Efficient Resource Allocation for Sparse Multiple Object Tracking

Rui Figueiredo[1], João Avelino[1], Atabak Dehban[1], Alexandre Bernardino[1], Pedro Lima[1], Helder Araújo[2]

[1]*Institute for Systems and Robotics, Instituto Superior Técnico*

[2]*Institute for Systems and Robotics, Universidade de Coimbra*

{*ruifigueiredo, joao.avelino, atabak, alex, pal*}*@isr.tecnico.ulisboa.pt, helder@isr.uc.pt*

Keywords: Active Sensing, Constrained Resource Allocation, Multiple Object Tracking

Abstract: In this work we address the multiple person tracking problem with resource constraints, which plays a fundamental role in the deployment of efficient mobile robots for real-time applications involved in Human Robot Interaction. We pose the multiple target tracking as a selective attention problem in which the perceptual agent tries to optimize the overall expected tracking accuracy. More specifically, we propose a resource constrained Partially Observable Markov Decision Process (POMDP) formulation that allows for real-time on-line planning. Using a transition model, we predict the true state from the current belief for a finite-horizon, and take actions to maximize future expected belief-dependent rewards. These rewards are based on the anticipated observation qualities, which are provided by an observation model that accounts for detection errors due to the discrete nature of a state-of-the-art pedestrian detector. Finally, a Monte Carlo Tree Search method is employed to solve the planning problem in real-time. The experiments show that directing the attentional focci to relevant image sub-regions allows for large detection speed-ups and improvements on tracking precision.

I. Introduction

Developing efficient adaptive sensing systems that are capable of dealing with computational and power limitations as well as timing requirements is of the utmost importance in a wide range of fields, including automatic surveillance (Sommerlade and Reid, 2010), sports analysis (Wang and Parameswaran, 2004) and human-robot interaction (HRI) (Mihaylova et al., 2002).

In multiple object tracking with resource constraints scenarios, the observer's goal is to predict the best regions in the visual field to attend, in the quest to evaluate if they pertain to a given set of persons of interest, and thus to prune the visual search space by filtering out irrelevant image locations. Current state-of-the art object detection algorithms are based on exhaustive search, sliding window approaches, which are typically inefficient and agnostic to top-down temporal context.

In this work, we propose a probabilistic framework which poses the multiple object tracking-by-detection problem as an on-line, resource constrained decision making, aimed at minimizing the combined targets' state uncertainty, while coping with computational processing limitations (see Figure 1). More specifically, we pose our decision framework within the *Partially Observable Markov Decision Processes* (POMDPs) domain in order to account for non-deterministic dynamics and partially observable states. The derived dynamic resource allocation decision process combines prior knowledge about the targets' state dynamics with accumulated probabilistic information provided from sequentially gathered observations, in order to optimize multiple target location estimation precision (i.e. minimize tracking uncertainty). In the proposed formulation, actions are taken from a low dimensional binary space. This allows for finding decision policies in real-time using on-line, tree-based, planning algorithms for finite horizon POMDPs (Ross et al., 2008). Our framework relies on object detections with associated confidence measures, obtained from visual information, that are used to drive the observer's attentional focus during multiple object tracking.

Our main contributions are the following. First, we model the state-dependent uncertainty that arises during detection due to the discrete nature of the sliding window based detector. Then, we apply an online Monte Carlo Tree Search method to solve the planning problem in real-time. The computational benefits of our methodology are demonstrated in a multiple person tracking scenario, by combining it with a state-of-the-art pedestrian detection algorithm (Dollár et al., 2014). Moreover, we note that the proposed decision making pipeline can be combined with any general object detection algorithm. All the methodologies have been implemented in C++ to make them suitable for video surveillance or real-time applications involving robotic platforms provided with vision.

The remainder of this paper is structured as follows. In section II we overview some related work available in the literature. In section III we describe the various components involved in the proposed adaptive tracking pipeline. In section IV we assess the proposed methodology performance by evaluating the balance between efficiency (low computational requirements) and effectiveness in multiple object tracking task-execution. Finally, in section V we wrap up with some conclusions and future work.

