi = [-\sin(\theta_{i,j}(x))\ \cos(\theta_{i,j}(x))]\mu_j$, and σ_j^2 and σ_j^2/α_j $(\alpha_j \geq 1)$ are the eigenvalues of the covariance matrix Σ_j of the prior of occluder j. Step (b) follows the assumption that the moving occluder diameter D is small compared to the occluder standard deviations. See Appendix C for the derivation of this formula.

To compute $p(\eta|u)$, first define $p^{\mathrm{mv}}_Q(x)$ to be the probability of all ηs of the cameras in a subset Q, given u, to be equal to 1.

$$\begin{aligned} p^{\mathrm{mv}}_Q(x) &:= \mathrm{P}\left(\bigcap_{i\in Q}\{\eta_i = 1\}\middle| u\right) = \mathrm{P}\left(\bigcap_{i\in Q}\bigcap_{j=1}^{M}\{\eta_{i,j} = 1\}\middle| u\right) \\ &= \mathrm{P}\left(\bigcap_{j=1}^{M}\bigcap_{i\in Q}\{\eta_{i,j} = 1\}\middle| u\right) \overset{(c)}{=} \prod_{j=1}^{M}\mathrm{P}\left(\bigcap_{i\in Q}\{\eta_{i,j} = 1\}\middle| u\right) \end{aligned}$$

$$
\begin{aligned}
&= \prod_{j=1}^{M}\left(1-\mathrm{P}\left(\bigcup_{i\in Q}\{\eta_{i,j}=0\}\middle| u\right)\right) \overset{(d)}{\approx} \prod_{j=1}^{M}\left(1-\sum_{i\in Q}\mathrm{P}\left\{\eta_{i,j}=0|u\right\}\right)\\
&= \prod_{j=1}^{M}\left(1-\sum_{i\in Q} q_{i,j}^{\mathrm{mv}}(x)\right), \qquad (8)
\end{aligned}
$$

where (c) follows by the assumption that the occluder positions are independent, and (d) follows from the assumption of small D and the reasonable assumption that the overlap between $A_i(x)$, $i \in Q$, is negligible. Note that cameras that satisfy this condition can still have significantly overlapping fields of views, since the fields of views are expected to be significantly larger than $A_i(x)$. Therefore, this condition does not contradict the argument that occlusion indicator variables η_i are dependent given u_ℓ.

Now we can compute $p^{\mathrm{mv}}(\upeta|u)$ using Equation (8) and recursion as follows. Choose any n such that $\eta_n = 0$ and define $\upeta_{-n} := [\eta_1, \dots, \eta_{n-1}, \eta_{n+1}, \dots, \eta_N]^T$ and $\upeta_{\bar{n}} := [\eta_1, \dots, \eta_{n-1}, 1, \eta_{n+1}, \dots, \eta_N]^T$. Then,

$$p^{\mathrm{mv}}(\upeta|u) = p^{\mathrm{mv}}(\upeta_{-n}|u) - p^{\mathrm{mv}}(\upeta_{\bar{n}}|u). \qquad (9)$$

Both terms in the right-handside of Equation (9) are one step closer to $p_Q^{\mathrm{mv}}(u)$ (with different Q), because one less element is zero in both $\upeta_{-n}$ and $\upeta_{\bar{n}}$. This means that any $p^{\mathrm{mv}}(\upeta|u)$ can be reduced recursively to terms consisting of $p_Q^{\mathrm{mv}}(x)$, using Equation (9).

Let us illustrate this with the following example. Assume we have $N = 3$ cameras and $\upeta = \{1, 0, 0\}$. Then,

$$
\begin{aligned}
p^{\mathrm{mv}}(\upeta|u) &= \mathrm{P}\left(\{\eta_1=1\}\cap\{\eta_2=0\}\cap\{\eta_3=0\}|u\right)\\
&= \mathrm{P}\left(\{\eta_1=1\}\cap\{\eta_2=0\}|u\right) - \mathrm{P}\left(\{\eta_1=1\}\cap\{\eta_2=0\}\cap\{\eta_3=1\}|u\right)\\
&= \mathrm{P}\left(\{\eta_1=1\}|u\right) - \mathrm{P}\left(\{\eta_1=1\}\cap\{\eta_2=1\}|u\right) -\\
&\quad\ \mathrm{P}\left(\{\eta_1=1\}\cap\{\eta_3=1\}|u\right) + \mathrm{P}\left(\{\eta_1=1\}\cap\{\eta_2=1\}\cap\{\eta_3=1\}|u\right)\\
&= p_{\{1\}}^{\mathrm{mv}}(x) - p_{\{1,2\}}^{\mathrm{mv}}(x) - p_{\{1,3\}}^{\mathrm{mv}}(x) + p_{\{1,2,3\}}^{\mathrm{mv}}(x),
\end{aligned}
$$

where we used the preceding trick twice to obtain four terms of the form $p_Q^{\mathrm{mv}}(x)$. Note that the computational load of this recursion is exponential in the number of zeros in $\upeta$. As we illustrate in the next section, this is not a problem in practice.

4.3. Incorporating the Effects of the Static Occluders and Limited Camera Field of View

Incorporating the effects of the static occluders and limited camera field of view to the procedure previously described involves a geometric partitioning of the particles. Each partition is assigned a set of cameras. Only the ηs of the assigned cameras are considered for the particles in that partition. This is explained using the example in Figure 4. In this example, we have two cameras and a single static occluder. As denoted by the dashed line in the figure, we have two partitions. Let $\eta_1 = 0$ and $\eta_2 = \gamma_2 \in \{0, 1\}$. Let us consider a particle belonging to the upper partition, namely particle ℓ_1 at x_{ℓ_1}. If the object is at x_{ℓ_1}, the static occluder makes $\eta_1 = 0$, independent of where the moving occluders are. So, only Cam$_2$ is assigned to this partition, and the first term in the likelihood is given by $\mathrm{P}\left(\{\eta_1 = 0\} \cap \{\eta_2 = \gamma_2\}|u_{\ell_1}\right) = p^{\mathrm{mv}}(\eta_2|u_{\ell_1})$. Similarly, $\mathrm{P}\left(\{\eta_1 = 1\} \cap \{\eta_2 = \gamma_2\}|u_{\ell_1}\right) = 0$, because if the object is at x_{ℓ_1}, $\eta_1 = 0$, and $\mathrm{P}\left(\{\eta_1 = \gamma_1\}\ \cap\{\eta_2 = \gamma_2\}|u_{\ell_2}\right) = p^{\mathrm{mv}}(\eta_1, \eta_2|u_{\ell_2})$, because the static occluder and limited field of view do not occlude particle ℓ_2. If a particle u_ℓ is out of the fields of views of all cameras, $\mathrm{P}\left(\{\eta_1 = 0\} \cap \{\eta_2 = 0\} \cap\ \cdots \cap \{\eta_N = 0\}|u_\ell\right) = 1$, and the likelihood of any other combination is 0.

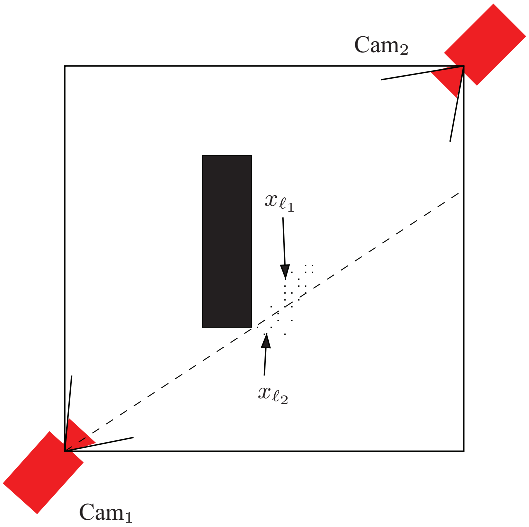

Fig. 4. Geometric partitioning to add the effects of static occluders and limited field of view into the likelihood. If $\eta_1 = 1$, the object cannot be at x_{ℓ_1}. If $\eta_1 = 0$, only Cam_2 needs to be considered for computing $p(\eta|u_{\ell_1})$. Both cameras need to be considered for computing $p(\eta|u_{\ell_2})$.

Note that the number of cameras assigned to a partition is not likely to be large in practice. This is because in practical installations, the cameras are spread out to monitor the area of interest, and because of their limited fields of views and the existence of static occluders, any given point can be seen only by a subset of cameras. This results in significant reduction in computational complexity of the likelihood computation.

We mentioned in Section 3 that the camera nodes can distinguish between the object and the occluders. To address non-perfect target distinction, one can introduce another random variable that indicates the event of detecting and recognizing the object and include its probability in the likelihood. We have not implemented this modification, however.

4.4. Obtaining Occluder Priors

Our tracker assumes the availability of priors for the moving occluder positions. In this section, we discuss some preliminary ideas on how these priors may be obtained. In Section 6.1, we investigate the trade-off between the accuracy of such priors and that of tracking.

Clearly, one could run a separate particle filter for each object and then fit Gaussians to the resulting particle distributions. This requires solving the data association problem, which in general requires substantial local and centralized processing. Another approach is to treat the states of all objects as a joint state and track them jointly [Doucet et al. 2002; Vihola 2007]. This approach, however, becomes computationally prohibitive for a large number of objects.

Another approach to obtaining the priors is to use a hybrid sensor network combining, for example, acoustic sensors in addition to cameras. As these sensors use less energy than cameras, they could be used to generate the priors for the moving occluders. An example of this approach can be found in Sheng and Hu [2005].

Yet another approach to obtaining the occluder priors involves reasoning about occupancy using the visual hull, as described in [Yang 2005]. The visual hull is obtained as follows. Locally at each camera, background subtracted images are vertically summed and thresholded to obtain a scan line. These scan lines are sent to the cluster head. The cluster head then computes the visual hull by back-projecting the blobs in the scan lines to cones in the room. The cones from the multiple cameras are intersected to compute the total visual hull. Since the resulting polygons are larger than the occupied areas and since phantom polygons that do not contain any objects may be present, the visual hull provides an upper bound on occupancy. The computation of the visual hull is relatively light-weight and does not require solving the data association problem. The visual hull can then be used to compute occluder priors by fitting ellipses to the polygons and using them as Gaussian priors. Alternatively, the priors can be assumed to be uniform distributions over these polygons. In this case, the computation of $q_{i,j}^{\mathrm{mv}}(x)$ in (8) would need to be modified.

Although the visual hull approach to computing occluder priors is quite appealing for a wireless sensor network implementation, several problems remain to be addressed, such as phantom removal [Yang 2005], which is necessary because their existence can cause the killing of many good particles.

5. SELECTION ALGORITHM

To conserve energy and make the tracking algorithm scalable, we perform camera node subset selection with tracking. Given the prior pdf of the position of the tracked object (which is obtained as particle-weight tuples from the tracker), the moving occluder priors, the positions and shapes of the static occluders, the camera fields of views and the camera noise parameters, we use the minimum mean square error (MSE) of the best linear estimate of the object position as a metric for selection. The best camera node subset is defined as the subset that minimizes the MSE metric. Every T time steps, a new subset of k cameras is selected, and the selected cameras are queried for measurements while the cameras that are not selected are put to sleep to conserve energy. Increasing T reduces the overhead of selection and increases energy efficiency by putting the camera nodes in sleep for longer periods. However a smaller T would result in a more up-to-date set of cameras to track the object, hence a better tracking performance.

