Using POMDPs to Control an Accuracy-Processing Time Trade-off in Video Surveillance

Komal Kapoor
Department of Computer Science and Engineering
University of Minnesota - Twin Cities
Minneapolis, Minnesota 55455, USA
Email: kapoo031@umn.edu

Christopher Amato *
CS & AI Laboratory (CSAIL)
Massachusetts Institute of Technology
Cambridge, Massachusetts, 02139, USA
Email: camato@csail.mit.edu

Nisheeth Srivastava
Department of Computer Science and Engineering
University of Minnesota - Twin Cities
Minneapolis, Minnesota 55455, USA
Email: nsriva@cs.umn.edu

Paul Schrater
Department of Computer Science and Engineering
University of Minnesota - Twin Cities
Minneapolis, Minnesota 55455, USA
Email: schrater@umn.edu

Abstract

With rapid profusion of video data, automated surveillance and intrusion detection is becoming closer to reality. In order to provide timely responses while limiting false alarms, an intrusion detection system must balance resources (e.g., time) and accuracy. In this paper, we show how such a system can be modeled with a partially observable Markov decision process (POMDP), representing possible computer vision filters and their costs in a way that is similar to human vision systems. The POMDP representation can be optimized to produce a dynamic sequence of operations and achieve a trade-off between time and detection quality, taking into account uncertainty in the filter predictions. In a set of experiments on actual video data, we show that our method can both outperform static "expert" models and scale to large dynamic domains. These results suggest that our method could be used in real-world intrusion detection systems.

1 Introduction

Intrusion detection through video surveillance is becoming increasingly important with the availability of inexpensive cameras and monitoring technology. However, the volume of graphic data generated by such systems and the need for real-time performance requires the deployment of automated investigation techniques. Furthermore, the computational complexity of sophisticated scene understanding algorithms renders online processing difficult. For automated surveillance to become practicable with low hardware budgets, it is essential that the average computational cost of the computer vision methods used be reduced.

To solve this problem, we turn to nature, which performs visual perception of extremely diverse stimuli in real-time very satisfactorily. A prominent feature of perception mechanisms in nature is that they are inherently hierarchical. Such a modular approach, in addition to being conducive to specialization, allows natural systems to be adaptive to problems of varying complexity and resource availability. These systems serve as an essential mechanism for saving cost and preserving functionality in varied real-time scenarios. Several prominent artificial vision algorithms attempt to mimic the human visual system at a *cortical* level through a hierarchy of filters (Lowe 2001; Ranzato et al. 2007). We, however, attempt to emulate the hierarchical nature of visual perception at a *cognitive* level (Marr 1982; Kanwisher et al. 1996). We envisage a surveillance system consisting of differentially sophisticated computer vision filters which are explicitly ordered in an adaptive hierarchy sensitive to processing cost requirements.

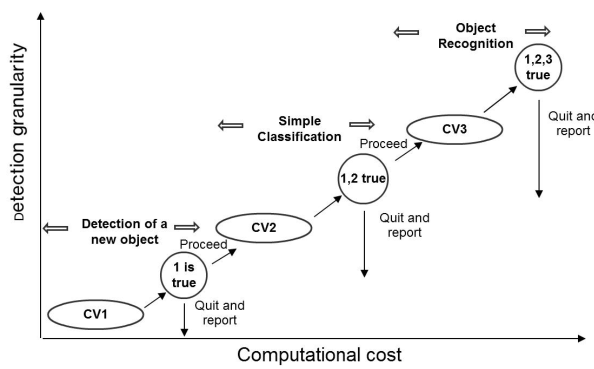

Figure 1: A schematic representation of a hierarchical surveillance system allowing an adaptive trade-off between granularity of object detection and computational cost. CV1, CV2 and CV3 are computer vision filters that are incrementally more informative about particular object identities.

Deriving detection strategies for different real world scenarios manually, perhaps with the help of a domain expert, is not feasible except for very simple problems. Additionally, because of the many available operators that could be utilized in different combinations such that not all are perfect, major fallacies can arise if the accumulated uncertainties are ignored. This is more so applicable for computer vision operators which are notoriously faulty. Nevertheless, by using the structure present in the imagery and combining different

*This work was completed while the author was employed at Aptima, Inc.

Copyright © 2012, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

analysis techniques in an intelligent and dynamic manner the strengths of each technique can be utilized, while minimizing weaknesses.

