Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling

A Decision-Theoretic Approach to Dynamic Sensor Selection in Camera Networks

Matthijs T. J. Spaan and **Pedro U. Lima**
Institute for Systems and Robotics
Instituto Superior Técnico
Av. Rovisco Pais, 1, 1049-001
Lisbon, Portugal

Abstract

Nowadays many urban areas have been equipped with networks of surveillance cameras, which can be used for automatic localization and tracking of people. However, given the large resource demands of imaging sensors in terms of bandwidth and computing power, processing the image streams of all cameras simultaneously might not be feasible. In this paper, we consider the problem of dynamical sensor selection based on user-defined objectives, such as maximizing coverage or improved localization uncertainty. We propose a decision-theoretic approach modeled as a POMDP, which selects k sensors to consider in the next time frame, incorporating all observations made in the past. We show how, by changing the POMDP's reward function, we can change the system's behavior in a straightforward manner, fulfilling the user's chosen objective. We successfully apply our techniques to a network of 10 cameras.

Introduction

In present-day society, large numbers of urban areas have been equipped with networks of surveillance cameras, in the interests of safety and security. These include outdoors (e.g., university campuses, parking lots) as well as indoors (e.g., office buildings, shopping malls) locations. Besides the classical surveillance setup in which a human operator monitors video streams, we can also automate this task, alleviating human operators of tedious work, which eventually leads to tiredness and lack of attention. A large body of literature has been developed on automatic localization and tracking of people, or the detection of several human behaviors of interest.

However, given the large resource demands of imaging sensors in terms of bandwidth and computing power, processing the video streams of a large number of cameras simultaneously might not be feasible. For instance, state-of-the-art human activity recognition algorithms require high-resolution video streams at a high frame rate, as well as significant computational resources. The bandwidth problem becomes even more prominent given the growing popularity of so-called IP-cameras, which are connected directly to a (local area) network, and need to share this medium. Also, when a human operator is monitoring a large number of camera streams, only displaying a small number of them will reduce the cognitive load on the operator. Camera selection schemes along the lines we present in this paper can be used to alleviate such cognitive limitations.

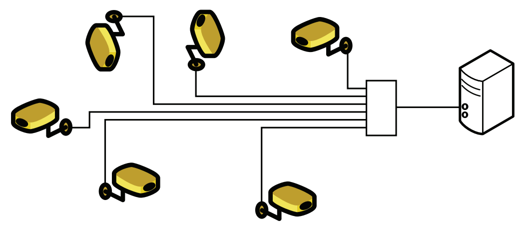


Figure 1: Illustration of our sensor selection problem. Images can only be processed on one server, and the cameras have to share the limited network bandwidth.

Given these resource constraints and a set of sensors, we study the problem of selecting a subset of sensors that can be active at any point in time. The goal of the system is to optimize a user-defined objective. We will consider several possible objectives, for instance maximizing coverage or minimizing uncertainty when tracking people. Related problems have been studied in the wireless sensor network literature (Rowaihy et al. 2007), where the resource constraint considered typically is energy, given each sensor's limited battery life. We focus on developing dynamic sensor selection methods, which can change the active subset of sensors over time. In this way, the system can react to the observed state of its environment, significantly improving the system's performance. Figure 1 illustrates our scenario, where a shared network medium forms a bottleneck for images to be transported from each camera to a server for analysis.

In this paper, we present a decision-theoretic approach to dynamic sensor selection. In particular, we propose to use Partially Observable Markov Decision Processes (POMDPs) (Kaelbling, Littman, and Cassandra 1998), as they form a strong methodological framework for sequential decision-making under uncertainty. We model the problem of tracking a person using n cameras as a POMDP, under the constraint that only k cameras can emit observations at any point in time, with $k \ll n$. This resource constraint forms a general way to model restrictions in available bandwidth or computing power. Now, the POMDP's actions are defined as selecting the k sensors to be active in the next time frame. As the POMDP's belief state forms a sufficient statistic for the decision-making problem, the system incorporates all observations made in the past. We show how, by changing the POMDP's reward function, we can change the system's behavior in a straightforward manner, fulfilling the user's chosen objective. We demonstrate our techniques on a network of 10 cameras, illustrating the rich set of behaviors we can achieve.

Background

We will give a brief introduction to POMDPs, followed by a short overview of the sensor selection literature.

POMDPs

A POMDP models the interaction of an agent with a stochastic and partially observable environment, and it provides a rich mathematical framework for acting optimally in such environments (Kaelbling, Littman, and Cassandra 1998).

A POMDP assumes that at any time step the environment is in a state $s \in S$, the agent takes an action $a \in A$ and receives a reward $r(s, a)$ from the environment as a result of this action, while the environment switches to a new state s' according to a known stochastic transition model $p(s'|s, a)$. The agent's task is defined by the reward it receives at each time step and its goal is to maximize its long-term reward. After transitioning to a new state, the agent perceives an observation $o \in O$, that may be conditional on its action, which provides information about the state s' through a known stochastic observation model $p(o|s', a)$.