Note that any selection metric has to be computed for a subset of cameras before the actual measurements are available; since if all the measurements were readily available, there wouldn't be any need for selection. The MSE of the best linear estimate of the object position is the expected squared error of localization, and it is computable locally at the cluster head with the available information (such as object and occluder statistics) but without the actual measurements. Therefore its computation does not require any communication to or from the camera nodes. In Section 6, we use the RMS error of the particle filter (PF) tracker as a measure of tracking performance. This metric, on the other hand, is computed after the measurements from the selected cameras are incorporated in the PF tracker. Unlike the best linear estimator, the PF is a nonlinear estimator. Due to these reasons, the MSE metric we use for selection is not exactly equal to the square of the tracker RMSE, however they are closely related.

To compute the MSE metric, the particle-weight tuples that are available from the tracker are advanced in time by one step before they are used as a prior for the selection step. This is done by passing them through an importance sampling step, as described in Section 4.1. Next, we describe the computation of the MSE metric.

The mean square error (MSE) of the object location x is given by $\mathrm{Tr}(\Sigma_x - \Sigma_{Zx}^T \Sigma_Z^{-1} \Sigma_{Zx})$ [Kailath et al. 1999], where $Z = [z_1 \; z_2 \; \ldots \; z_N]^T$ is the vector of measurements from all cameras, Σ_Z is the covariance of the measurements Z, Σ_x is the prior covariance of the object location x, and Σ_{Zx} is the cross-covariance between Z and x. The MSE formula can be interpreted as follows. The prior uncertainty in the object location (Σ_x) is reduced by the expected information from the measurements ($\Sigma_{Zx}^T \Sigma_Z^{-1} \Sigma_{Zx}$).

To compute the MSE, we assume the noisy perspective camera measurement model including occlusions presented in Section 3.1. To reduce computation, we also consider a weak perspective model [Trucco and Verri 1998]. The results using the two models are compared in simulations in Section 6.2. As we shall see, the results from the two models are very close. Hence, we use only the weak perspective model in the experiments (Section 7).

5.1. Perspective Camera Model

The MSE cannot be computed for the measurement model in Equation (4) as is because of the NaN. Also remember that the cameras that are occluded cannot be excluded from selection, since the cameras can be queried only after selection is performed. Therefore, for the computation of the selection metric only, we modify this model slightly: we assume that if a camera cannot see the object, the cluster head assumes the expected value of its measurement instead of an NaN. Let us denote this modified perspective measurement for camera i by $\breve{z}_i$.

$$\breve{z}_i = \begin{cases} f_i \frac{h_i(x)}{d_i(x)} + v_i, & \text{if } \eta_i = 1, \\ \mathrm{E}(\breve{z}_i), & \text{if } \eta_i = 0, \end{cases} \tag{10}$$

where $\mathrm{E}(\cdot)$ represents the expectation operator.

As indicated before, no communication is needed to or from the camera nodes to compute the MSE metric. Therefore, $\breve{z}_i$ is not actually received from camera i. It is just the model that the cluster head uses to compute the MSE metric. Once the selection is performed, actual measurements from the selected cameras are as in the original perspective model of Equation (4).

It can be shown that the expected value of $\breve{z}_i$ is

$$\mathrm{E}(\breve{z}_i) = f_i \mathrm{E}_x \left(\frac{h_i(x)}{d_i(x)} \right), \tag{11}$$

and that the covariances in the MSE metric for the perspective camera model in Equation (10) can be computed as follows.

$$\Sigma_{Zx}(i,:) = \mathrm{P}\{\eta_i = 1\} f_i \mathrm{E}_x \left(\frac{h_i(x)}{d_i(x)} \tilde{x}^T \right), \tag{12}$$

$$\Sigma_Z(i,j) = \mathrm{P}\{\eta_i = 1, \eta_j = 1\} \left[f_i f_j \mathrm{E}_x \left(\frac{h_i(x) h_j(x)}{d_i(x) d_j(x)} \right) - \mathrm{E}(\breve{z}_i)\mathrm{E}(\breve{z}_j) + \left\{ \begin{matrix} \sigma_{v_i}^2, & i = j \\ 0, & i \neq j \end{matrix} \right\} \right], \tag{13}$$

where $\Sigma_{Zx}(i,:)$ denotes the ith row of Σ_{Zx}, $\tilde{x} := x - \mu_x$, $h_i(x)$, $d_i(x)$ and f_i are defined in Figure 2, $\mathrm{P}\{\eta_i = 1, \eta_j = 1\} = \mathrm{E}_u(\mathrm{P}\{\eta_i = 1, \eta_j = 1 | u\})$, and $\sigma_{v_i}^2 = \mathrm{E}_x(\sigma_{v_i|x}^2)$. See Appendix D.1 for the derivation of Equations (11)–(13). All expectations over u or x are approximated by the weighted average over the particle distributions. For example,

$$\mathrm{P}\{\eta_i = 1, \eta_j = 1\} = \mathrm{E}_u \left(\mathrm{P}\{\eta_i = 1, \eta_j = 1 | u\} \right) \approx \sum_{\ell=1}^{L} w_\ell \mathrm{P}\{\eta_i = 1, \eta_j = 1 | u_\ell\},$$

where $\mathrm{P}\{\eta_i = 1, \eta_j = 1 | u_\ell\}$ is computed as described in Sections 4.2 and 4.3.

5.2. Weak Perspective Model

The computation of the MSE can be simplified by using a weak perspective camera model [Trucco and Verri 1998], which is an affine approximation to the perspective model, and it is valid when the object is far away from the camera. Note that the tracking is performed with the actual measurements that conform to the perspective model of Equation (4), regardless of the model used for selection.

Using the weak perspective assumption, we assume that the object's distance to the camera along its principal axis (i.e., $d_i(x)$ in Figure 2) is much greater than $\sqrt{\mathrm{Tr}(\Sigma_x)}$ (i.e., much greater than a measure of the width of the object location prior distribution). We further assume that $d_i(x)$ is much greater than the object's distance to its projection on the principal axis (i.e., $h_i(x)$ in Figure 2). Therefore, $d_i(x)$ can be approximated by $\bar{d}_i := d_i(\mu_x)$, where μ_x is the mean of the object's prior. Note that μ_x is available to the cluster head through the tracker. Thus, $\bar{d}_i$ is known and one can scale the measurements in the perspective camera model in Equation (10) by $\frac{\bar{d}_i}{f_i}$ without changing its information content. Denote these scaled measurements by

$$\frac{\bar{d}_i}{f_i}\check{z}_i := \tilde{z}_i = \begin{cases} a_i^T x + \tilde{v}_i, & \text{if } \eta_i = 1, \\ \mathrm{E}(\tilde{z}_i), & \text{if } \eta_i = 0, \end{cases} \tag{14}$$

where $\mathrm{E}(\tilde{z}_i) = a_i^T \mu_x$ (see Appendix D.2 for the derivation), $a_i^T = [\sin(\theta_i)\ -\cos(\theta_i)]$, and $\tilde{v}_i = \bar{d}_i / f_i v_i$. Here, we ignore a constant term involving the inner product of a_i with the position vector of camera i. This does not affect the MSE metric, however. To compute the variance of $\tilde{v}_i$, consider the conditional variance in Equation (2).

$$\begin{aligned} \sigma^2_{v_i|x} &= f_i^2\left(1 + \frac{h_i^2(x)}{d_i^2(x)}\right)^2 \sigma_\theta^2 + f_i^2\left(\frac{h_i^2(x) + d_i^2(x)}{d_i^4(x)}\right)\sigma^2_{\text{pos}} + \sigma^2_{\text{read}} \\ &\approx f_i^2\sigma_\theta^2 + \frac{f_i^2}{\bar{d}_i^2}\sigma^2_{\text{pos}} + \sigma^2_{\text{read}}, \end{aligned}$$

where we used the assumption that $d_i(x) \approx \bar{d}_i \gg h_i(x)$. Note that with the weak perspective approximation, $\sigma^2_{v_i|x}$ is not a function of x, and $\sigma^2_{v_i|x} = \sigma^2_{v_i}$. As such, the variance of $\tilde{v}_i$ is given by

$$\sigma^2_{\tilde{v}_i} = \frac{\bar{d}_i^2}{f_i^2}\sigma^2_{v_i} \approx \left(\sigma_\theta^2 + \frac{\sigma^2_{\text{read}}}{f_i^2}\right)\bar{d}_i^2 + \sigma^2_{\text{pos}}.$$

Thus, under the weak perspective model, we can assume that $\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_N$ are independent, zero mean Gaussian random variables.

Under the weak perspective model, the covariance matrices in the MSE metric change to the following (see Appendix D.2 for the derivation).

$$\Sigma_{Zx}(i,:) = \mathrm{P}\{\eta_i = 1\} a_i^T \Sigma_x, \tag{15}$$

$$\Sigma_Z(i,j) = \mathrm{P}\{\eta_i = 1, \eta_j = 1\}\left[a_i^T \Sigma_x a_j + \begin{Bmatrix} \sigma^2_{\tilde{v}_i}, & i = j \\ 0, & i \neq j \end{Bmatrix}\right]. \tag{16}$$

Note that the complexity of computing an element of Σ_Z or Σ_{Zx} is $O(LM)$ for both perspective and weak perspective models, where L is the number of particles and M is the number of moving occluders. This can be explained as follows. The complexity of computing $\mathrm{P}\{\eta_i = 1\}$ or $\mathrm{P}\{\eta_i = 1, \eta_j = 1\}$ is $O(LM)$. This term is common to both models and dominates the complexity. However, computing the MSE using the weak perspective model is still cheaper. To see this, compare Equations (13) and (16). The complexity of computing the expectation in Equation (13) is $O(L)$, while the complexity of the matrix

ALGORITHM 2: Greedy Selection – The Greedy Camera Node Selection Algorithm

```
Input: Object's prior (particle - weight tuples): ${u_\ell, w_\ell}_{\ell=1}^{L}$; dynamic occluders' priors:
    ${\mu_j, \Sigma_j}_{j=1}^{M}$; number of camera nodes to select: $k$; struct room (camera fields of views,
    positions and orientations; room's shape and sizes; static occluder information).
Output: Selected subset: $S$.
$S := \emptyset$;
for ($counter = 1 \dots k$) do
    $lowest := \infty$;
    for ($i = 1 \dots N$) do
        if ($i \notin S$) then
            $S := S \cup \{i\}$;
            $e :=$ MSE($S$);
            if ($e < lowest$) then
                $lowest := e$;
                $sel := i$;
            end
            $S := S \backslash \{i\}$;
        end
    end
    $S := S \cup \{sel\}$;
end
```

multiplication in Equation (16) is $O(1)$, assuming that Σ_x is precomputed.[1] If the computed $\mathrm{P}\{\eta_i = 1, \eta_j = 1 | u_\ell\}$ values can be stored and reused in the tracking algorithm, the computational savings previously described becomes even more significant.