II. Background

Adaptive sensing is a trendy topic in many areas including computer vision (Gedikli et al., 2007), robotics (Spaan et al., 2015) and neuroscience (Van Rooij, 2008). Like biological systems, artificial systems are equipped with limited computational and energetic resources, and thus, modeling and replicating the observed mechanisms of selective attention in humans, is of primordial importance to develop more efficient and robust strategies for visual tasks.

Current state-of-the art object detectors are based on expensive binary classifiers which typically operate over the full image space, in a sliding window manner. When combined with fast bottom-up saliency-based approaches that generate object bounding box proposals, the overall detection process becomes more efficient (Zitnick and Dollár, 2014), since regions that are unlikely to contain objects are discarded for further processing. However, these approaches are agnostic to object dynamics, and are solely based on low-level visual features.

Resource-constrained adaptive sensing, is within a different line of research, and accounts for dynamical uncertain environments and noisy sensors for sequential decision making. The temporal integration of continuously gathered noisy detections is used to predict future environment states and decide, in a top-down manner, where to allocate the limited sensing resources, according to some task-related goal. It has been shown that adaptive sensing improves not only processing efficiency but also estimation robustness when compared to non-adaptive approaches (Malloy and Nowak, 2014).

Adaptive sensing problems can be formulated as POMDPs (Ahmad and Yu, 2013)(Butko and Movellan, 2010)(Chong et al., 2008) that, depending on the way they compute the policies, belong to two different paradigms: Offline methods compute full policies before run time. Despite achieving remarkable performance in visual search tasks, these often require the evaluation of many possible situations, via backward induction, and hence take a considerable amount of time (e.g. hours). Online decision approaches avoid the computational burden of computing full policies for many situations, by departing from the current belief state and simulating future rewards for a finite planning horizon (Ross et al., 2008).

Within the online POMDP domain, the work closest to ours is the one in (Chong et al., 2008), which proposed a formulation for general adaptive sensing problems. The authors applied rollout techniques which are guaranteed to improve upon a provided base policy, that may be hard or impossible to compute. Rollout techniques evaluate the candidate actions, by running many Monte-Carlo simulations and returning the action with the best average outcome.

In this work we rely on a different, widely known algorithm named Monte Carlo Tree Search (MCTS) (Browne et al., 2012), which has recently been given much attention by the Artificial Intelligence community due to its outstanding performance in the game Go (Gelly et al., 2012). MCTS combines tree search with randomized rolllout simulations, being ideal for decision making under uncertainty. To our knowledge we are the first to apply an online tree-based POMDP solver in a stochastic resource-constrained multiple object tracking scenario.

III. Adaptive Sensing: Probabilistic Multiple Person Tracking under Resource Constraints

A POMDP for general active sensing can be defined as a 6-element tuple $(\mathcal{X}, \mathcal{A}, \mathcal{Y}, T, O, R)$ where