We express the intrusion detection problem as an instance of a sequential decision-making problem where each operator is characterized by both its accuracy and its cost (see Figure 1). In this work, we implement this in practice using partially observable Markov decision process (POMDP)(Kaelbling, Littman, and Cassandra 1998) models, which optimize the sequential decision-making problem while considering uncertainty about the performance of each computer vision filter in different situations. Such a formulation allows the system to cost-efficiently select appropriate sequences of vision filters to discover potential intrusions in scenarios of varying complexity given resource constraints.

The remainder of this paper is organized as follows. We begin with a summary of related work in Section 2. We then describe the video data that we used in this paper and give an overview of our intrusion detection system in Section 3. Section 4 provides details of the POMDP model utilized for intrusion detection which allows us to dynamically balance the computational cost of video analysis against the operator's accuracy. Section 5 presents our experimental results, showing our model can outperform a traditional "expert" approach using real-world data and scale to large problems. Finally, we present our conclusion in Section 6.

2 Related Work

Automated visual surveillance systems targeted at minimizing human intervention are rapidly gaining importance. As sophisticated computer vision techniques for object tracking and detection with real-time implementations become available, there is significant interest in bringing these techniques together to design integrated scene understanding systems (Collins et al. 2000; Thirde et al. 2006).

Many surveillance systems deal with a sequence of operators being applied to an image. However, little attention has been paid to making such systems adaptive to varying requirements of cost and accuracy. Furthermore, the accumulation of uncertainty has not been addressed so far. There have been earlier attempts (Toth and Kruegel 2002) to make intrusion detection systems cost-sensitive, where costs are defined generally as covering detection cost, manual intervention costs, non-intervention costs etc. For example, (Lee et al. 2002) describe a classification approach using cost-sensitive decision rule sets to implement cost-sensitive intrusion detection. Unfortunately, Lane (Lane 2006) has recently shown how classification approaches insufficiently express the desiderata of intrusion detection systems, since they are typically insensitive to false positive frequencies and the reflexive effects of detecting intrusions. He further shows that POMDPs can be used in a class of network security intrusion detection problems.

POMDPs have been used as cost-optimizing controllers in multiple domains, e.g., for active chemical sensing (Gosangi and Gutierrez-Osuna 2009) and for sensor scheduling (He and Chong 2004). Although POMDPs possess a high computational complexity, several approximate and heuristics based approach have lead to their successful application to real world problems having millions of states (Boger et al. 2005; Young et al. 2010).

Somewhat similar to our approach, Zhang et al. (Zhang and Sridharan 2010) use an hierarchical POMDP model for solving a scene analysis problem in a robot domain, but they do not consider the dynamic case (using video) or learn expected costs of computer vision filters (instead they consider color and shape filters only). They explicitly define a three-layered hierarchy, such that the different layers in the hierarchy address the questions; 'where to look?', 'what to process?' and finally 'how to process?'. In our problem of intrusion detection, instead of using an explicit hierarchical POMDP implementation, we implicitly define an hierarchy of visual operators using insights from human cognition models for object classification and tracking and use a POMDP based controller to adaptively generate detection strategies. The structure of our problem allows us to adopt a factored POMDP formulation, which could be incorporated into other models such as that of Zhang and Sridharan to utilize a wider range of filters and allow dynamic information to be incorporated.

3 Scenario and System

We describe a real-world use case to demonstrate the various merits of our approach. In our problem, we have a single camera continuously monitoring a room. The room has a set of objects forming the static background against which any intrusion is detected. The system maintains a list of objects recognized by it; any intrusion apart from this list of objects raises an alarm. In our scaled-down model, this list of recognized objects (the "whitelist") comprises of a box, and two men named Mike and Jonathan. A database of 1280×720 megapixel videos of the scene is generated for training and testing purposes. Each video includes the view of the scene taken from a Cannon Powershot S80 camera placed on a tripod. In each video, either Mike, Jonathan or the box is moving through the scene. Both forward and backward orientations of Mike and Jonathan are captured separately. This general scenario could be extended to any number of cameras and objects.

Description of the Intrusion Detection System

A real-time intrusion detection system would comprise of two components, an object detection and a tracking system. The detection system locates a new object and defines an appropriate region of interest (ROI), while the tracking system tracks it through multiple frames. The detection system also includes an object classification component that characterizes what the object is. An intrusion is detected on failure to classify the object. We design both these components by adopting a hierarchical approach implicitly through our choice of operators, which consist of computer vision filters and video operations (as discussed below). Through a training phase we learn cost and accuracy values for each operator. We use these values to build a POMDP model which also encodes the desired trade-off between accuracy and resource requirements leading to an optimized solution.