The selection problem involves minimizing MSE(S) subject to $|S| = k$. Here, MSE(S) denotes the MSE computed using the cameras in subset $S \subset \{1, 2, \dots, N\}$. A brute-force search to find the optimal solution to this problem requires $O(N^k)$ trials. This can be too costly in a wireless camera network setting. Instead, we use the greedy selection algorithm given in Algorithm 2. The computational complexity of the greedy algorithm is $O(k^2MNL + k^4N)$, where k is the subset size, M is the number of moving occluders, N is the number of cameras, and L is the number of particles. This can be explained as follows. Line 7 of Algorithm 2 requires the computation of the MSE of the subset S. This amounts to computing $O(k)$ new elements in the covariance matrices ($O(kLM)$ computations) and a matrix inversion and two matrix multiplications ($O(k^3)$ computations). Therefore, the complexity of line 7 is $O(kLM+k^3)$. There are two for loops of size k and N surrounding this line; therefore, the total complexity is $O(k^2LMN+k^4N)$.

6. SIMULATION RESULTS

In a practical tracking setting, one is given the room structure (including information about the static occluders), the range of the number of moving occluders and their motion model, and the required object tracking accuracy. Based on this information, one needs to decide on the number of cameras to use in the room and the amount of prior information about the moving occluder positions needed and how to best obtain this information. Making these decisions involves several trade-offs, for example, between the occluder prior accuracy and the tracker performance, between the number of cameras used and the required occluder prior accuracy, and between the number of occluders present and the tracking performance. In this section we assume preselected

[1] Precomputing the other costly terms in Equation (13) requires $O(LN^2)$ computations. In practice, the total number of cameras N can be a lot larger than the selected number of cameras k. To achieve linear complexity on N on the overall selection algorithm, only Σ_x is assumed to be precomputed.

ALGORITHM 3: Top – Top-Level Algorithm Used in Simulations

```
Input: Struct room (camera fields of views, positions and orientations; room's shape and sizes;
       static occluder information); number of cameras to use: $k$.
Output: Estimated object track: $\hat{x}(t)$.
$\{u_\ell(0), w_\ell(0)\}_{\ell=1}^{L}$ = init_particles(room);
$objPos(0)$ = init_objects(room);
for ($t = 1, \ldots, T_{\max}$) do
    $objPos(t)$ = move_objects($objPos(t-1)$), room);
    $\{\mu_j(t), \Sigma_j(t)\}_{j=1}^{M}$ = obtain_priors($objPos(t)$); /* Section 6.1 */
    if $t == 1$ mod(T) /* Selection is performed every $T$ steps, see Section 5 */ then
        for $\ell = 1, \ldots, L$ do
            $\tilde{u}_\ell(t-1) \sim f(u|u_\ell(t-1))$; /* Section 4.1 */
        end
        $S$ = selection($\{\tilde{u}_\ell(t-1), w_\ell(t-1)\}_{\ell=1}^{L}$, $\{\mu_j(t), \Sigma_j(t)\}_{j=1}^{M}$, k,room);
    end
    $Z(t)$ = get_measurements($S$, $objPos(t)$, room);
    $\{u_\ell(t), w_\ell(t)\}_{\ell=1}^{L}$ = ASIR($\{u_\ell(t-1), w_\ell(t-1)\}_{\ell=1}^{L}$, $\{\mu_j(t), \Sigma_j(t)\}_{j=1}^{M}$, $Z(t)$, room);
    $\hat{x}(t) = \sum_{\ell=1}^{L} w_\ell(t) x_\ell(t)$;
end
```

subsets of cameras and explore these trade-offs in tracking simulations. We also investigate the effect of dynamically selecting the subsets of camera nodes on the tracking performance.

Algorithm 3 describes the top-level algorithm that we use in the simulations. In the algorithm, μ_j denotes the mean and Σ_j denotes the covariance of the prior for occluder j. The procedures `init_particles` and `init_objects` initializes the particles' and the objects' states. The procedure `obtain_priors` obtains the moving occluder prior statistics from virtual measurements, as will be explained later in Section 6.1. The procedure `selection` uses a fixed subset of camera nodes in Section 6.1 and the Greedy Selection algorithm (Algorithm 2) or other heuristics in Section 6.2. The procedure `get_measurements` obtains measurements from the cameras in subset S according to the perspective model of Equation (4). Procedure `ASIR` is given in Algorithm 1.

The procedure `move_objects` moves the objects in the room. We assume that the objects move according to random waypoints model. This is similar to the way we draw new particles from the importance density function, as discussed in Section 4.1 with the following differences.

—The objects are only in regimes MTD (move toward destination) or CD (change destination). There is no W (wait) regime.
—The objects choose their regimes deterministically, not randomly. If an object reaches its destination or is heading toward the inside of a static occluder or outside the room boundaries, it transitions to the CD regime.
—Objects go around each other instead of colliding.

The average speed of the objects is set to 1 unit per time step. The standard deviation of the noise added to the motion each time step is 0.33 units.

6.1. Tracking Simulations Using Pre-Selected Subsets of Cameras

In this section, we assume a square room of size 100 × 100 units and eight cameras placed around its periphery (see Figure 5). We explore the trade-offs previously mentioned in tracking simulations using preselected subsets of cameras. That is, the procedure `selection` in Algorithm 3 always returns the first k elements of

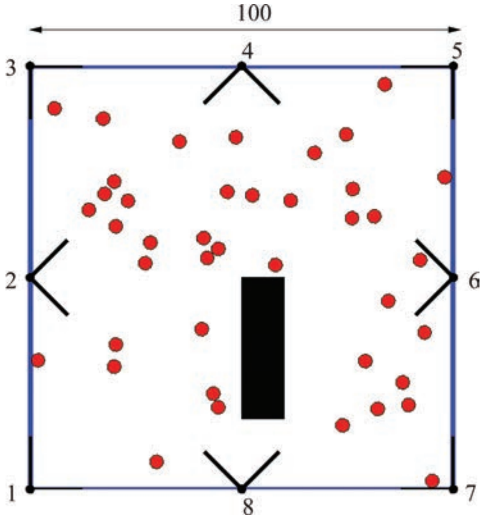

Fig. 5. The setup used in tracking simulations with preselected subsets of cameras.

$\{1, 7, 5, 3, 2, 4, 6, 8\}$, where k is the number of cameras to select. The cameras on the vertices are selected before the ones on the edges, because their fields of views have better coverage of the room. For two cameras, cameras that are placed orthogonally are used for better triangulation of the object position [Ercan et al. 2006].

The black rectangle in Figure 5 depicts a static occluder. Note, however, that in some of the simulations, we assume no static occluders. The cameras' fields of views are assumed to be 90°. The standard deviation of the camera position error is $\sigma_{\text{pos}} = 1$ unit, that of camera angle error is $\sigma_\theta = 0.01$ radians, and the read noise standard deviation is $\sigma_{\text{read}} = 2$ pixels. The diameter of each moving occluder is assumed to be $D = 3.33$ units. Figure 5 also shows a snapshot of the objects for $M = 40$ occluders. In the PF tracker, we use $L = 1{,}000$ particles and they are independently initialized according to a uniform distribution over the room. In each simulation, the object and the occluders move according to the random waypoints model for $T_{\max} = 4{,}000$ time steps.

To investigate trade-offs involving moving occluder prior accuracy, we need a measure for the accuracy of the occluder prior. To develop such a measure, we assume that the priors are obtained using a Kalman filter run on virtual measurements of the moving occluder positions of the form $y_j(t) = x_j(t) + \psi_j(t),\ \ j = 1, 2, \ldots, M$, where $x_j(t)$ is the true occluder position, $\psi_j(t)$ is white Gaussian noise with covariance $\sigma_\psi^2 I$, and $y_j(t)$ is the virtual measurement. The occluder position distributions estimated by the Kalman filter are used as occluder priors for the tracking algorithm (Algorithm 3, line 5), and the average RMSE of the Kalman filter (RMSE_{occ}) is used as a measure of occluder prior accuracy. Lower RMSE_{occ} means higher accuracy sensors or more computation is used to obtain the priors, which results in more energy consumption in the network. At the extremes, $\text{RMSE}_{\text{occ}} = 0$ (when $\sigma_\psi = 0$) corresponds to complete knowledge of the moving occluder positions, and $\text{RMSE}_{\text{occ}} = \text{RMSE}_{\max}$ (when $\sigma_\psi \to \infty$) corresponds to no knowledge of the moving occluder positions. Note that the worst case $\text{RMSE}_{\max}$ is finite because when there are no measurements about the occluder positions, one can simply assume that they are located at the center of the room. This corresponds to $\text{RMSE}_{\max} = 24.4$ units for the setup in Figure 5.

To implement the tracker for these two extreme cases, we modify the computation of the likelihood of the occlusion indicator functions as follows. We assign 0 or 1 to

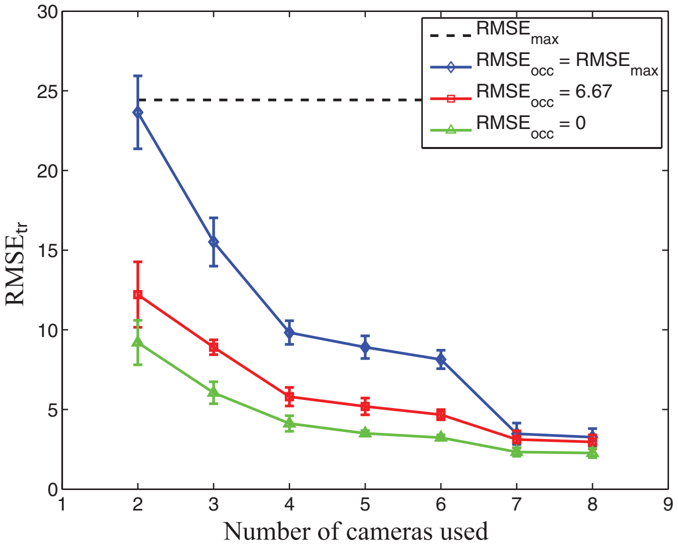

Fig. 6. Tracker RMSE versus the number of cameras for $M = 40$ moving occluders and one static occluder. The solid lines denote the averages and the error bars denote one standard deviation of the RMSE. The dotted line is the worst case average RMSE when no tracking is performed and the object is assumed to be at the center of the room.

$p(\eta|u)$ depending on the consistency of η with our knowledge about the occluders. For RMSE$_{occ} = 0$, that is, when we have complete information about the moving occluder positions, the moving occluders are treated as static occluders. On the other hand, for RMSE$_{occ}$ = RMSE$_{max}$, that is, when there is no information about the moving occluder positions, we check the consistency with only the static occluder and the limited field of view information to assign zero probabilities to some particles. For the example in Figure 4, we set $\mathrm{P}\left(\{\eta_1 = 1\} \cap \{\eta_2 = \gamma_2\}|u_{\ell_1}\right) = 0$, because if cam$_1$ sees the object, the object cannot be at x_{ℓ_1}. Any other occlusion indicator variable likelihood that is nonzero is set to 1. Note that for these two extreme cases, we no longer need the recursion discussed in Section 4.2 to compute the likelihood. Hence, the computational complexity is lighter compared to using Gaussian priors.

First in Figure 6 we plot the RMSE of the tracker (RMSE$_{tr}$) over five simulation runs for the two extreme cases of RMSE$_{occ} = 0$ and RMSE$_{occ}$ = RMSE$_{max}$ and for RMSE$_{occ} = 6.67$ (obtained by setting $\sigma_\psi = 8$) versus the number of cameras. The solid lines denote the averages and the error bars denote one standard deviation of the RMSE$_{tr}$. The dotted line represents the worst case average RMSE (RMSE$_{max}$) when there are no measurements and the object is assumed to be in the center of the room.