Given the transition and observation model the POMDP can be transformed to a belief-state MDP: the agent summarizes all information about its past using a belief vector $b(s)$. The belief b is a probability distribution over S, which forms a Markovian signal for the planning task. The initial state of the system is drawn from the initial belief b_0, which is typically part of a POMDP's problem definition. Every time the agent takes an action a and observes o, its belief is updated by Bayes' rule; for the discrete case:

$$b_a^o(s') = \frac{p(o|s', a)}{p(o|a, b)} \sum_{s \in S} p(s'|s, a) b(s), \tag{1}$$

where $p(o|a, b) = \sum_{s' \in S} p(o|s', a) \sum_{s \in S} p(s'|s, a) b(s)$ is a normalizing constant.

In POMDP literature, a plan is called a policy $\pi(b)$ and maps beliefs to actions. As solving POMDPs optimally is very hard, we will consider approximate algorithms. Recent years have seen much progress in approximate POMDP solving which we can leverage.

Sensor selection

Sensor networks typically require sensor selection algorithms, so as to limit the number of active sensors at a given time step to those essential to achieve the network task(s), thus minimizing the cost of their operation, usually corresponding to the energy expenditure. Sensor selection problems can be defined on two layers: on the layer of the physical network infrastructure and on the application layer, which is a higher level related to the particular application of the sensor network.

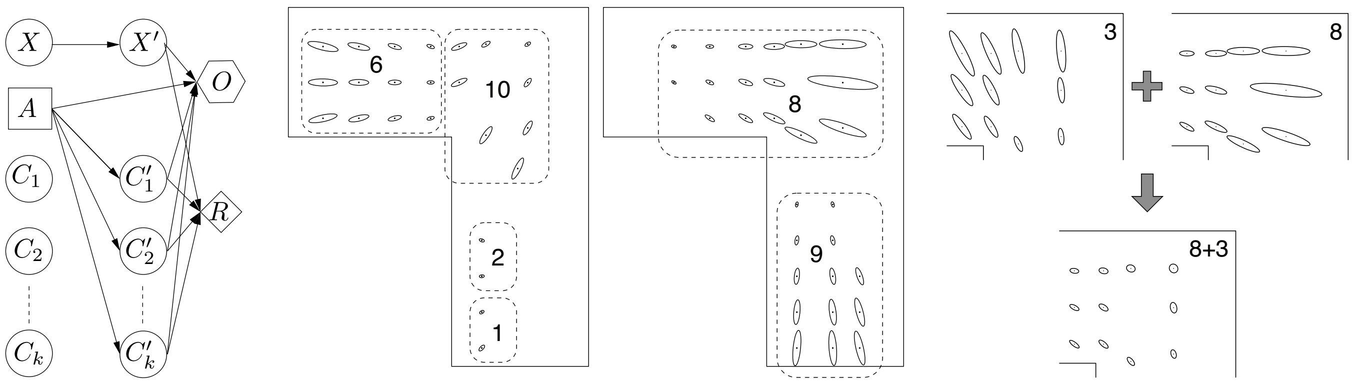
When considering the sensor network infrastructure, the purpose of basic sensor selection problems concerns complete coverage (i.e., ensuring that every point in a given region is within the sensing range of at least one sensor) and target tracking and localization (Rowaihy et al. 2007). In the application layer, one may distinguish the purpose of sensor selection as to whether they refer to single task or multiple task problems. Examples of single-task sensor selection problem are area coverage or person tracking and localization. Changing the focus of attention to high-priority events, such as detecting a person entering the room, while keeping the task of tracking other people already inside the room is an example of a multiple-task problem for sensor networks.

The sensor selection problem can be defined as determining the best subset of k sensors from the complete set of n networked sensors, such that the overall utility is maximized, while the overall cost is less than a given budget. In this context, utility is frequently identified with accuracy, and cost with the energy spent to activate and operate the sensors. These are typical features for the infrastructure layer of the network. However, when one considers the application layer, utility may have very different meanings, such as minimizing energy consumption in buildings, maximizing the reliability of high-priority event detection (Spaan 2008) or maximizing assistance to people. Even at the infrastructure layer, utility may refer to other attributes, such as information timeliness or security. When mobile sensors (e.g., mounted on mobile robots) are involved, cost should include the energy spent moving the sensor, but it could also refer to the uncertainty about the sensor location, which in turn affects the uncertainty of the target location in a global frame.

Solutions for the sensor selection problem range from finding heuristics to approximately solve coverage NP-Complete optimization problems, to target tracking and localization minimum entropy, maximum information gain or minimum mean-squared error solutions, see (Rowaihy et al. 2007) and references therein. Entropy-based solutions are widely used. However, one must be careful with implementing the concept of reducing the entropy as a means of increasing accuracy, as this is only true in unbiased sensors, where accuracy is similar to precision.

Dynamic sensor selection using POMDPs

In this paper, we propose a POMDP-based approach to the sensor selection problem. As POMDPs model dynamical systems with only partial observability of the system state, they form a natural framework for tackling the sensor selection problem. Their generality allows the system designer to express more intricate objectives than many of the approaches mentioned above. POMDP-based approaches have been proposed before for sensor management problems, for an overview see (Williams 2007; Hero et al. 2008), but often they only compute myopic solutions (Kreucher, Kastella, and Hero 2005) or simulation-based open-loop feedback



(a) POMDP model. (b) Fields of view for different cameras. (c) Fusion.