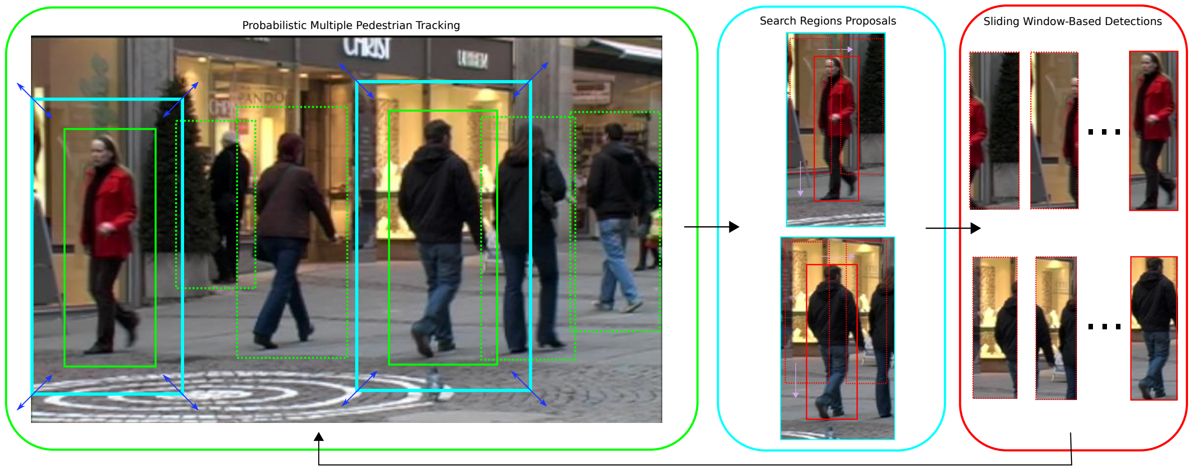
Probabilistic Multiple Pedestrian Tracking
Search Regions Proposals
Sliding Window-Based Detections

Fig. 1: The proposed resource-constrained multiple pedestrian tracking pipeline. Given a set of persons being tracked, our decision making algorithm decides which sub-regions of the visual scene to attend. Then, a sliding window-based detector is applied to the selected search regions, instead of the whole image. For each region a winning candidate is obtained via maximum suppression and fed to the associated tracker with probabilistic measures queried from the observation model.

$\mathcal{X}$, $\mathcal{A}$ and $\mathcal{Y}$ denote the set of the possible environment states, perceptual actions and observations, respectively. State transitions are modeled as a Markov process and represented by the probability distribution function (pdf) $T(x_t, x_{t-1}) = p(x_t|x_{t-1})$. Observations are generated from states according to the pdf $O(x_t, a_t, y_t) = p(y_t|x_t, a_t)$.

Under the resource-constrained adaptive sensing domain, the goal of the planning agent is to devise control strategies that generate perceptual actions from belief states, such that some intrinsic cumulative reward is maximized, while accounting for perceptual limitations. In the rest of this section we describe our resource-constrained POMDP formulation for multiple pedestrian tracking scenarios.

Let us consider a set of targets indexed by $\mathcal{K} = \{1, ..., K\}$, being tracked in a 2D image plane $\mathcal{I}$, with state $x^k \in \mathcal{X} \subset \mathbb{R}^3$ given by

$$x_t^k = \begin{bmatrix} x_t^{k,c} \\ x_t^{k,s} \end{bmatrix} \tag{1}$$

where $x^{k,c} = (u, v)$ and $x^{k,s}$ represent the bounding box centroid image coordinates and scale, respectively. Moreover, let us assume a stationary Markov chain $p(x_t^k|x_{t-1}^k)$ in order to model the object's state transition between consecutive frames. Similarly to (Bewley et al., 2016) we assume sparsity-in-space and independence among targets, and a linear constant-velocity dynamics model, which is a good approximation for targets that move with low acceleration in 3D and are not too close to the image plane. Finally, we assume that the targets' states are partially observable and statistically explained by the observation model distribution $p(y_t^k|x_t^k)$.

A. Recursive Bayesian Estimation

Object tracking can be achieved by means of recursive Bayesian estimation, according to

$$\begin{aligned} b_t^k &\overset{\text{def}}{=} p(x_t^k|y_{1:t}^k) \\ &= \eta p(y_t^k|x_t^k)\bar{b}_t^k \end{aligned} \tag{2}$$

where b_t^k represents the *belief* posterior probability over the target state x_t^k, given the set of all gathered observations $y_{1:t}^k$ taken up to time t, η is a normalizing factor and

$$\bar{b}_t^k = \int p(x_t^k|x_{t-1}^k) b_{t-1}^k dx_{t-1} \tag{3}$$

represents the belief after the prediction step. Furthermore, we assume Gaussian state transition and observation noises and hence tracking is optimally performed using K independent Kalman filters. At each time instant, each Kalman filter provides a parametric posterior probability distribution function (pdf) over the target state