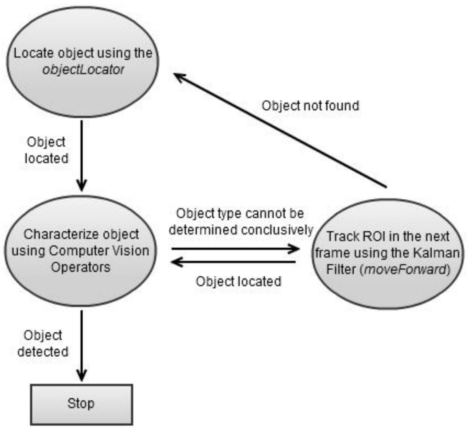

Figure 2: A flow diagram demonstrating how our system intelligently deploys a Kalman Filter tracking mechanism to track an object across multiple frames till it can recognize it with confidence.

Object Detection and Tracking Foveated visual search which fixates on only the relevant parts of an image is an important part of our system for saving cost. In our current implementation, we sample locations where the target is most likely to be found as predicted by a tracking system implemented using an implementation of a Kalman Filter (KF) (Murphy 1998). We use the uncertainty in KF output to determine the ROI size in the next frame with $PredictedSize_x = PreviousSize_x + 2 * sqrt(Var_x)$. Here, Var_x is the variance for the KF prediction. $PredictedSize_y$ can be computed similarly. The operator for this action is called *moveForward*. The described mechanism is represented pictorially in Figure 2.

Object Recognition Component We utilize a number computer vision filters as operators in our system, but many more could be added. The operators in our current implementation are described below. Of the 5 operators, background subtraction, saliency and skin detection are implemented to assign an Interest Score to a ROI, which is then thresholded to obtain a final binary output. The thresholds are learnt during the training phase.

Background Subtraction (backSub) A probabilistic background subtraction technique is used such that a model of the expected background (mean and variance) is learned for each pixel in the image using a few images. The element of surprise (s) for a pixel measures the deviation of the pixel's value from the learned model parameters. The interest score of the ROI is given by: $\frac{count(s(pixel) > threshold)}{count(pixels)}$

Saliency (saliency) A saliency score is assigned to each pixel of an image using a complex visual cognition model of attention (Walther and Koch 2006). We obtain baseline saliency scores for the pixels using a given background image. The Interest Score for the ROI is computed using: $\sum_{pixels} SaliencyScore(image) - SaliencyScore(background)$

Skin Detection (skin) Skin detection works by assigning to each pixel of an image the probability that it represents skin, based on the learnt model of the skin hue (Conaire, O'Connor, and Smeaton 2007). The Interest Score for the ROI is computed using: $count(P(skin) > 0)$

SIFT (SIFT) SIFT features (Lowe 2001) have been popularly used for object detection as they are found to be discriminative under changes in scale, orientation and illumination. In the training stage, SIFT features for a box, Jonathan and Mike are computed and stored in the database using representative images. These features are compared against the features extracted from the test images to determine a match.

Face SIFT (faceSIFT) While the SIFT algorithm would generically search for interesting features in the entire image patch passed to it, most discriminating features in persons are concentrated in their facial features. Therefore, we combine SIFT feature matching with a face detection (Viola and Jones 2004) front-end. SIFT feature matching then proceeds as above, at a smaller computational cost because of the reduced size of the patch and higher accuracy of prediction as superfluous background features are removed.

objectLocator This operator is used at the beginning of a new detection process to orient the Intrusion Detection System to portions of the image that are different than the standard image background. A simple background subtraction technique is used for this purpose. Each pixel is allotted an anomaly score as described for the backSub operator earlier and a binary image is obtained by thresholding these anomaly scores. The intruding object is identified as the largest connected component extracted from the binary image and the center of this object is returned.