Figure 2: (a) 2DBN representation of our POMDP model. Circles indicate state variables, the square the action variable, the hexagon the observation variable, and the diamond the reward. The variables in the left column refer to the current timestep, and in the right column to the following time step. (b) The average covariance matrices for several cameras. (c) Detail of the uncertainty models of cameras 3, 8, and their fusion.

controllers (Singh et al. 2007). On the other hand, we propose a closed-loop non-myopic controller, exploiting recent advances in approximate POMDP solving.

Problem definition

Similar to the literature on wireless sensor networks, e.g., (Pahalawatta, Pappas, and Katsaggelos 2004), we define our problem as selecting a subset of k active sensors out of a total of n deployed sensors. In our case, these sensors are cameras, and the k active cameras are allowed to process image data and transmit resulting measured events (such as a person detection). In this way, we abstract constraints on bandwidth and processing power to a choice of k: the number of cameras the system can handle at any point in time.

In contrast to some of the techniques in the sensor network literature, we consider dynamic sensor selection schemes, in which the subset of active cameras can be changed over time. As is common in the POMDP literature, we discretize time and take decisions at predefined intervals. At each decision moment, the sensor selection scheme determines the k cameras to be active. These cameras will send the image frames they observe to a server for processing. Events are extracted from the image data, and are fed back to the selection algorithm. This way, it can react to the observed events, allowing for better performance than a static configuration of k cameras can provide.

We consider the problem of selecting sensors given a set of deployed sensors, in contrast to approaches which consider the problem of optimal placement of sensors, e.g., (Krause, Singh, and Guestrin 2008). Given the already installed base of surveillance cameras and the relative burden of deploying cameras (compared to wireless sensor nodes), we focus on the sensor selection scenario. Also note that the current scenario could extended to include more complicated network infrastructures, or to add local processing power to cameras, resulting in hierarchical setups (Sridharan, Wyatt, and Dearden 2008).

Models for decision-theoretic sensor selection

In order to take advantage of (context-specific) independence between several state variables, we model our POMDP as a two-stage dynamic Bayesian network (2DBN). This allows us to use approximate POMDP solvers that specifically exploit independence (Poupart 2005), and scale up to larger problems. Although our methods and models are quite general, we will consider a specific scenario, namely tracking a single person by a network of ceiling-mounted cameras.

States and transitions

Figure 2a shows a graphical representation of our POMDP model. We need to model the location of the person as well as which k cameras are active. The latter is modeled by k variables $C_1 \dots C_k$, each having as possible values $\{\mathit{off}, 1, 2, \dots, n\}$, where n is the total number of deployed cameras. Note that the *off* value is only used to define the initial state of the system (to avoid a bias), but there are no actions that switch cameras off (this could be an interesting extension, assuming a cost for each active camera).

The location of the person is represented by variable X. When we want to track multiple people or other types of targets, we should include a variable for each of them. As most POMDP theory assumes discrete state spaces, we discretize the person's possible positions in a set of nodes. POMDP techniques have been developed for continuous settings, but require certain model assumptions, for instance that they are represented using (mixtures of) Gaussians (Porta et al. 2006; Brunskill et al. 2008). In this paper we consider the discrete case for simplicity. However, as we are only considering which image data streams to feed to a data fusion algorithm (operating in a continuous metric space), the resulting fused person localization estimates will be continuous.

As shown in the 2DBN (Figure 2a), X is assumed independent of the other variables, as the person's path is not influenced by the state of the camera network. Given the lack of knowledge about the person's intended path, we assume a random motion pattern, in which the person can either stay in its current node, or move to neighboring ones, given an assumed maximum velocity. Such a representation allows us to model movement constraints posed by the environment (e.g., corridors, walls, or other obstacles), which constrain a person's possible paths, and facilitate more informed transition models. Note that we also could learn the transition model, as for instance proposed in (Gilbert and Bowden 2008), where probabilistic transition models between spatially separated cameras are learned in an online, unsupervised way.

The values of the camera variables $C_1 \ldots C_k$ are deterministic, and only depend on the last action (and not, for instance, on their assignment in the previous time step). The action space is simply the $\binom{n}{k}$ possible assignments of n cameras to k variables (where order does not matter and each camera can only be assigned once). When $\binom{n}{k}$ is too large to efficiently compute a POMDP policy, one could limit the size of the action set by only allowing a certain predefined number of camera variables to change at a particular time step (e.g., half the cameras at odd time steps, and the other half at even time steps). Another option would be to prune actions a priori, for instance, when considering tracking a single person we could focus the action set on actions that assign cameras with overlapping fields of view.

Observation model

The observation space O has the same size as the state space X, where each observation $o \in O$ indicates an observation of a person close to a corresponding $x \in X$. Note that this modeling is independent of which camera made the observation. The observation model considers two kinds of errors: false positives and false negatives for detecting a person, and given a detection, the error in the true position of the detected target.