We then investigate the dependency of the tracker accuracy on the accuracy of the moving occluder priors. Figure 7 plots the RMSE for the tracker over five simulation runs versus RMSE$_{occ}$ for $N = 4$ cameras. In order to include the effect of moving occluder priors only, we used no static occluders in these simulations. RMSE$_{max}$ reduces to 21.4 units for this case. Note that there is around a factor of 2.34 times increase in average RMSE$_{tr}$ from the case of perfect occluder information (RMSE$_{occ} = 0$) to the case of no occluder information (RMSE$_{occ}$ = RMSE$_{max}$). Moreover, it is not realistic to assume that the occluder prior accuracy would be better than that of the tracker. With this consideration, the improvement reduces to around 1.94 times (this is obtained by noting that average RMSE$_{tr}$ = RMSE$_{occ}$ at around 3.72). The variation in the tracking accuracy, measured by the standard deviation of RMSE$_{tr}$, also improves

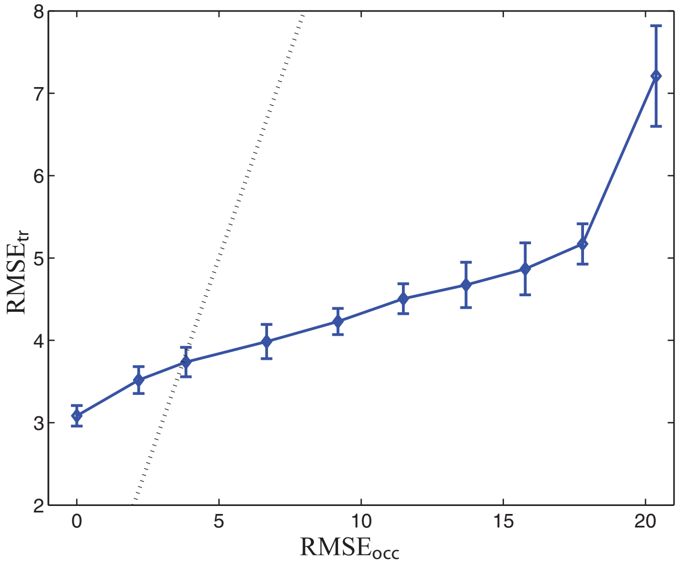

Fig. 7. Dependency of the tracker RMSE on the accuracy of the occluder priors for $N = 4$ cameras, $M = 40$ moving occluder, and no static occluders. The solid line denotes the averages and the error bars denote one standard deviation of RMSE$_{tr}$. The dotted line is for RMSE$_{tr}$ = RMSE$_{occ}$.

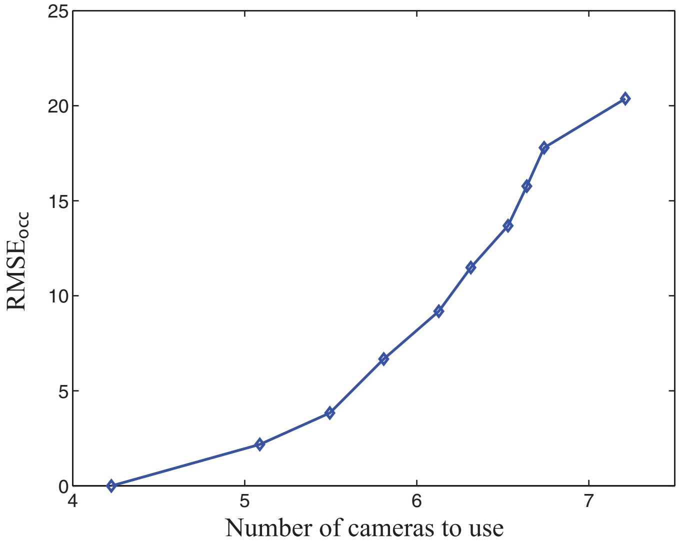

Fig. 8. Trade-off between the number of cameras and moving occluder prior accuracy for a target tracker average RMSE = 3 units for $M = 40$ moving occluders and no static occluders.

by about 3.4 times for these points. These observations suggest that obtaining prior information may not be worthwhile in practice, unless it can be obtained cheaply and to a reasonable accuracy.

The trade-off between RMSE$_{occ}$ and the number of cameras needed to achieve average RMSE$_{tr}$ = 3 is plotted in Figure 8. As expected, there is a trade-off between the number of cameras and the accuracy of the moving occluder priors, as measured by RMSE$_{occ}$.

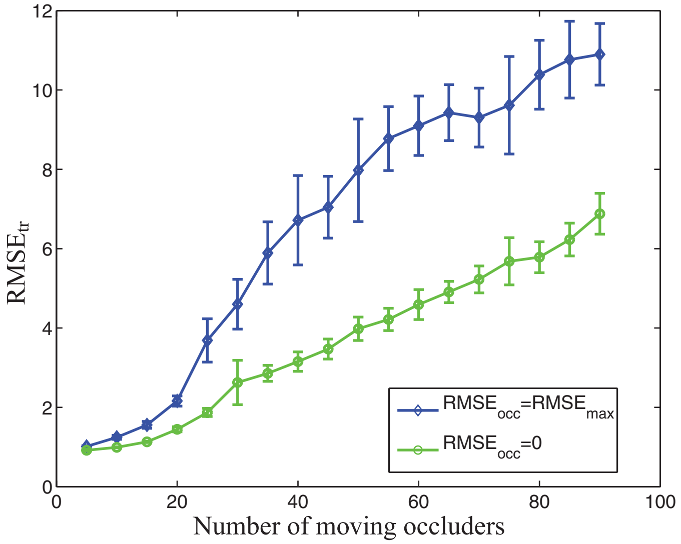

Fig. 9. Tracker RMSE versus the number of moving occluders for the two extreme cases, $\text{RMSE}_{\text{occ}} = 0$ and $\text{RMSE}_{\text{occ}} = \text{RMSE}_{\text{max}}$. The solid lines denote the averages and the error bars denote one standard deviation of RMSE. Here there are $N = 4$ cameras and no static occluders.

As more cameras are used, the accuracy of the prior information needed decreases. The plot suggests that if a large enough number of cameras is used, no prior information would be needed at all. Of course, having more cameras means more communications and processing cost. So, in the design of a tracking system, one needs to compare the cost of deploying more cameras to that of obtaining better occluder priors.

Next, we explore the question of how the needed moving occluder prior accuracy depends on the number of occluders present. To do so, in Figure 9 we plot the RMSE_{tr} versus the number of moving occluders for the two extreme cases, $\text{RMSE}_{\text{occ}} = 0$ and $\text{RMSE}_{\text{occ}} = \text{RMSE}_{\text{max}}$. The solid lines denote the averages and the error bars denote one standard deviation of RMSE_{tr}. Note that the difference between the RMSE_{tr} for the two cases is the potential improvement in the tracking performance achieved by having occluder prior information. When there are very few moving occluders, prior information does not help (because the object is not occluded most of the time). As the number of occluder increases, prior information becomes more useful. But the difference in RMSE_{tr} between the two extreme cases decreases when too many occluders are present (because the object becomes occluded most of the time).

It is also notable in Figures 6–9 that the variations of RMSE_{tr}, measured by the standard deviations, also increase as RMSE_{occ} increases (i.e., moving occluder prior accuracy decreases).

In Section 4.3, we mentioned that the complexity of computing the likelihood given u_ℓ is exponential in the number of occluded cameras among the ones assigned to the partition that particle ℓ belongs to. We proposed that in practice, the complexity is significantly lower than that of the exponential in N because the number of assigned cameras to a partition is a fraction of N. To verify this, in Figure 10 we plot the average CPU time (per time step) used to compute the likelihood relative to that of the $\text{RMSE}_{\text{occ}} = \text{RMSE}_{\text{max}}$ case for two cameras versus the total number of cameras in the room. The simulations were performed on a 3 GHz Intel Xeon Processor. Note that the rate of increase of the CPU time using priors is significantly lower than 2^N, where N is the number of cameras used, and it is close to the rate of increase of the

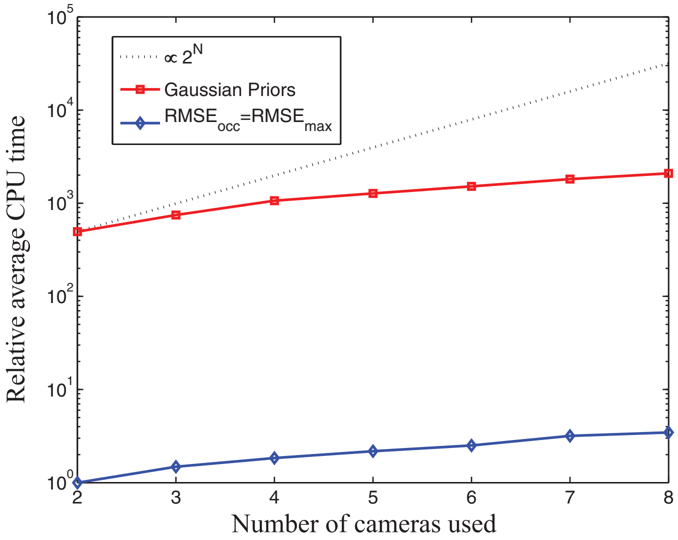

Fig. 10. Average CPU time for computing the likelihoods relative to that for the case of two cameras and no occluder priors, that is, $\text{RMSE}_{\text{occ}} = \text{RMSE}_{\text{max}}$. Here $M = 40$ and there is one static occluder.

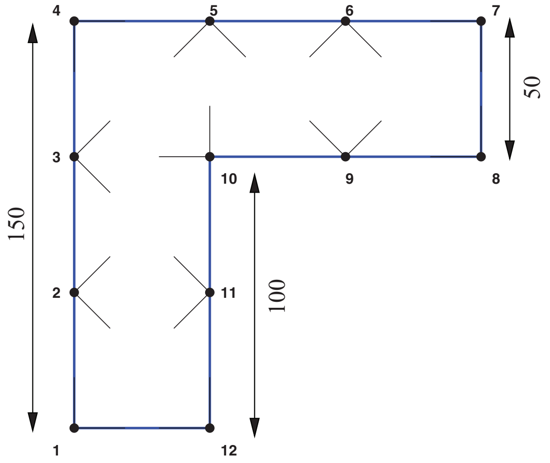

Fig. 11. The setup used in simulations to test the effect of dynamically selecting subsets of camera nodes used in tracking on the tracking performance.

$\text{RMSE}_{\text{occ}} = \text{RMSE}_{\text{max}}$ case. In fact, the rate of increase for this particular example is close to linear in N.