$$b_t^k = \mathcal{N}(\hat{x}_t^k, \Sigma_t^k) \tag{4}$$

where

$$\hat{x}_t^k = \begin{bmatrix} \hat{x}_t^{k,c} \\ \hat{x}_t^{k,s} \end{bmatrix} \tag{5}$$

is the estimated state and

$$\Sigma_t^k = \begin{bmatrix} \sigma_t^{k,c} & 0 \\ 0 & \sigma_t^{k,s} \end{bmatrix} \tag{6}$$

is the error covariance matrix. Note that here we consider a diagonal covariance matrix and aggregate the centroid components in order to ease the notation.

B. Observation Model

The observations provided by the object detector are localized bounding boxes, obtained with a pedestrian detection algorithm. More specifically, at each time instant the agent collects a set of observations

$$\mathcal{Y}_t = \left\{ y_t^k, k = 1, ..., K \right\} \tag{7}$$

each corresponding to a noisy projection of the k target state.

Detection noise has several origins, the easiest to model being the one originated by the discrete nature of the detector. The noise affecting the center of a bounding box $\varepsilon_{x_t^{k,c}}$ has two origins, both depending on the scale of the bounding boxes: ε_{sl}, the error due to the sliding window process and ε_{sc_i}, the error due to the uncertainty of the size of the bounding box. The value of sliding window jumps Q_{sl} depends on the scale of the detection:

$$Q_{sl}^n = s^n Q_{sl}(0) \tag{8}$$

where Q_{sl}^n is the number of pixels between two consecutive sliding window positions at scale n and s^n is the value of scale n, defined as:

$$s^n = 2^{\frac{n}{N}} \tag{9}$$

where N is the number of scales per octave. The present implementation of the detector has $N = 8$.

The value of the jumps of the bounding box center-bottom due to scale change, depend on the scale. The number of pixels is given by Q_{scx} and Q_{scy}, for the x and y coordinates, respectively. In the worst case scenario, a jump from the actual scale to the coarsest one, these values are given by:

$$Q_{scx}^n = w^0 (2^{\frac{n+1}{8}} - 2^{\frac{n}{8}})/2 \tag{10}$$

$$Q_{scy}^n = h^0 (2^{\frac{n+1}{8}} - 2^{\frac{n}{8}})/2 \tag{11}$$

where w^0 and h^0 are the width and the height of the smallest bounding box ($n = 0$).

Assuming a Gaussian distribution for these quantization errors, the statistics of ε_{sl} are given by

$$\mu_{sl}^n = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \mathbf{\Sigma}_{sl}^n = \begin{bmatrix} (Q_{sl}^n)^2 & 0 \\ 0 & (Q_{sl}^n)^2 \end{bmatrix} \tag{12}$$

Regarding ε_{sc_i}, we approximate the statistics of these errors by the worst case which is given by

$$\mu_{sc}^n, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \mathbf{\Sigma}_{sc}^n \approx \begin{bmatrix} (Q_{scx}^n)^2 & 0 \\ 0 & (Q_{scy}^n)^2 \end{bmatrix} \tag{13}$$

Since both sources of noise are independent but not additive, our observation model considers the largest one at each time. This yields the final image observation error ε^n:

$$\varepsilon^n \sim \mathcal{N}(\mathbf{0}, \Sigma^n) \tag{14}$$

where

$$\Sigma^n = \max(\Sigma_{sl}^n, \Sigma_{sc}^n) \tag{15}$$

C. Dynamic Search Regions

Let us now consider different time-varying (dynamic) regions of interest (i.e. bounding boxes) to be attended, each delimiting a target instance hypothesis

$$\mathbf{u}_t = \bigcup_{k \in \mathcal{K}} u_t^k \quad \text{where} \quad u_t^k \subset \mathcal{X} \tag{16}$$