First, in order to be able to make an informed choice about which cameras to activate given a belief over the person's location, we need to know the field of view of each camera. A reliable way of creating such an observation model is by learning it from the sensor data itself. We create a training set by letting a person move around the environment, and record each measurement made by a particular camera. Next, for each measurement, we find the closest node $x \in X$. When a node x has measurements from camera c assigned to it, this effectively means that camera c is capable of detecting people close to node x. In this case, we assign a high probability 0.8 that the correct observation o will be made when the person is close to x and c is one of the activated cameras. Also, we assign a low probability to false positives, accounting for noise and inaccuracies due to the high level of abstraction.

The above procedure tells us which camera can observe which node, but does not provide information about the expected performance of a camera. To be able to decide which subset of cameras is best suited to observe a particular node, we need a model of the sensor's uncertainty. Therefore, we perform an empirical analysis of the object detection algorithm used to detect people, namely the adaptive background subtraction method proposed in (Boult et al. 2001). The location of detected objects has an error in the image plane that we assume is normally distributed. This error is estimated by the comparison of a large number of detections with human-provided ground truth. We then extract its standard deviation and we assume the error in x and y direction in the image plane to be uncorrelated. To obtain a detected object's position and error in the metric space, we use the Unscented Transform (Julier, Uhlmann, and Durrant-Whyte 2000) to map its mean position and covariance from the image plane to the ground plane. Although the normal distribution will be distorted by the homography, we fit a Gaussian to it and thus obtain the covariance matrix in the metric space. More details can be found in (Barbosa et al. 2009).

Using this procedure, each person detection is defined as a mean position with a covariance matrix. Now, given the previously detailed training set, we not only determine which node x can be observed by camera c, we also average the covariance matrices of measurements assigned to a particular node. This gives us an estimate of the uncertainty we can expect a camera to deliver should it detect a person at a particular node. Figure 2b plots the resulting average covariances for several cameras in our lab. As one would expect, the error increases when the node is further away from the camera, but not as much in the angular direction. When considering more than 1 camera to observe a particular node, we fuse the covariance matrices involved using a simple weighted linear combination. Figure 2c shows an example: the resulting model for cameras 3 and 8 after fusing camera 3 with camera 8.

Encoding objectives

A key benefit of using a POMDP approach to the sensor selection problem is that it is relatively straightforward to implement and balance different objectives. Currently, we will consider two objectives, coverage and uncertainty. Coverage means that when a person is in the field of view of the camera network, at least one of the k active cameras is observing it. Uncertainty refers to the concept that we would not only like the k cameras to observe the person, we want to choose the subset of k cameras that minimizes the uncertainty regarding the person's true location.

To encode the coverage objective, we define a reward function that assigns a constant positive reward r_{cov} when the person's location x is inside the field of view of one of the k active cameras. This guides the POMDP to try to keep the active cameras covering the nodes it believes the person to be, but without caring about uncertainty. The resulting POMDP controller we will denote by POMDPCOV.

If we care about uncertainty as well as coverage, we can define a different reward function, which also takes into account the expected uncertainty of observing a person at a node by a particular camera. We define a positive reward $r_{unc}(x, \{c\})$ as the determinant of the covariance matrix for a (node,cameras) tuple $(x, \{c\})$, as plotted in Figure 2b, mapped through a sigmoid function for regularization purposes. Intuitively speaking, the smaller the size (determinant) of the covariance matrices, the lower the uncertainty, and the higher our reward will be. The resulting POMDP

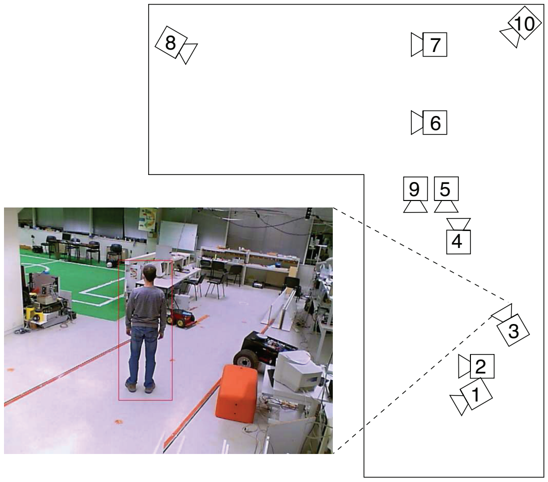


Figure 3: The location of each of the 10 ceiling-mounted cameras in our lab, and an example image from camera 3.

controller is called POMDPUNC.

Note that we define a standard POMDP reward model over states, instead of opting for reward models defined over beliefs instead of states, i.e., $r(b, a)$, as for instance in (Krishnamurthy and Djonin 2007). In the latter case we could define a reward model based on the belief entropy, and a natural interpretation would be to give higher reward to low-entropy beliefs. However, a reward model defined over beliefs does significantly raise the complexity of planning, as the value function will no longer be piecewise linear and convex. Such a compact representation is being exploited by many optimal and approximate POMDP solvers.

Experiments

To evaluate our methods, we tested them in our lab on a camera network consisting of 10 ceiling-mounted cameras. More details on the experimental setup are provided in (Barbosa et al. 2009).