6.2. Effect of Selection Algorithms on Tracking Performance

In this section, we explore the effect of dynamically selecting the subset of camera nodes used in tracking on the tracking performance. An ‘L’ shaped room (see Figure 11) is used in order to emphasize the effect of selection. We use Algorithm 3 with the procedure `selection` employing different selection methods to compare their tracking performance. We assume $M = 40$ moving occluders and no static occluders. The moving

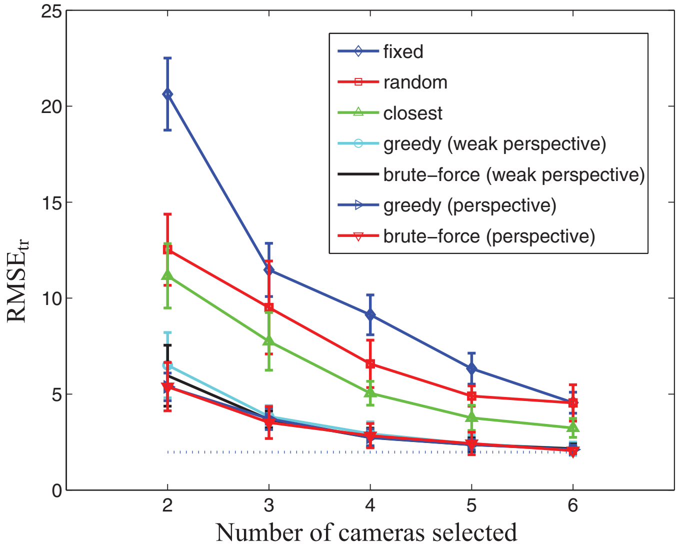

Fig. 12. The tracker RMSE versus the number of cameras for $M = 40$. The solid lines denote the averages and the error bars denote one standard deviation of the RMSE. The dotted line is the average tracker RMSE achieved by using all 12 cameras. The worst case average RMSE (RMSE$_{max}$) when no tracking is performed and the object is assumed to be at the center of the room is 47.9 units (not shown). The occluder prior accuracy (RMSE$_{occ}$) is 11.5 units.

occluder prior accuracy is RMSE$_{occ}$ = 11.5. The selection is performed every $T = 5$ time steps. All other parameters are the same as in Section 6.1.

We compare the greedy selection algorithm (Algorithm 2) to the brute-force method, in which an exhaustive search is performed to find the subset that minimizes the localization MSE, as well as to the following heuristics.

—*Fixed*. Use a preselected set of cameras. This amounts to the first k elements of $\{1, 4, 7, 10, 8, 12, 3, 5, 2, 6, 11, 9\}$.
—*Random*. Use randomly selected cameras.
—*Closest*. Pick k closest cameras to the object location mean.

In addition, the simulations for the greedy algorithm and the brute-force method are performed using both the perspective (Section 5.1) and weak perspective (Section 5.2) camera models. The performance of different selection methods are also compared to using all cameras.

Figure 12 compares the tracking performance achieved by different selection methods averaged over ten simulation runs, for $k = 2$ to six selected cameras. The solid lines denote the averages and the error bars denote one standard deviation of RMSE$_{tr}$. The dotted line is the average tracker RMSE achieved by using all 12 cameras. The worst case average RMSE (RMSE$_{max}$) when no tracking is performed and the object is assumed to be at the center of the room is 47.9 (not shown in the figure). As seen from the figure, the greedy selection algorithm performs close to the brute-force method and outperforms the other selection heuristics. The fixed selection heuristic performs worst because the subset is not updated dynamically and therefore some of the selected cameras cannot see the object due to the “L” shape of the room. Particularly note that the closest selection heuristic does not perform as well as one might expect. The reason for this is two-fold. First, cameras are angle sensors, not range sensors. Therefore,

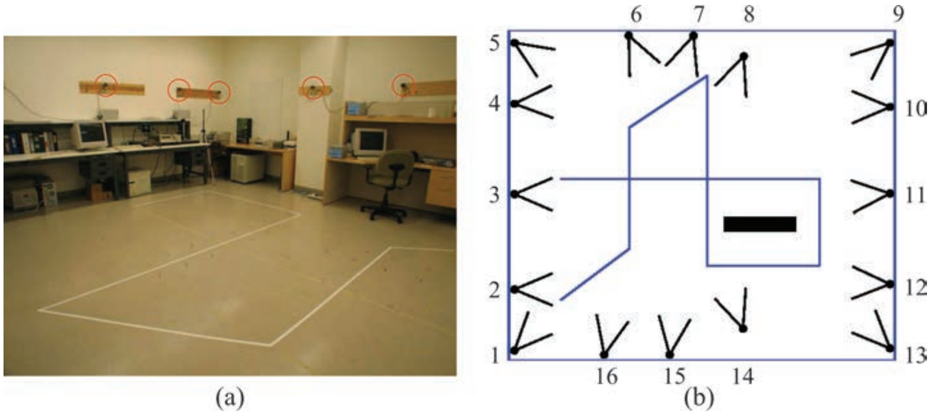

Fig. 13. Experimental setup. (a) View of lab (cameras are circled). (b) Relative locations of cameras and virtual static occluder. The solid line shows the actual path of the object to track.

choosing the closest sensor is not the right thing to do with cameras. A camera oriented orthogonally to the direction of highest uncertainty in the object location would perform better. Such a camera is favored by the MSE metric. Second, the MSE metric favors a more diverse selection in the viewing angle for $k \geq 2$, compared to the closest selection heuristic. These ideas are also illustrated in Figure 7 of Ercan et al. [2006]. In addition, the fixed, random, and closest selection heuristics do not take static or moving occlusions into account.

We also observe from Figure 12 that the performance achieved using the weak perspective model is within one standard deviation of that of the perspective model. This justifies the use of the weak perspective model for selection in order to save computation energy. Another interesting observation is by only using six cameras and performing greedy selection, the performance of using all 12 cameras (dotted line in the figure) can be achieved.

7. EXPERIMENTAL RESULTS

We tested our tracking algorithm in an experimental setup consisting of 16 Web cameras placed around a $22' \times 19'$ room. The horizontal FOV of the cameras used is 47°. A picture of the lab is shown in Figure 13(a) and the relative positions and orientations of the cameras in the room are provided in Figure 13(b). Each pair of cameras is connected to a PC via IEEE 1394 (FireWire) interface and each can provide 8-bit 3-channel (RGB) raw video at 7.5 frames/s. The data from each camera is processed independently, as described in Section 3. The measurement data is then sent to a central PC (cluster head) where further processing is performed.

The object follows the predefined path (shown in Figure 13(b)) with no occlusions present and $T_{\text{max}} = 200$ time steps of data is collected. The effect of static and moving occluders is simulated using one virtual static occluder and $M = 20$ virtual moving occluders: we threw away the measurements from the cameras that would have been occluded had there been real occluders. The virtual moving occluders walk according to the model explained in Section 6. D is chosen 12 inches for the moving occluders. The camera noise parameters were assumed $\sigma_{\text{pos}} = 1$ inch, $\sigma_{\text{read}} = 2$ pixels, and $\sigma_\theta = 0.068$ radians in the computation of the likelihood or the MSE selection metric.

We first explore the performance of our tracker in experiments versus number of cameras used for different occluder prior accuracies. For selection of the camera

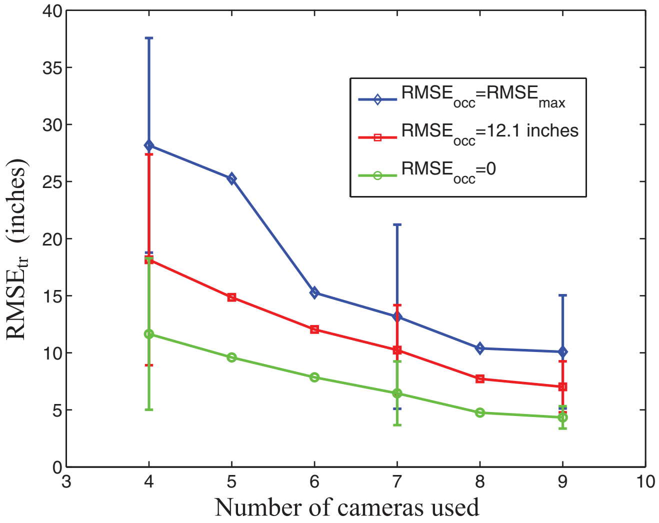

Fig. 14. Experimental results on the tracker RMSE versus the number of cameras for different occluder prior accuracies. There are $M = 20$ moving occluders and one static occluder. The solid lines denote the averages and the error bars denote one standard deviation of RMSE over different simulation runs. Not all error bars are shown in order not to clutter the graphs. A fixed selection heuristic is used. The worst case average RMSE (RMSE$_{\text{max}}$) when no tracking is performed and the object is assumed to be at the center of the room is 77.5 inches (not shown).

subsets, we use the fixed selection heuristic which returns the first k elements of $\{1, 13, 9, 5, 3, 8, 11, 16, 6\}$. Figure 14 plots the RMSE of the tracker over 50 runs for the two extreme cases of $\text{RMSE}_{\text{occ}} = \text{RMSE}_{\text{max}} = 77.5$ inches and $\text{RMSE}_{\text{occ}} = 0$ and for the case of $\text{RMSE}_{\text{occ}} = 12.1$ inches versus the number of cameras. Because the data set is for only 200 time steps, averaging in the computation of the RMSE is performed starting from $t = 5$ to reduce the effects of the initial transients. The solid lines in the figure denote the averages and the error bars denote one standard deviation of RMSE over different runs. Not all error bars are shown in order not to clutter the graphs. There is some difference in the performance between the three cases throughout the entire plot. However the difference is not substantial considering the variations around the average RMSEs, as shown by the error bars. However, the average RMSEs improve as the number of cameras used or as the available occluder prior accuracy increases, agreeing with the trade-offs discussed in Section 6.

Next, we explore the effect of dynamically selecting the subset of camera nodes used in tracking on the tracking performance. Figure 15 plots the RMSE of the tracker over 50 runs for the different selection methods described in Section 6.2. The solid lines denote the averages and the error bars denote one standard deviation of RMSE over different runs. Not all error bars are shown in order not to clutter the graphs. We used only the weak perspective model in the computation of the selection metric for experiments, since the results achieved by using perspective and weak perspective models are shown to be close in Section 6.2. The occluder prior accuracy (RMSE$_{\text{occ}}$) is 12.1 inches. Also in the figure, the average tracker RMSE achieved by using all 16 cameras is denoted by the dotted line. The average tracker RMSE for the greedy selection method is close to that of the brute-force method and better than the other heuristics. Note that in this case the variations of the RMSEs values are also large.

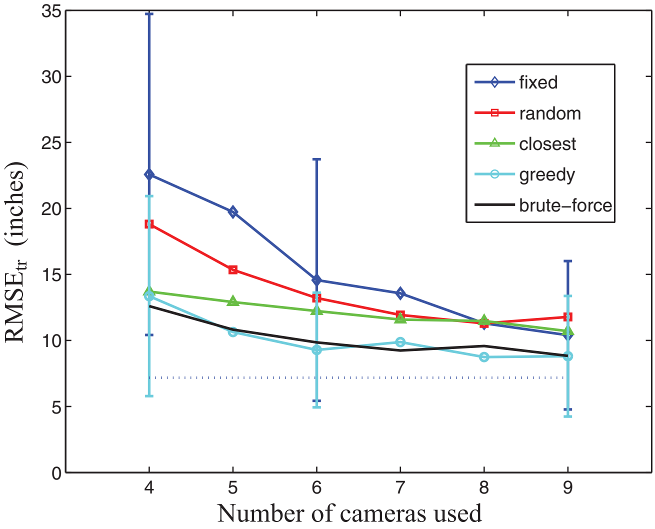

Fig. 15. Experimental results on the effect of dynamically selecting the subset of camera nodes used in tracking. Tracker RMSE versus the number of cameras for different selection heuristics are shown. The error bars represent one standard deviation of the RMSE. Not all error bars are shown in order not to clutter the graphs. $M = 20$ and one static occluder is present. The dotted line is the average tracker RMSE achieved by using all 16 cameras. The worst case average RMSE ($RMSE_{max}$) when no tracking is performed and the object is assumed to be at the center of the room is 77.5 inches (not shown). The occluder prior accuracy ($RMSE_{occ}$) is 12.1 inches. The weak perspective camera model is used for the greedy algorithm and the brute-force method.