4 The POMDP-based Controller

POMDPs are a mathematical framework for sequential decision-making under uncertainty and partial observability. Formally, a POMDP is a tuple $\langle S, A, T, R, \Omega, O, \gamma \rangle$ where S, A and Ω are finite sets of states, actions and observations respectively, $T : S \times A \times S \rightarrow [0, 1]$ denotes the transition model, $R : S \times A \rightarrow \mathbb{R}$ is the reward function, $O : \Omega \times A \times S \rightarrow [0, 1]$ represents the observation model, and γ is the discount factor. The goal is to maximize the expected total cumulative reward starting at an initial state of the system. The resulting solution can be viewed as a controller that chooses actions based on the sequences of observations that are seen.

Our POMDP Model

For our system we define the POMDP tuple as follows:

- $S \in X \times Y \times Size \times Orient \times Type$, where :
  - X : X-coordinate of the ROI enclosing the object
  - Y : Y-coordinate of the ROI enclosing the object
  - $Size$: Size of the ROI enclosing the object
  - $Orient$: $\{F, B\}$, orientation of the object (Forward 'F' or Backward 'B').
  - $Type$: $\{Jon(J), Mike(M), Box(B), Nothing(N)\}$
- $A = A_{vision} \times size \cup A_{move} \cup A_{found}$, where $A_{vision} = \{$*backSub*, *saliency*, *skin*, *SIFT*, *faceSIFT*$\}$ and $A_{move} = \{$*moveForward*$\}$ as discussed in Section 3. A_{vision} can be applied using bounding boxes of different

sizes (ROIs). A_{found} is a set of actions representing the type, orientation, location, size of the object that is found, terminating the problem after it is taken.

- T, an identity matrix is used in our system. This is because the Kalman filter is used for tracking and foveal sampling is used to keep an object centered. These two systems could be combined in the future to allow the POMDP to use and direct tracking in a more systematic way (rather than through observations as discussed below).
- R, with $R(s, a)$ defined differently for vision operators and found actions. When $a \in \{$*backSub*, *saliency*, *skin*, *SIFT*, *faceSIFT*, *moveForward*$\}$, R is the cost of performing the operator a on an ROI of size *size* (defined as the expected time required by that operator on the ROI). When $a \in A_{found}$, R denotes the positive or negative values for correctly or incorrectly identifying the object.
- Ω, the values returned by the different operators, which are based on the Interest Scores or noisy state information as described below.
- O, where the probabilities are learnt for $a \in \{$*backSub*, *saliency*, *skin*, *SIFT*, *faceSIFT*$\}$ as well as Gaussian noise adjusted size and location values for $a \in \{$*moveForward*$\}$. For example, O returns the probability that skin detection will return *true* given the ROI size, object type and orientation. For SIFT there is an observation probability for each object type and for movement operators, x, y and s values are grouped into pairs that cannot be distinguished and Gaussian noise is added.

Training and Policy Construction

The POMDP model was trained using a set of 60 training images with approximately 10 images each of Jon Forward, Mike Forward, Jon Backward, Mike Backward, Box and images with nothing happening. The system can track n possible ROI sizes given by $size = (128{*}i, 72{*}i)$ for $i = 1\, to\, n$. Training ROIs were constructed around the identified object returned using *objectLocator*, for these ROI sizes. Each operator was applied in turn on each of the training ROIs and the obtained output was recorded. The observation function was calculated using the following formula for each vision operator action: $P(Obs = x | State = y, Action = a) = \frac{\text{No of images of State y for which observation } = x}{\text{Total Number of images of state y}}$

The POMDP controller adaptively controls the ROI size on which an action is performed. Increasing the size of the ROI patch reduces loss of information but is susceptible to noise as irrelevant parts of the image get incorporated.For example, *skin* can detect skin when applied to the creme colored wall in the scene. The (negative) reward function for each vision operator was obtained by computing the average cost (i.e., time) of running the operators on the training images for each state.

5 Experiments and Evaluation

We demonstrate the capabilities of our approach through a series of experiments. The experiments were conducted on a test set comprising of 300 images containing 50 images each for JF, JB,MF,MB,B, and N picked in order from our set of videos. The POMDP model was solved using symbolic Perseus (Poupart 2005), which is a fast, approximate method that is able to use the fact the the states and the observations are decomposable into different factors that do not influence each other. Many other optimal or approximate POMDP solution methods could be used instead.

Static Model The first POMDP model we developed, called the Static Model, concentrated on the Object Recognition Module given a ROI. Hence, the *moveForward* filter was omitted and the POMDP was initialized with a ROI centered around the coordinates provided by the *objectLocator* operator for each successive image.