Experimental setup

Each camera in the network has a framerate of 30 fps, and a resolution of 640×480 pixels. The location of each camera as well as an example image from one of them is shown in Figure 3. The location and direction indicated in this plot is not sufficient to tell the field of view of each camera: for instance, although cameras 5 and 9 appear similar, camera 5 is oriented downwards, resulting in a small field of view (but with low uncertainty).

As mentioned before, we use an adaptive background subtraction method (Boult et al. 2001) (complimented by outlier removal) to detect people in image streams. The event detections of the k active cameras are processed by a Bayesian data fusion algorithm, which computes a weighted linear combination of all measurements received in each fusion interval of $\Delta_f = 0.1s$ (Hackett and Shah 1990). For every such interval, we record whether any events were detected by the active cameras, and if so, we record the fused data. We can define quantitative performance criteria based on this data. In particular, we will compare selection schemes on their coverage as well as uncertainty. Let $T = \{t_1, t_2, ..., t_m\}$ be the total set of m fusion intervals that we consider. First we find the subset T_{any} of T in which any of the total of n cameras has detected a person, to establish a baseline. No selection scheme can ever detect people in intervals $T \setminus T_{any}$. Hence, given the set of fusion intervals T_π in which any of the k cameras chosen by a sensor selection scheme π observed events, we define coverage as

$$C_{cov}(T_\pi) = \frac{|T_\pi|}{|T_{any}|},$$

i.e., the ratio between the number of intervals an event was detected and the total number of intervals in which an event could have been detected. Note that we only consider those intervals in which T_{any} has a detection, in which case false positives do not count towards C_{cov}. Subsequently, we define the average uncertainty $C_{unc}(T_\pi)$ as the mean of the determinant of the covariances matrices of the fused data in T_π.

To wrap up our POMDP model, we manually chose a set X of 38 nodes that cover the free space in the environment. Initially, all cameras are switched off, and the initial belief over X is set to a uniform distribution, reflecting the lack of a priori knowledge about the person's location. The POMDP controllers were computed using SYMBOLIC PERSEUS (Poupart 2005) with a discount rate $\gamma = 0.95$. SYMBOLIC PERSEUS is an approximate point-based POMDP solver, based on PERSEUS (Spaan and Vlassis 2005), but instead of using a flat POMDP representation, it exploits an Algebraic Decision Diagram (ADD) representation to tackle large factored POMDPs. Point-based POMDP methods have become popular in recent years, for instance one advantage is that we can influence their run time by varying the size of the belief set that they operate on. Solving the larger models takes in the order of a few hours, with belief sets of 500 beliefs sampled using a Q_{MDP} heuristic.

We recorded a test set of $293s$ in which a person walks around the environment. The test set contains events from all $n = 10$ cameras, on which we compared our POMDP-based sensor selection schemes POMDPCOV and POMDPUNC with MDP-based schemes, and with a heuristic one, HEUR. The latter selects the k cameras closest to the last measurement, where the distance is defined to the mean of the field of view of each camera. When no measurements have been received for $1s$, HEUR assumes it lost the tracking of the person, and keeps on selecting k cameras uniformly at random until the person is observed again. We also compared with the well-known Q_{MDP} method (Littman, Cassandra, and Kaelbling 1995), a simple approximation technique that treats the POMDP as if it were fully observable and solves the MDP, e.g., using value iteration. The resulting $Q(s, a)$ values are used to define a control policy by $\pi(b) = \arg\max_a \sum_s b(s)Q(s, a)$. QMDPCOV uses the same reward function as POMDPCOV, and QMDPUNC the one of POMDPUNC. Finally, we tested all static assignments of cameras (STATIC), and report on the best one (in

$k = 1$	C_{cov}	C_{unc}
POMDPCOV	0.821	0.00575
POMDPUNC	0.696	0.00190
HEUR	0.513	0.00659
QMDPCOV	0.367	0.00512
STATIC(9)	0.352	0.00371
QMDPUNC	0.092	0.00310
$k = 2$		
POMDPUNC	0.929	0.00436
POMDPCOV	0.860	0.00653
QMDPCOV	0.836	0.00737
HEUR	0.775	0.00669
QMDPUNC	0.726	0.00498
STATIC(3,9)	0.685	0.0122

Table 1: Quantitative results for the test set.

	Interval A		Interval B	
$k = 1$	C_{cov}	C_{unc}	C_{cov}	C_{unc}
POMDPCOV	0.799	$2.43e{-3}$	0.750	$7.89e{-3}$
POMDPUNC	0.799	$6.71e{-4}$	0.460	$2.00e{-3}$
HEUR	0.462	$9.91e{-4}$	0.390	$1.67e{-2}$
$k = 2$				
POMDPCOV	0.920	$4.53e{-3}$	0.690	$2.27e{-2}$
POMDPUNC	0.889	$2.42e{-3}$	0.720	$3.44e{-3}$
HEUR	0.698	$1.75e{-3}$	0.660	$1.82e{-2}$

Table 2: Quantitative results for intervals A and B.

terms of C_{cov}). Each scheme determined a new subset of k cameras every $\Delta_d = 1s$.