8. CONCLUSIONS

We described a sensor network approach for tracking a single object in a structured environment using multiple cameras. Instead of tracking all objects in the environment, which is computationally very costly, we track only the target object and treat others as occluders. The tracker is provided with complete information about the static occluders and some prior information about the moving occluders. A key contribution of this article is developing a systematic way to incorporate this information into the tracker formulation.

Using preselected subsets of cameras, we explored the trade-offs involving the occluder prior accuracy, the number of cameras used, the number of occluders present, and the accuracy of tracking. Based on our simulations, we generally found the following.

—Obtaining moving occluder prior information may not be worthwhile in practice, unless it can be obtained cheaply and to a reasonable accuracy.

—There is a trade-off between the number of cameras used and the amount of occluder prior information needed. As more cameras are used, the accuracy of the prior information needed decreases. Having more cameras, however, means incurring higher communications and processing cost. So, in the design of a tracking system, one needs to compare the cost of deploying more cameras to that of obtaining more accurate occluder priors.

—The amount of prior occluder position information needed depends on the number of occluders present. When there are very few moving occluders, prior information does not help (because the object is not occluded most of the time). When there is a moderate number of occluders, prior information becomes more useful. However, when there are too many occluders, prior information becomes less useful (because the object becomes occluded most of the time).

We also explored the effect of dynamically selecting the subsets of camera nodes used in tracking on the tracking performance. The minimum MSE of the best linear estimate of object position based on camera measurements is used as a metric for selection. We showed through simulations that a greedy selection algorithm performs close to the brute-force method and outperforms other selection heuristics. We also showed that the performance achieved by greedily selecting a fraction of the cameras is close to that of using all cameras, which translates to savings in bandwidth and energy.

APPENDIX

A. LIST OF SELECTED SYMBOLS

Symbol	Description	Section
x	Position of the tracked object	3
μ_x	Mean of x	5
Σ_x	Covariance of x	5
u	State of the object	4
S	A subset of selected cameras	5
k	Number of selected cameras	5
T	Number of time steps between selections	5
$T_{\max}$	Total number of time steps in one simulation run	6
D	Diameter of moving occluders	3
M	Total number of moving occluders	3
j	Enumerator for moving occluders	4
x_j	Position of moving occluder j	4
μ_j	Mean of occluder j's prior	4
Σ_j	Covariance of occluder j's prior	4
L	Total number of particles	4
ℓ	Enumerator for particles	4
u_ℓ	State of particle ℓ	4
w_ℓ	Weight of particle ℓ	4
x_ℓ	Position of particle ℓ	4
N	Total number of cameras	3
i	Enumerator for cameras	3
θ_i	Orientation (yaw angle) of camera i	3
η_i	Occlusion indicator variable for camera i	3
z_i	Measurement from camera i	3
f_i	Focal length of camera i	3
v_i	Additive Gaussian noise to z_i	3
σ_θ	Standard deviation of camera orientation (θ_i) inaccuracy	3
σ_{pos}	Standard deviation of the camera position inaccuracy	3
σ_{read}	Standard deviation of the read-out noise	3

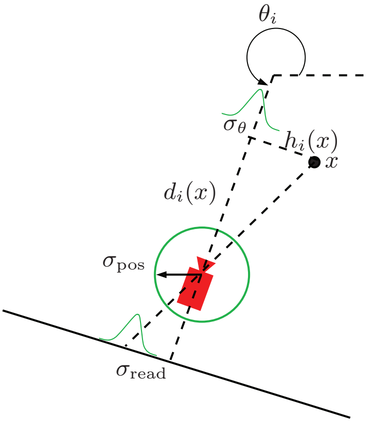

Fig. 16. Illustrations of read noise and camera calibration inaccuracies leading to the camera measurement noise v_i.

B. DERIVATION OF THE CAMERA MEASUREMENT NOISE VARIANCE

To derive the conditional mean and variance of camera measurement noise v_i given object position x, we assume the read-out noise and the inaccuracies in camera position and orientation calibration to be zero mean with variances σ^2_{read}, σ^2_{pos}, and σ^2_θ, respectively (see Figure 16). Further, we assume that these sources of noise are mutually independent. Assume that the camera is at $x_{ci} := [x_{ci1}\ x_{ci2}]^T$ and the object is at $x := [x_1\ x_2]^T$. Then $h_i(x)$ and $d_i(x)$ are given by

$$\begin{aligned} h_i(x) &= \sin(\theta_i)(x_1 - x_{ci1}) - \cos(\theta_i)(x_2 - x_{ci2}),\\ d_i(x) &= -\cos(\theta_i)(x_1 - x_{ci1}) - \sin(\theta_i)(x_2 - x_{ci2}). \end{aligned}$$

Taking the partial derivatives of z_i in Equation (1) with respect to θ_i, x_{c1}, and x_{c2}, we obtain the following.

$$\begin{aligned} \frac{\partial z_i}{\partial \theta_i} &= -f_i\left(1 + \frac{h_i^2(x)}{d_i^2(x)}\right),\\ \frac{\partial z_i}{\partial x_{ci1}} &= -f_i\left(\frac{d_i(x)\sin(\theta_i) + h_i(x)\cos(\theta_i)}{d_i^2(x)}\right),\\ \frac{\partial z_i}{\partial x_{ci2}} &= f_i\left(\frac{d_i(x)\cos(\theta_i) - h_i(x)\sin(\theta_i)}{d_i^2(x)}\right). \end{aligned}$$

Let us denote the zero mean errors in camera positions x_{ci1} and x_{ci2} by $\Delta_{x_{ci1}}$ and $\Delta_{x_{ci2}}$, respectively. We assume that the variances of the position error in both directions are equal and given by σ^2_{pos}. The read-out noise and error in camera orientation are denoted by Δ_{read} and Δ_θ, respectively. Then

$$\mathrm{E}(v_i|x) \approx \frac{\partial z_i}{\partial \theta_i}\mathrm{E}(\Delta_\theta) + \frac{\partial z_i}{\partial x_{ci1}}\mathrm{E}(\Delta_{x_{ci1}}) + \frac{\partial z_i}{\partial x_{ci2}}\mathrm{E}(\Delta_{x_{ci2}}) + \mathrm{E}(\Delta_{\text{read}}) = 0.$$

Using the independence assumption between the error sources, we obtain the following conditional variance.

$$\begin{aligned}\sigma^2_{v_i|x} &\approx \left(\frac{\partial z_i}{\partial \theta_i}\right)^2 \sigma^2_{\theta_i} + \left(\frac{\partial z_i}{\partial x_{ci1}}\right)^2 \sigma^2_{\Delta_{x_{ci1}}} + \left(\frac{\partial z_i}{\partial x_{ci2}}\right)^2 \sigma^2_{\Delta_{x_{ci2}}} + \sigma^2_{\text{read}} \\ &= f_i^2 \left(1 + \frac{h_i^2(x)}{d_i^2(x)}\right)^2 \sigma^2_\theta + f_i^2 \left(\frac{h_i^2(x) + d_i^2(x)}{d_i^4(x)}\right) \sigma^2_{\text{pos}} + \sigma^2_{\text{read}}.\end{aligned}$$

C. DERIVATION OF EQUATION (8)

In this section, the derivation of Equation (8) is provided. In Section 4.2, we assumed the rotation of the coordinate system where the major axis of moving occluder j's prior is horizontal. Consider another rotation, where the rectangle $A_i(x)$ is horizontal. Let the mean and the covariance matrix of occluder j be μ'_j and Σ'_j at this orientation. Let $\theta_{i,j}(x) := \theta$ for brevity. Without loss of generality, assuming that the camera is at the origin, we have the relations $\mu'_j = R_\theta^T \mu_j$, $\Sigma'_j = R_\theta^T \Sigma_j R_\theta$, and $|\Sigma'_j| = |\Sigma_j| = \frac{\sigma_j^4}{\alpha_j}$, where R_θ is the rotation matrix by θ. Then $q^{\text{mv}}_{i,j}(x)$ is found by

$$q^{\text{mv}}_{i,j}(x) = \int_{-\frac{D}{2}}^{\frac{D}{2}} \int_0^{\|x\|} \frac{1}{2\pi\sqrt{|\Sigma'_j|}} \exp\left(-\frac{1}{2}(x' - \mu'_j)^T {\Sigma'_j}^{-1}(x' - \mu'_j)\right) dx' \tag{17}$$

$$= \int_{-\frac{D}{2}}^{\frac{D}{2}} \int_0^{\|x\|} \frac{\sqrt{\alpha_j}}{2\pi\sigma_j^2} \exp\left(\underbrace{-1/2\, {x'}^T {\Sigma'_j}^{-1} x'}_{\mathbf{A}} + \underbrace{{\mu'_j}^T {\Sigma'_j}^{-1} x'}_{\mathbf{B}} \underbrace{-1/2\, {\mu'_j}^T {\Sigma'_j}^{-1} \mu'_j}_{\mathbf{C}}\right) dx'. \tag{18}$$

Let us look at each preceding term defined separately. First define

$$G := \begin{bmatrix} \cos\theta & -\sin\theta \\ \sqrt{\alpha_j}\sin\theta & \sqrt{\alpha_j}\cos\theta \end{bmatrix} = [g_1\, g_2],$$

where g_1 and g_2 are the columns of G. Note that

$${\Sigma'_j}^{-1} = R_\theta^T {\Sigma_j}^{-1} R_\theta = \frac{1}{\sigma_j^2} G^T G.$$

Define $x' := [x'_1\ x'_2]^T$, then

$$\mathbf{A} = -\frac{1}{2} {x'}^T {\Sigma'_j}^{-1} x' = -\frac{1}{2\sigma_j^2}\left(\|g_1\|^2 {x'_1}^2 + \|g_2\|^2 {x'_2}^2 + 2 g_1^T g_2 x'_1 x'_2\right).$$

To compute $\mathbf{B}$, define

$$O := \begin{bmatrix} \cos\theta & -\sin\theta \\ \alpha_j \sin\theta & \alpha_j \cos\theta \end{bmatrix} = [o_1\, o_2],$$

where o_1 and o_2 are the columns of O. Then

$$\mathbf{B} = {\mu'_j}^T {\Sigma'_j}^{-1} x' = \left({\mu_j}^T R_\theta\right)\left(R_\theta^T {\Sigma_j}^{-1} R_\theta\right) x' = \frac{1}{\sigma_j^2} {\mu_j}^T O x' = \frac{1}{\sigma_j^2}\left({\mu_j}^T o_1 x'_1 + {\mu_j}^T o_2 x'_2\right)$$

Finally,

$$\mathbf{C} = -\frac{1}{2} {\mu'_j}^T {\Sigma'_j}^{-1} \mu'_j = -\frac{1}{2} {\mu_j}^T R_\theta R_\theta^T {\Sigma_j}^{-1} R_\theta R_\theta^T \mu_j = -\frac{1}{2} {\mu_j}^T {\Sigma_j}^{-1} \mu_j.$$