We constructed 3 versions of the POMDP controller each expressing a different trade-off criterion between accuracy and the cost of execution. The different cost-accuracy trade-offs were brought about by appropriately setting the reward functions in the model. The *POMDPUpper* and the *POMDPLower* Models represent the two extremes in the POMDP Model construction such that *POMDPUpper* is obtained when the cost of executing the operators is ignored (such a model maximizes its accuracy) and the *POMDPLower* is obtained when the the model gets the same reward irrespective of its prediction (this model has accuracy corresponding to a random prediction). The *POMDPMid* model represents an intermediate model between the two extremes. The rewards for this model reflect the test criteria used below: -100 is given for choosing the wrong object, 100 is given for choosing the correct object and orientation and 0 is given for choosing the correct object, but incorrect orientation. Note that *POMDPUpper* represents an upper bound not only on a POMDP solution, but also for any method that uses the same learnt data model from our test set.

Dynamic Model The dynamic model integrates the static model with the object tracking module. In this model, the POMDP controller can call the moveForward operator if a reasonable decision cannot be made based on the current frame. The Kalman filter's X and Y mean predictions tell our system where to look for intrusions, whereupon the cost-sensitive controller identifies the correct size patch around the filter's mean location to examine using vision algorithms. For this model, $|X|$, $|Y|$ and $|S|$ were discretized to ten each. Here, the rewards are the same as the static case, except additive rewards are included for choosing locations and size. Additional values of 10 and -10 are given for choosing each the location and size correctly or incorrectly.

Results In Figure 3 we demonstrate how our intrusion detection system functions with different accuracy-cost requirements. The adaptive cost control in our system is provided by the Object Recognition Module and hence we focus on the static model for this experiment. We compare how our POMDP based controllers performed against a solution provided by a human expert, which we call the Expert Model. The expert begins with background subtraction to identify if something is happening in the scene. If something is happening, it subselects the area of the image. Then it performs skin detection which allows it to differentiate between humans and the non-humans. Finally, it either applies SIFTFace if

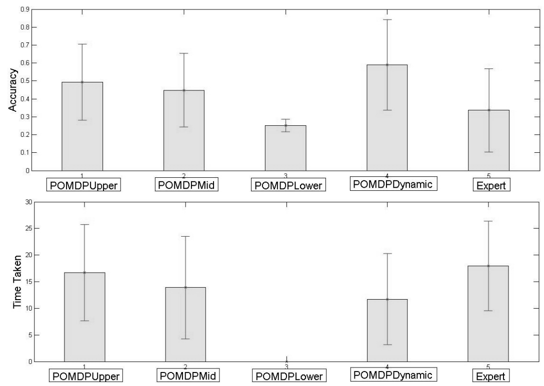

Figure 3: A comparison of accuracy and time for execution of the *POMDPDynamic* model against the *POMDPUpper*, *POMDPMid*, *POMDPLower* and *Expert* models.

skin is detected or else applies SIFT and classifies the object. The saliency operator is omitted as its incorporation is seen to reduce the model's performance. In this experiment, accuracy is obtained by scoring each correct detection as 1 and each incorrect detection as 0. Models are penalized by 0.5 if they get the orientation of Jon and Mike wrong. The cost of execution is the execution time recorded on the same machine as used for training.

The different versions of the POMDP model reflect the ease with which such a formulation can adapt to different cost-accuracy requirements. On average, all POMDP models required less time than the expert model and only *POMDPLower*, which represents a random policy, had a lower accuracy. *POMDPMid* displays a trade-off between time and quality, slightly decreasing accuracy while decreasing time. Note that the data itself is very noisy (as demonstrated by the relatively low accuracy performance of *POMDPUpper*), showing that methods that consider uncertainty, such as POMDPs, are likely to outperform other methods, such as the expert model, in noisy domains.

In Table 1, we show that our intrusion detection system adapts cost of execution to the complexity of the scenario better than the expert and the baseline SIFT model which applies the SIFT operator on the entire image each time. People (Jonathan and Mike), an inanimate object (Box) and 'Nothing' comprise three levels of complexity in our construed scenario. Since SIFT cannot detect the orientation of the people in the image, the accuracy values in the figures is calculated such that correct detection is scored 1 point irrespective of orientation. The controller (*POMDPMid*) takes considerably less time to process a box and 'no change' frames than frames with humans present, while producing much better accuracy than the other models.