Results

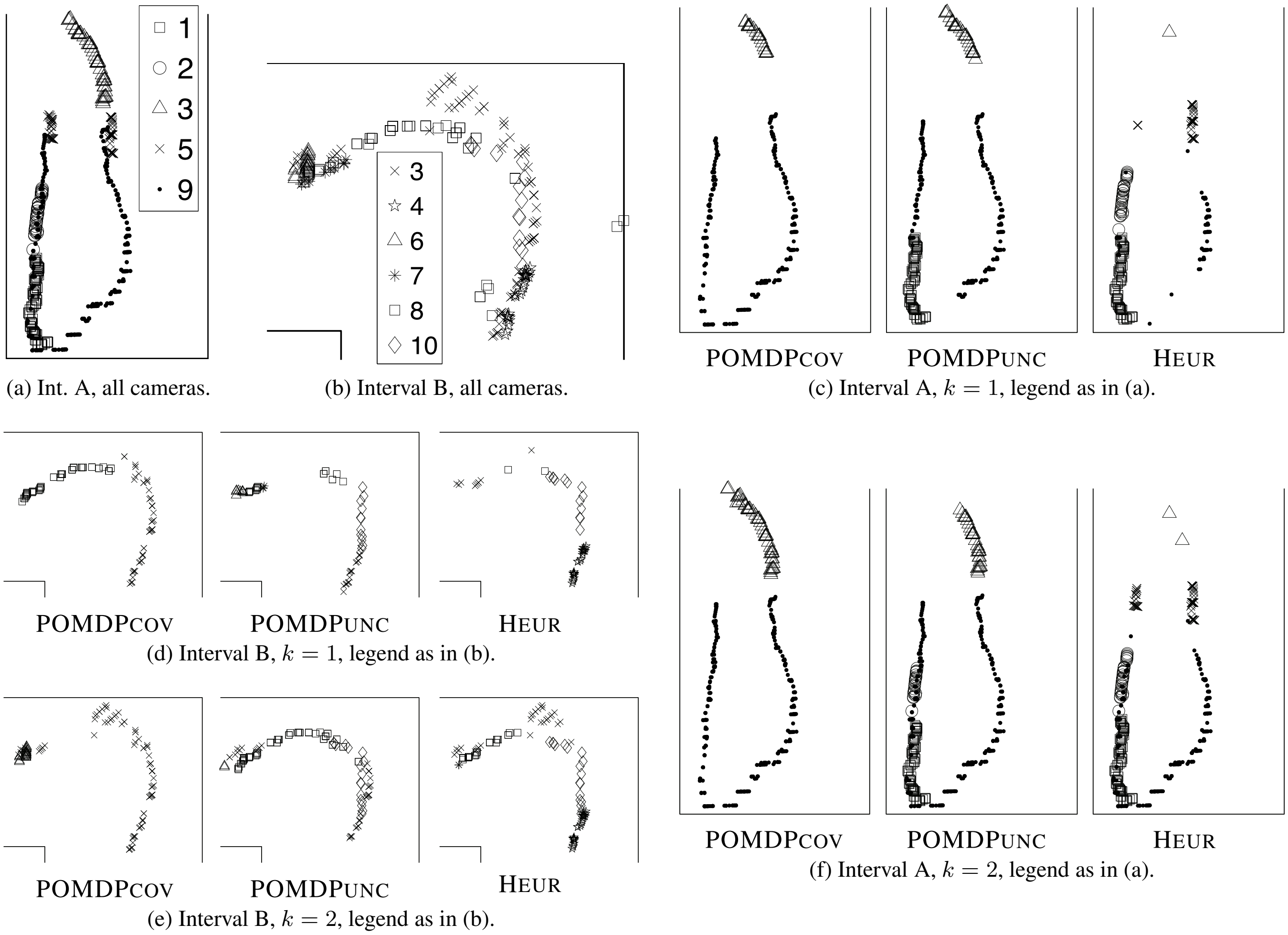
Table 1 shows the values of C_{cov} and C_{unc} for $k = 1$ and $k = 2$. We can see both POMDP methods outperform the baselines significantly in terms of coverage. For $k = 1$, POMDPUNC sacrifices coverage for low uncertainty, switching to cameras that provide low-uncertainty measurements but have only a small field of view (more details below). The Q_{MDP} based controllers perform poorly in general (with the exception of QMDPCOV for $k = 2$), which can be attributed to their heuristic nature. Note that C_{unc} only refers to intervals in which some events were observed, i.e., a low C_{unc} by itself does not indicate good performance. As an example (and the extreme case), STATIC(2) has a C_{unc} of $7e{-5}$, but only $C_{cov} = 0.013$: camera 2 has a very small field of view, and in the few cases it observes a person, it has very low uncertainty (Figure 2b). For $k = 2$, POMDPUNC obtained a lower C_{unc}, but also a higher C_{cov} than POMDPCOV. The latter can be explained by the fact that for $k > 1$ the POMDPCOV policy will try to maximize coverage by choosing only non-overlapping cameras. On this test set however, given that only one person was present, this leads to slightly less coverage than POMDPUNC.

In Figure 4 we provide a detailed exposition of the effect of different sensor selection schemes, by focusing on two time intervals in our test set. During interval A ($20s$) the person passes through the field of view of cameras 1, 2, 3, 5, and 9, and in interval B the person passes through a corner of the lab in $10s$, observed by cameras 3, 4, 6, 7, 8, and 10. The raw data from each camera are shown in Figure 4a and 4b for interval A resp. B. In general, any gaps in Figure 4 occur when cameras have been selected that do not observe the target at that moment.