By substituting **A**, **B**, and **C** into Equation (18) and using the formula

$$\int_{c_1}^{c_2} \exp(-\zeta\rho^2 + 2\xi\rho)d\rho = \frac{1}{2}\sqrt{\frac{\pi}{\zeta}}\exp\left(\frac{\xi^2}{\zeta}\right)\left[\operatorname{erf}\left(\frac{\zeta c_2 - \xi}{\sqrt{\zeta}}\right) - \operatorname{erf}\left(\frac{\zeta c_1 - \xi}{\sqrt{\zeta}}\right)\right], \tag{19}$$

we reach

$$\begin{aligned} q_{i,j}^{\mathrm{mv}}(x) &= \frac{1}{2}\sqrt{\frac{\alpha_j}{2\pi\sigma_j^2\|g_1\|^2}}\exp\left(-\frac{1}{2}{\mu_j}^T{\Sigma_j}^{-1}\mu_j\right) \\ &\int_{-\frac{D}{2}}^{\frac{D}{2}}\exp\left(\frac{2{\mu_j}^T o_2 x_2' - \|g_2\|^2{x_2'}^2}{2\sigma_j^2} + \frac{({\mu_j}^T o_1 - g_1^T g_2 x_2')^2}{2\sigma_j^2\|g_1\|^2}\right) \\ &\left[\operatorname{erf}\left(\frac{\|g_1\|^2\|x\| - \mu_j o_1 + g_1^T g_2 x_2'}{\sqrt{2}\sigma_j\|g_1\|}\right) + \operatorname{erf}\left(\frac{\mu_j o_1 - g_1^T g_2 x_2'}{\sqrt{2}\sigma_j\|g_1\|}\right)\right] dx_2'. \end{aligned}$$

Notice that there are three places where we have $g_1^T g_2 x_2' = x_2'(\alpha_j - 1)\sin(2\theta)/2$. Here, $\sqrt{\alpha_j} \geq 1$ is the ratio of the major axis of occluder j's prior to the minor axis (see Figure 3). We assume that α_j is not too big and D is small with respect to σ_j, such that $g_1^T g_2 x_2'$ can be ignored.

$$\begin{aligned} q_{i,j}^{\mathrm{mv}}(x) &\approx \frac{1}{2}\sqrt{\frac{\alpha_j}{2\pi\sigma_j^2\|g_1\|^2}}\exp\left(-\frac{1}{2}{\mu_j}^T{\Sigma_j}^{-1}\mu_j\right)\exp\left(\frac{({\mu_j}^T o_1)^2}{2\sigma_j^2\|g_1\|^2}\right) \\ &\left[\operatorname{erf}\left(\frac{\|g_1\|^2\|x\| - {\mu_j}^T o_1}{\sqrt{2}\sigma_j\|g_1\|}\right) + \operatorname{erf}\left(\frac{{\mu_j}^T o_1}{\sqrt{2}\sigma_j\|g_1\|}\right)\right] \\ &\int_{-\frac{D}{2}}^{\frac{D}{2}}\exp\left(\frac{2{\mu_j}^T o_2 x_2' - \|g_2\|^2{x_2'}^2}{2\sigma_j^2}\right) dx_2'. \end{aligned}$$

The formula in Equation (19) is then used once more to get Equation (8). Note that when α_j is too big, the prior of occluder j can be treated as a degenerate 1D Gaussian function in 2D, and one could still perform the integral in Equation (17) using Equation (19) once, as the prior is effectively one dimensional. However, we did not implement this modification.

To test the validity of the preceding approximation, we performed several simulations. We selected random priors for the occluders and ran Monte-Carlo simulations to find $q_{i,j}^{\mathrm{mv}}(x)$ empirically. We compared these values to the ones computed by using Equation (8). For example, in Figure 17, you see Monte-Carlo runs for 16,000 random points. The solid line is for denoting $y = x$ and the error bars represent the $\pm 3\sigma$ tolerance for the Monte-Carlo simulation. Here $D = 3.33$, $\sigma_j = 2$, $\alpha_j = 4$. For this example, although $D > \sigma_j$ and α_j is considerably greater than 1, most of the 16,000 points still lie in the $\pm 3\sigma$ tolerance range.

D. DERIVATION OF THE LOCALIZATION MSE

In this section, we derive the formulas for the covariance matrices required for the computation of the MSE for both perspective and weak perspective model.

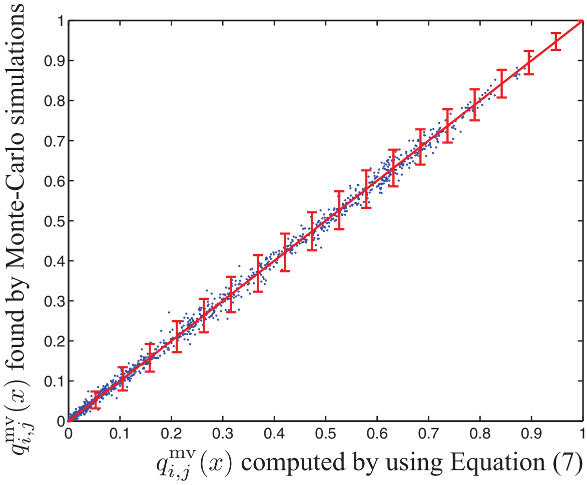

Fig. 17. Monte-Carlo simulations to test the accuracy of Equation (8). Here $D = 3.33$, $\sigma_j = 2$, $\alpha = 4$.

D.1. Perspective Model

The expected value of $\breve{z}_i$ can be simplified as follows.

$$\mathrm{E}(\breve{z}_i) = \mathrm{P}\{\eta_i = 1\}\mathrm{E}\left(f_i\frac{h_i(x)}{d_i(x)} + v_i\right) + \mathrm{P}\{\eta_i = 0\}\mathrm{E}(\breve{z}_i)$$

$$(1 - \mathrm{P}\{\eta_i = 0\})\mathrm{E}(\breve{z}_i) = \mathrm{P}\{\eta_i = 1\}\left[\mathrm{E}_x\left(f_i\frac{h_i(x)}{d_i(x)}\right) + \mathrm{E}_x(\mathrm{E}_{v_i}(v_i|x))\right].$$

Using the fact that $v_i|x$ is zero mean (see Appendix B), we find $\mathrm{E}(\breve{z}_i) = f_i\mathrm{E}_x\left(\frac{h_i(x)}{d_i(x)}\right)$. To compute the elements of the covariance matrices required for the computation of the MSE, define $\tilde{x} := x - \mu_x$. Then

$$\begin{aligned}\Sigma_{Zx}(i,:) &= \mathrm{E}((\breve{z}_i - \mathrm{E}(\breve{z}_i))\tilde{x}^T) = \mathrm{E}(\breve{z}_i\tilde{x}^T)\\ &= \mathrm{P}\{\eta_i = 1\}\mathrm{E}\left(\left(f_i\frac{h_i(x)}{d_i(x)} + v_i\right)\tilde{x}^T\right) + \mathrm{P}\{\eta_i = 0\}\mathrm{E}(\breve{z}_i)\mathrm{E}(\tilde{x}^T)\\ &= \mathrm{P}\{\eta_i = 1\}f_i\mathrm{E}_x\left[\left(\frac{h_i(x)}{d_i(x)}\right)\tilde{x}^T\right],\end{aligned}$$

where $\Sigma_{Zx}(i,:)$ denotes the ith row of Σ_{Zx}. Similarly,

$$\begin{aligned}\Sigma_Z(i,j) \;=\;& \mathrm{E}(\breve{z}_i\breve{z}_j) - \mathrm{E}(\breve{z}_i)\mathrm{E}(\breve{z}_j)\\ =\;& \mathrm{P}\{\eta_i = 1, \eta_j = 1\}\mathrm{E}(\breve{z}_i\breve{z}_j|\eta_i = 1, \eta_j = 1) + \mathrm{P}\{\eta_i = 1, \eta_j = 0\}\mathrm{E}(\breve{z}_i\breve{z}_j|\eta_i = 1, \eta_j = 0)\\ &+ \mathrm{P}\{\eta_i = 0, \eta_j = 1\}\mathrm{E}(\breve{z}_i\breve{z}_j|\eta_i = 0, \eta_j = 1)\\ &+ \mathrm{P}\{\eta_i = 0, \eta_j = 0\}\mathrm{E}(\breve{z}_i\breve{z}_j|\eta_i = 0, \eta_j = 0) - \mathrm{E}(\breve{z}_i)\mathrm{E}(\breve{z}_j).\end{aligned}$$

Note that $\mathrm{E}(\breve{z}_i\breve{z}_j|\eta_i = 1, \eta_j = 0) = \mathrm{E}(\breve{z}_i\breve{z}_j|\eta_i = 0, \eta_j = 1) = \mathrm{E}(\breve{z}_i\breve{z}_j|\eta_i = 0, \eta_j = 0) = \mathrm{E}(\breve{z}_i)\mathrm{E}(\breve{z}_j)$. Therefore,

$$\begin{aligned}
\Sigma_Z(i, j) &= \mathrm{P}\{\eta_i = 1, \eta_j = 1\} \left(\mathrm{E}\left(f_i \frac{h_i(x)}{d_i(x)} + v_i \right) \mathrm{E}\left(f_j \frac{h_j(x)}{d_j(x)} + v_j \right) - \mathrm{E}(\breve{z}_i)\mathrm{E}(\breve{z}_j) \right) \\
&= \mathrm{P}\{\eta_i = 1, \eta_j = 1\} \left[\mathrm{E}\left(f_i f_j \frac{h_i(x)}{d_i(x)} \frac{h_j(x)}{d_j(x)} \right) - \mathrm{E}(\breve{z}_i)\mathrm{E}(\breve{z}_j) + \mathrm{E}(v_i v_j) \right] \\
&= \mathrm{P}\{\eta_i = 1, \eta_j = 1\} \left[f_i f_j \mathrm{E}\left(\frac{h_i(x)h_j(x)}{d_i(x)d_j(x)} \right) - \mathrm{E}(\breve{z}_i)\mathrm{E}(\breve{z}_j) + \left\{ \begin{array}{ll} \sigma^2_{v_i}, & i = j \\ 0, & i \neq j \end{array} \right\} \right].
\end{aligned}$$

D.2. Weak Perspective Model

$\mathrm{E}(\tilde{z}_i)$ is found by

$$\begin{aligned}
\mathrm{E}(\tilde{z}_i) &= \mathrm{P}\{\eta_i = 1\}(a_i^T \mu_x + \mathrm{E}(\tilde{v}_i)) + \mathrm{P}\{\eta_i = 0\}\mathrm{E}(\tilde{z}_i) \\
\mathrm{P}\{\eta_i = 1\}\mathrm{E}(\tilde{z}_i) &= \mathrm{P}\{\eta_i = 1\}a_i^T \mu_x \implies E(\tilde{z}_i) = a_i^T \mu_x.
\end{aligned}$$