It can also be seen that the action sequences for the different categories of objects reflect that the POMDP formulation optimizes the inherent hierarchy in the operators and effectively utilizes it for different scenarios. For example, a typical action sequence for an image having Jonathan facing backward is backSub(yes)→ Skin(yes)→ SIFTFace(unknown)→SIFT(Jonathan) where

	People	Objects	Nothing
POMDPMid(accuracy=0.533)	18.494	4.367	4.981
Expert(accuracy=0.365)	22.844	6.635	9.719
SIFT(accuracy=0.289)	30.372	19.848	15.464

Table 1: Average time taken (in seconds) for people, objects and nothing using the static, expert and SIFT Baseline model

	$\lvert S\rvert$	$\lvert A\rvert$	$\lvert\Omega\rvert$	Time
Static	1,024	57	8	37
Dynamic 1	10,240	67	40	140
Dynamic 2	102,400	157	200	512
Dynamic 3	1,024,000	167	1,000	3674

Table 2: POMDP problem sizes and running time (in seconds with 500 belief points and alpha vectors)

values in parentheses denote the results on applying the operators. On the other hand, for 'Nothing' the controller's action sequence applies backSub(no) followed by SIFT(Nothing). The low accuracy numbers obtained from SIFT reflect the inherent complexity of the problem and the limitations of the detection operators used. The strength of our methodology is in the ability to make the most of the given operators and combine them efficiently over time.

Finally the performance of the dynamic model is compared against the previous models. Here, to make the comparison fair the same evaluation criteria was used as in the static case, without incorporating object size or location into the accuracy score. As shown in Figure 3, the dynamic model outperforms all the previous models in both accuracy and computational cost. By employing an efficient detection-tracking-detection mechanism it is able to quickly gather more information about the object providing a more reliable classification. Furthermore, the ability of the model to skip uninformative frames allows it to to be more cost effective. In the dynamic case, the POMDP policy is very similar to *POMDPMid*, except moveForward is called additional times to determine additional size and location information.

The scalability of the POMDP solution is shown in Table 2. Different versions of the dynamic model were created and the sizes of these, along with the time taken to solve them using factored Perseus is given. The full dynamic model is called Dynamic 3 here while cases that do not include x, y location and s are called Dynamic 1 and 2 respectively.

6 Conclusion

In this work, we design an intrusion detection system using a hierarchical approach to scene understanding. We also successfully integrate a POMDP formulation for object recognition that balances the computational cost of image analysis against the classification accuracy. We demonstrate through experimentation on actual video data that the POMDP outperforms baseline models made from an expert system and SIFT. This shows that a POMDP model that produces an optimized sequential solution that is adaptive based on the information it receives can produce better results than standard static models. The results also show that our approach is robust to noisy data by producing values near the upper bound by retaining uncertainty estimates and finding ways to reduce that uncertainty. The POMDP model is able to scale to large problems with high uncertainty, producing the best solution for a problem with over 1 million states and 100 observations.

Our approach is also very extensible. As computer vision algorithms improve or other systems are developed, they can be incorporated by learning their expected time and accuracy parameters and including them as actions in our POMDP model. Furthermore, it has been shown that visual search based on human eye tracking can be modeled using a POMDP (Butko and Movellan 2008). This idea can also be incorporated into our model, changing our POMDP transition model to retain estimates of an object's location.

More generally, the trade-off between processing cost and prediction accuracy is natural across a variety of epistemological settings, many of which could benefit from adopting an investigative approach similar to ours. In particular, scenarios where the decision space can be naturally factored and the decision process structured as a decision tree of operator/action sequences can easily benefit from our approach. For example, in health and medicine, such a methodology can advance solutions for cost efficient analysis (Lubell et al. 2008). In climate studies, copious satellite data must be analyzed to continuously track climate changes. Some examples include monitoring tropical deforestation (Tucker and Townshend 2000), snow cover (Romanov, Gutman, and Csiszar 2010) etc. A quick and efficient screening procedure, along the lines we suggest, can be used for identifying the most promising areas for further tracking while maintaining performance guarantees. Precisely the same conditions hold in mining astronomical data for interesting stellar events (Feroz and Hobson 2008). Our hierarchical approach to information exploration could be profitably employed in many such situations.

Acknowledgments

We gratefully acknowledge our sponsors, APTIMA via Contract No. USDOD NOOOI4-I 1-M-0038.