Looking at Figure 4c, we see that POMDPCOV always selects camera 9 when the person is in camera 9's FOV, i.e., in the bottom part of the environment. POMDPUNC, on the other hand, switches to camera 1 when it believes the person to be in camera 1's FOV (bottom left corner). Camera 1 only has a small FOV, but low uncertainty, which is why POMDPUNC switches to it, but not POMDPCOV. If we only care about coverage, it is safer to select a camera with a large FOV. On the other hand, if we care about uncertainty as well, we should consider switching to camera with a small FOV but low expected uncertainty (cf. the small ellipses in Figure 2b), at the cost of potentially losing coverage. Table 2 shows that indeed the average uncertainty of POMDPUNC is much lower than the one of POMDPCOV. HEUR (Figure 4c) switches to other cameras beyond those selected by POMDPCOV and POMDPUNC, resulting in poor performance. As it does not keep a belief over the person's state, it cannot trade off properly the FOVs of different cameras: Table 2 shows that it has poor coverage compared to POMDPCOV and POMDPUNC. If we set $k = 2$, Figure 4f shows similar results as for $k = 1$. POMDPCOV selects camera 9 and a camera that covers the rest of the environment (not shown), while POMDPUNC selects camera 9 and one of the low-uncertainty cameras, 1 and 2. For interval B, we can see similar patterns: POMDPUNC prefers camera 10 when the person is believed to be near the top-right corner, while POMDPCOV stays with camera 3. The latter has a larger FOV but is further away from the corner, resulting in higher uncertainty.

Discussion and Conclusions

The results we presented in this paper form the basis for more extensive research into POMDPs for sensor selection problems. For instance, currently we assume no cost for switching between cameras, but adding such a cost would be a trivial extension. Also, we can encode more objectives in the POMDP's reward function. For instance, some areas in our environment might be of higher interest than others, and increasing the reward in the high-interest areas will cause the POMDP controllers to pay more attention to them. Furthermore, we can extend the current work to consider more than one person, or to consider different types of events. In these cases, a POMDP approach will provide flexibility to prioritize between events, for instance trading off detecting a possible fire with tracking a person. The former would have a high priority, and the POMDP's reward function can be easily devised in such a way that even a low likelihood of fire will result in surveillance resources being dedicated to it. To explore the scalability of POMDP approaches, larger empirical studies with different application scenarios have to be performed, for instance by adding local processing capability to cameras. This would result in hierarchical structure, which is a key point to achieving scalability when apply-



(a) Int. A, all cameras. (b) Interval B, all cameras. (c) Interval A, $k = 1$, legend as in (a).

(d) Interval B, $k = 1$, legend as in (b).

(e) Interval B, $k = 2$, legend as in (b). (f) Interval A, $k = 2$, legend as in (a).

Figure 4: Detailed results for two time intervals, A ($t = 228.0-248.0$) and B ($t = 162.0-172.0$), showing the effect of different sensor selection schemes, POMDPCOV, POMDPUNC, and HEUR. We plot the raw measurements from the camera(s) selected by the different schemes, where the marker type indicates the camera that generated the measurement. Figures (a) and (b) show the data generated by all cameras during the two time intervals. Quantitative performance summaries of the data plotted in figures (c) through (f) can be found in Table 2.

ing POMDPs to real-world problems (Pineau et al. 2003; Sridharan, Wyatt, and Dearden 2008).

To our knowledge not many papers have been published applying closed-loop non-myopic POMDP solutions to sensor selection problems. Although it is known that many related problems can be formulated as POMDPs, the complexity of solving continuous-state POMDPs in closed form has obstructed their solution, leading for instance to open-loop feedback controllers (Singh et al. 2007). In our case, we tackle this problem by discretizing the state space, however, the final output of the system is still a continuous localization estimate. More related to this paper is the work described in (Krishnamurthy and Djonin 2007), who study so-called threshold policies for POMDPs for sensor scheduling. A crucial difference is that they consider reward functions that are not linear in the belief state, for instance based on the entropy of the belief state. In this case, the POMDP is nonstandard, and the optimal value function is no longer piecewise linear and convex. By defining a standard reward function over states, we remain in the standard POMDP setting, for which many results are known and successful approximate algorithms have been developed. Also, we provide experimental results in real-world scenarios, while in (Krishnamurthy and Djonin 2007) only simulations of small domains are provided.

In (Ji, Parr, and Carin 2007) a POMDP formulation of the problem of multiaspect sensing on a single platform is studied. Apart from targeting other types of applications (an unmanned underwater vehicle vs. a network of cameras), a key difference is that we learn an observation model from data, instead of computing it from a physical model (Ji, Parr, and Carin 2007). Learning the model from data is potentially more reliable, as it will take into account limitations of the sensor or the event detection algorithm. In our case, for instance, occlusions were automatically accounted for. Besides the different application domain, a technical difference is that in (Ji, Parr, and Carin 2007) a classification problem is considered, while our work considers several objectives with respect to tracking uncertainty. Note that incorporating classification decisions combined with tracking uncertainty is an obvious and straightforward avenue of future work.