The elements of the covariance matrices are computed by

$$\begin{aligned}
\Sigma_{Zx}(i, :) &= \mathrm{E}\left((\tilde{z}_i - \mathrm{E}(\tilde{z}_i))\tilde{x}^T \right) = \mathrm{E}(\tilde{z}_i\tilde{x}^T) = \mathrm{P}\{\eta_i = 1\}\mathrm{E}[(a_i^T(\tilde{x} + \mu_x) + \tilde{v}_i)\tilde{x}^T] \\
&= \mathrm{P}\{\eta_i = 1\}a_i^T \mathrm{E}(\tilde{x}\tilde{x}^T) = \mathrm{P}\{\eta_i = 1\}a_i^T \Sigma_x.
\end{aligned}$$

$$\begin{aligned}
\Sigma_Z(i, j) \; &= \; \mathrm{E}(\tilde{z}_i\tilde{z}_j) - \mathrm{E}(\tilde{z}_i)\mathrm{E}(\tilde{z}_j) \\
&= \; \mathrm{P}\{\eta_i = 1, \eta_j = 1\}\mathrm{E}(\tilde{z}_i\tilde{z}_j|\eta_i = 1, \eta_j = 1) \\
&\quad + \mathrm{P}\{\eta_i = 1, \eta_j = 0\}\mathrm{E}(\tilde{z}_i\tilde{z}_j|\eta_i = 1, \eta_j = 0) \\
&\quad + \mathrm{P}\{\eta_i = 0, \eta_j = 1\}\mathrm{E}(\tilde{z}_i\tilde{z}_j|\eta_i = 0, \eta_j = 1) \\
&\quad + \mathrm{P}\{\eta_i = 0, \eta_j = 0\}\mathrm{E}(\tilde{z}_i\tilde{z}_j|\eta_i = 0, \eta_j = 0) - \mathrm{E}(\tilde{z}_i)\mathrm{E}(\tilde{z}_j) \\
&= \; \mathrm{P}\{\eta_i = 1, \eta_j = 1\}\mathrm{E}((a_i^T(\tilde{x} + \mu_x) + v_i)(a_j^T(\tilde{x} + \mu_x) + v_j) - \mathrm{E}(\tilde{z}_i)\mathrm{E}(\tilde{z}_j)) \\
&= \; \mathrm{P}\{\eta_i = 1, \eta_j = 1\} \left[a_i^T \Sigma_x a_j + \left\{ \begin{array}{ll} \sigma^2_{\tilde{v}_i}, & i = j \\ 0, & i \neq j \end{array} \right\} \right].
\end{aligned}$$

ACKNOWLEDGMENTS

The authors wish to thank the anonymous reviewers for their feedback, Dr. John S. Wenstrand, Prof. Hamid Aghajan, and Dr. Danny Yang for fruitful discussions, and Prof. John T. Gill III and Kyle Heath for their help with the experimental setup.

REFERENCES

Bar-Shalom, Y., Li, X. R., and Kirubarajan, T. 2001. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons Inc., New York, NY.

Bettstetter, C., Hartenstein, H., and Pérez-Costa, X. 2002. Stochastic properties of the random waypoint mobility model: Epoch length, direction distribution, and cell change rate. In *Proceedings of the International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems*. 7–14.

Bhanu, B., Ravishankar, C., Roy-Chowdhury, A., Aghajan, H., and Terzopoulos, D. 2011. *Distributed Video Sensor Networks*. Springer-Verlag New York Inc.

Caron, F., Davy, M., Duflos, E., and Vanheeghe, P. 2007. Particle filtering for multisensor data fusion with switching observation models: Application to land vehicle positioning. *IEEE Trans. Signal Process. 55,* 6, 2703–2719.

Chu, M., Haussecker, H., and Zhao, F. 2002. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *Int. J. High Perform. Comput. Appl. 16,* 3.

DEL BLANCO, C. R., MOHEDANO, R., GARCIA, N., SALGADO, L., AND JAUREGUIZAR, F. 2008. Color-based 3d particle filtering for robust tracking in heterogeneous environments. In *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*. 1–10.

DOCKSTANDER, S. L. AND TEKALP, A. M. 2001. Multiple camera tracking of interacting and occluded human motion. *Proc. IEEE 89,* 10, 1441–1455.

DOUCET, A., VO, B.-N., ANDRIEU, C., AND DAVY, M. 2002. Particle filtering for multi-target tracking and sensor management. In *Proceedings of the 5th International Conference on Information Fusion*. Vol. 1, 474–481 vol.1.

ERCAN, A. O., EL GAMAL, A., AND GUIBAS, L. J. 2006. Camera network node selection for target localization in the presence of occlusions. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys) Workshop on Distributed Smart Cameras*.

ERCAN, A. O., EL GAMAL, A., AND GUIBAS, L. J. 2007. Object tracking in the presence of occlusions via a camera network. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*. 509–518.

ERCAN, A. O., YANG, D. B.-R., EL GAMAL, A., AND GUIBAS, L. J. 2006. Optimal placement and selection of camera network nodes for target localization. In *Proceedings of the Inernational Conference on Distributed Computing in Sensor Systems*.

ERTIN, E., FISHER III, J. W., AND POTTER, L. C. 2003. Maximum mutual information principle for dynamic sensor query problems. In *Proceedings of the Inernational Conference on Information Processing in Sensor Networks (IPSN)*.

FUNIAK, S., GUESTRIN, C., PASKIN, M., AND SUKTHANKAR, R. 2006. Distributed localization of networked cameras. In *Proceedings of the Inernational Conference on Information Processing in Sensor Networks (IPSN)*.

HEINZELMAN, W. B., CANDRAKASAN, A. P., AND BALAKRISHNAN, H. 2002. An application specific protocol architecture for wireless microsensor networks. *IEEE Trans. Wirel. Commun. 1,* 4, 660–670.

ISLER, V. AND BAJCSY, R. 2005. The sensor selection problem for bounded uncertainty sensing models. In *Proceedings of the Inernational Conference on Information Processing in Sensor Networks (IPSN)*. 151–158.

KAILATH, T., SAYED, A. H., AND HASSIBI, B. 1999. *Linear Estimation*. Prentice Hall, Upper Saddle River, NJ.

KHAN, S., JAVED, O., RASHEED, Z., AND SHAH, M. 2001. Human tracking in multiple cameras. In *Proceedings of the Inernational Conference on International Conference on Computer Vision (ICCV)*.

KIM, W., MECHITOV, K., CHOI, J.-Y., AND HAM, S. 2005. On target tracking with binary proximity sensors. In *Proceedings of the Inernational Conference on Information Processing in Sensor Networks (IPSN)*.

LI, D., WONG, K. D., HU, Y. H., AND SAYEED, A. M. 2002. Detection, classification and tracking of targets. *IEEE Signal Process. Mag.*, 17–29.

NIU, R., ZUO, L., MAŞAZADE, E., AND VARSHNEY, P. 2011. Conditional posterior Cramér–Rao lower bound and its applications in adaptive sensor management. *Distrib. Video Sen. Netw.*, 303–317.

OTSUKA, K. AND MUKAWA, N. 2004. Multiview occlusion analysis for tracking densely populated objects based on 2-d visual angles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

PAHALAWATTA, P. V., DEPALOV, D., PAPPAS, T. N., AND KATSAGGELOS, A. K. 2003. Detection, classification, and collaborative tracking of multiple targets using video sensors. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*. 529–544.

RISTIC, B., ARULAMPALAM, S., AND GORDON, N. 2004. *Beyond the Kalman Filter, Particle Filters for Tracking Applications*. Artech House, Norwood, MA.

SANKARANARAYANAN, A., CHELLAPPA, R., AND BARANIUK, R. 2011. Distributed sensing and processing for multi-camera networks. *Distrib. Video Sens. Netw.* 85–101.

SANKARANARAYANAN, A. C., VEERARAGHAVAN, A., AND CHELLAPPA, R. 2008. Object detection, tracking and recognition for multiple smart cameras. *Proc. IEEE 96,* 10, 1606–1624.

SHENG, X. AND HU, Y.-H. 2005. Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks. *IEEE Trans. Signal Process. 53,* 1, 44–53.

SHI, J. AND TOMASI, C. 1994. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 593 –600.

SHRIVASTAVA, N., MUDUMBAI, R., AND MADHOW, U. 2006. Target tracking with binary proximity sensors: Fundamental limits, minimal descriptions, and algorithms. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*.

TAYLOR, C., RAHIMI, A., BACHRACH, J., SHROBE, H., AND GRUE, A. 2006. Simultaneous localization, calibration and tracking in an ad-hoc sensor network. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*.

TESSENS, L., MORBEE, M., LEE, H., PHILIPS, W., AND AGHAJAN, H. 2008. Principal view determination for camera selection in distributed smart camera networks. In *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*. 1–10.

TRUCCO, E. AND VERRI, A. 1998. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, Upper Saddle River, NJ.

UTSUMI, A., MORI, H., OHYA, J., AND YACHIDA, M. 1998. Multiple-view-based tracking of multiple humans. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*.

VAZQUEZ, P.-P., FEIXAS, M., SBERT, M., AND HEIDRICH, W. 2001. Viewpoint selection using viewpoint entropy. In *Proceedings of the Vision Modeling and Visualization Conference*.

VIHOLA, M. 2007. Rao-blackwellised particle filtering in random set multitarget tracking. *IEEE Trans. Aerospace Electron. Sys. 43,* 2, 689–705.

WANG, H., YAO, K., POTTIE, G., AND ESTRIN, D. 2004. Entropy-based sensor selection heuristic for localization. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*.

WONG, L., DUMONT, C., AND ABIDI, M. 1999. Next best view system in a 3d object modeling task. In *Proceedings of the International Symposiume on Computational Intelligence in Robotics and Automation*.

YANG, D., GONZALEZ-BANOS, H., AND GUIBAS, L. 2003. Counting people in crowds with a real-time network of simple image sensors. In *Proceedings of the 9th IEEE International Conference on Computer Vision*. 122–129.

YANG, D. B. 2005. Counting and localizing targets with a camera network. Ph.D. dissertation, on Stanford University.

YANG, D. B.-R., SHIN, J.-W., ERCAN, A. O., AND GUIBAS, L. J. 2004. Sensor tasking for occupancy reasoning in a network of cameras. In *Proceedings of the 1^{st} Workshop on Broadband Advanced Sensor Networks Workshop (BaseNets)*.

YANG, M.-H., KRIEGMAN, D., AND AHUJA, N. 2002. Detecting faces in images: A survey. *IEEE Trans. Pattern Anal. Machine Intell. 24,* 1, 34–58.

YILMAZ, A., LI, X., AND SHAH, M. 2004. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Trans. Pattern Anal. Machine Intell. 26,* 11, 1531–1536.

ZAJDEL, W., CEMGIL, A. T., AND KROSE, B. J. A. 2004. Online multicamera tracking with a switching state-space model. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*.

ZHAO, F. AND GUIBAS, L. 2004. *Wireless Sensor Networks*. Elsevier Inc., Amsterdam.

ZHAO, W., CHELLAPPA, R., PHILLIPS, P. J., AND ROSENFELD, A. 2003. Face recognition: A literature survey. *ACM Comput. Surv. 35,* 4, 399–458.

ZUO, L., NIU, R., AND VARSHNEY, P. 2011. Conditional posterior Cramér Rao lower bounds for nonlinear sequential Bayesian estimation. *IEEE Trans. Signal Process. 59,* 1, 1 –14.

Received July 2009; revised March 2011; accepted August 2011