References

Boger, J.; Poupart, P.; Hoey, J.; Boutilier, C.; Fernie, G.; and Mihailidis, A. 2005. A decision-theoretic approach to task assistance for persons with dementia. In *International Joint Conference on Artificial Intelligence*, volume 19, 1293.

Butko, N., and Movellan, J. 2008. I-POMDP: An infomax model of eye movement. In *Development and Learning, 2008. ICDL 2008. 7th IEEE International Conference on*, 139–144.

Collins, R.; Lipton, A.; Kanade, T.; Fujiyoshi, H.; Duggins, D.; Tsin, Y.; Tolliver, D.; Enomoto, N.; and Hasegawa, O. 2000. A system for video surveillance and monitoring. Technical Report CMU-RI-TR-00-12, Carnegie Mellon University.

Conaire, C.; O'Connor, N.; and Smeaton, A. 2007. Detector adaptation by maximising agreement between independent data sources. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1–6.

Feroz, F., and Hobson, M. 2008. Multimodal nested sampling: an efficient and robust alternative to markov chain monte carlo methods for astronomical data analyses. *Monthly Notices of the Royal Astronomical Society* 384(2):449–463.

Gosangi, R., and Gutierrez-Osuna, R. 2009. Active Chemical Sensing With Partially Observable Markov Decision Processes. In *American Institute of Physics Conference Series*, volume 1137, 562–565.

He, Y., and Chong, K. 2004. Sensor scheduling for target tracking in sensor networks. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 1, 743 –748.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101(1-2):99 – 134.

Kanwisher, N.; Chun, M. M.; McDermott, J.; and Ledden, P. J. 1996. Functional imaging of human visual recognition. *Cognitive Brain Research* 5(1-2):55 – 67.

Lane, T. 2006. A decision-theoretic, semi-supervised model for intrusion detection. In Maloof, M.; Jain, L.; and Wu, X., eds., *Machine Learning and Data Mining for Computer Security*. 157–177.

Lee, W.; Fan, W.; Miller, M.; Stolfo, S.; and Zadok, E. 2002. Toward cost-sensitive modeling for intrusion detection and response. *Journal of Computer Security* 10:5–22.

Lowe, D. 2001. Local feature view clustering for 3d object recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*.

Lubell, Y.; Reyburn, H.; Mbakilwa, H.; Mwangi, R.; Chonya, S.; Whitty, C.; and Mills, A. 2008. The impact of response to the results of diagnostic tests for malaria: cost-benefit analysis. *British Medical Journal* 336(7637):202–205.

Marr, D. 1982. *Vision: A computational approach*. Freeman & Co.

Murphy, K. 1998. Kalman filter toolbox for matlab.

Poupart, P. 2005. *Exploiting structure to efficiently solve large scale partially observable Markov decision processes*. Ph.D. Dissertation.

Ranzato, M.; Huang, F.; Boureau, Y.; and LeCun, Y. 2007. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1–8.

Romanov, P.; Gutman, G.; and Csiszar, I. 2010. Automated monitoring of snow cover over north america with multispectral satellite data. *Journal of Applied Meteorology and Climatology* 39:1866–1880.

Thirde, D.; Borg, M.; Ferryman, J.; Fusier, F.; Valentin, V.; Bremond, F.; and Thonnat, M. 2006. A real-time scene understanding system for airport apron monitoring. In *Proceedings of the Fourth IEEE International Conference on Computer Vision Systems*.

Toth, T., and Kruegel, C. 2002. Evaluating the impact of automated intrusion response mechanisms. In *Computer Security Applications Conference, 2002. Proceedings. 18th Annual*, 301 – 310.

Tucker, C., and Townshend, J. 2000. Strategies for monitoring tropical deforestation using satellite data. *International Journal of Remote Sensing* 21(6-7):1461–1471.

Viola, P., and Jones, M. J. 2004. Robust real-time face detection. *International Journal of Computer Vision* 57:137–154.

Walther, D., and Koch, C. 2006. Modeling attention to salient proto-objects. *Neural Networks* 19(9):1395–1407.

Young, S.; Gasic, M.; Keizer, S.; Mairesse, F.; Schatzmann, J.; Thomson, B.; and Yu, K. 2010. The hidden information state model: a practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language* 24(2):150–174.

Zhang, S., and Sridharan, M. 2010. Vision-based scene analysis on a mobile robot using layered POMDPs. In *International POMDP Practitioners Workshop, Toronto, Canada*.