Concluding, we presented a decision-theoretic approach to dynamic sensor selection, with a focus on tracking a person in a network of surveillance cameras using only a limited number of cameras simultaneously. We showed how we can model this problem as a POMDP, and how we can encode objectives such as maximizing coverage or improving localization uncertainty. We successfully implemented our techniques in a person tracking scenario with 10 cameras.

Acknowledgments

This work was supported by the European Project FP6-2005-IST-6-045062-URUS, and by ISR/IST pluriannual funding through the POS_Conhecimento Program that includes FEDER funds. The authors would like to thank all the people involved in developing the ISROBOTNET testbed, in particular Marco Barbosa, Alexandre Bernardino, Dario Figueira, José Gaspar, Plinio Moreno and Abdolkarim Pahliani, as well as Pascal Poupart for releasing the SYMBOLIC PERSEUS implementation.

References

Barbosa, M.; Bernardino, A.; Figueira, D.; Gaspar, J.; Gonçalves, N.; Lima, P. U.; Moreno, P.; Pahliani, A.; Santos-Victor, J.; Spaan, M. T. J.; and Sequeira, J. 2009. ISRobotNet: A testbed for sensor and robot network systems. In *Proc. of International Conference on Intelligent Robots and Systems*. To appear.

Boult, T. E.; Micheals, R. J.; Gao, X.; and Eckmann, M. 2001. Into the woods: Visual surveillance of noncooperative and camouflaged targets in complex outdoor settings. *Proceedings of the IEEE* 89(10):1382–1402.

Brunskill, E.; Kaelbling, L.; Lozano-Perez, T.; and Roy, N. 2008. Continuous-state POMDPs with hybrid dynamics. In *Proc. of the Int. Symposium on Artificial Intelligence and Mathematics*.

Gilbert, A., and Bowden, R. 2008. Incremental, scalable tracking of objects inter camera. *Computer Vision and Image Understanding* 111(43–58).

Hackett, J., and Shah, M. 1990. Multi-sensor fusion: a perspective. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

Hero, A.; Castañón, D.; Cochran, D.; and Kastella, K., eds. 2008. *Foundations and Applications of Sensor Management*. Springer.

Ji, S.; Parr, R.; and Carin, L. 2007. Non-myopic multi-aspect sensing with partially observable Markov decision processes. *IEEE Trans. Signal Processing* 55(6):2720–2730.

Julier, S.; Uhlmann, J.; and Durrant-Whyte, H. F. 2000. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control* 45(3):477–482.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101:99–134.

Krause, A.; Singh, A.; and Guestrin, C. 2008. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research* 9:235–284.

Kreucher, C.; Kastella, K.; and Hero, A. 2005. Sensor management using an active sensing approach. *Signal Processing* 85(3):607–624.

Krishnamurthy, V., and Djonin, D. 2007. Structured threshold policies for dynamic sensor scheduling–a partially observed Markov decision process approach. *IEEE Transactions on Signal Processing* 55(10):4938–4957.

Littman, M. L.; Cassandra, A. R.; and Kaelbling, L. P. 1995. Learning policies for partially observable environments: Scaling up. In *International Conference on Machine Learning*.

Pahalawatta, P.; Pappas, T.; and Katsaggelos, A. 2004. Optimal sensor selection for video-based target tracking in a wireless sensor network. In *Int. Conf. on Image Processing*.

Pineau, J.; Montemerlo, M.; Pollack, M.; Roy, N.; and Thrun, S. 2003. Towards robotic assistants in nursing homes: Challenges and results. *Robotics and Autonomous Systems* 42(3–4):271–281.

Porta, J. M.; Vlassis, N.; Spaan, M. T. J.; and Poupart, P. 2006. Point-based value iteration for continuous POMDPs. *Journal of Machine Learning Research* 7:2329–2367.

Poupart, P. 2005. *Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes*. Ph.D. Dissertation, University of Toronto.

Rowaihy, H.; Eswaran, S.; Johnson, P.; Brown, T.; Barnoy, A.; Verma, D.; and La Porta, T. F. 2007. A survey of sensor selection schemes in wireless sensor networks. In *SPIE Unattended Ground, Sea, and Air Sensor Technologies and Applications IX*.

Singh, S. S.; Kantas, N.; Vo, B.-N.; Doucet, A.; and Evans, R. J. 2007. Simulation-based optimal sensor scheduling with application to observer trajectory planning. *Automatica* 43(5):817–830.

Spaan, M. T. J., and Vlassis, N. 2005. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research* 24:195–220.

Spaan, M. T. J. 2008. Cooperative active perception using POMDPs. In *AAAI 2008 Workshop on Advancements in POMDP Solvers*.

Sridharan, M.; Wyatt, J.; and Dearden, R. 2008. Hippo: Hierarchical POMDPs for planning information processing and sensing actions on a robot. In *Int. Conf. on Automated Planning and Scheduling*.

Williams, J. L. 2007. *Information theoretic sensor management*. Ph.D. Dissertation, Massachusetts Institute of Technology.