Search regions are deterministically and analytically determined from beliefs according to the following mapping function

$$f : \hat{x}_t^k, \Sigma_t^k \rightarrow u_t^k \tag{17}$$

which is defined as follows

$$u_t^k = \left[\hat{x}_t^{k,c} - \alpha_c \sigma_t^{k,c}, \hat{x}_t^{k,c} + \alpha_c \sigma_t^{k,c} \right] \times \tag{18}$$

$$\left[\hat{x}_t^{k,s} - \alpha_s \sigma_t^{k,s}, \hat{x}_t^{k,s} + \alpha_s \sigma_t^{k,s} \right] \tag{19}$$

where α_s and α_c are user selected parameters that control the width of the confidence bounds and thus the size of the search regions. This definition accounts for the confidence level of the true target state being within the search region. The user selected parameters permit balancing the trade-off between accuracy and allocation effort (larger vs smaller regions).

Furthermore, we assume that each region has a deterministic, time-varying binary activation state

$$\mathcal{A} = \{ a^k \in \mathbb{B}, k \in \mathcal{K} \} = \mathbb{B}^K \tag{20}$$

where $\mathbb{B} = \{0, 1\}$ with 0 and 1 meaning ”not processing” and ”processing”, respectively. Decision making is therefore performed in a finite multi-dimensional binary action space and involves selecting which sub regions of the image space to apply the sliding window detector to perform measurement update steps. The belief becomes dependent on actions as follows

$$b_t^k(a_t^k) = \begin{cases} \bar{b}_t^k & \text{if } a_t^k = 0 \\ \eta p(y_t^k|x_t^k)\bar{b}_t^k & \text{if } a_t^k = 1 \end{cases} \tag{21}$$

where η is a normalizing constant. For attended regions, the predicted belief is approximated by the expected expected observation uncertainty given by the observation model, over a finite set of space points corresponding to detection windows $\mathcal{Y}^k \subset \mathcal{X}$ in the search region k, according to

$$b_t^k(a_t^k) \approx c \sum_{i=1}^{|\mathcal{Y}^k|} p(y_t^k|x_t^{k,i})\bar{b}_t^{k,i} \quad \text{if } a_t^k = 1 \tag{22}$$

where c is a normalizing constant, $|\mathcal{Y}^k|$ is the number of detection windows and $\bar{b}_t^{k,i} = p(x_t^i|\bar{b}_t^k)$. Each $p(y_t^k|x_t^{k,i})$ is queried on-line from the learned observation model. Assessing multiple $x_t^i \in u_t^k$ instead of just $\hat{x}_t$ should better approximate the error distribution.

D. Resource constrained POMDP with belief-dependent rewards

As previously noted the decision making involved in resource constrained multiple target tracking scenarios can be formulated within the POMDP framework. The perceptual agent tries to minimize tracking uncertainty by prioritizing its limited attentional resources to promising image regions. The instantaneous reward function should thus reflect the action contribution to maximizing the information regarding the targets' states. Similarly to (Araya et al., 2010) let us define the instantaneous reward at time t as the negative entropy of the belief state, given by the following expectation

$$r(b_t^k(a_t^k)) = \int b_t^k \log b_t^k dx_t \tag{23}$$

For Gaussian beliefs this reward becomes simply given by

$$r(b_t^k(a_t^k)) \approx -\log(|\Sigma_t^k|) \tag{24}$$

Inspired by the evidence of visual processing capacity limitations in humans (Xu and Chun, 2009), we formulate the proposed resource constrained information maximization as follows:

$$\begin{aligned} \underset{a}{\text{maximize}} \quad & R_T = E\left[\sum_{\tau=1}^{T} \gamma^\tau \sum_{k=1}^{K} r(b_{t+\tau}^k(a_{t+\tau}^k))\right] \\ \text{subject to} \quad & \sum_{k=1}^{K} a_{t+\tau}^k \le K_{\text{max}} \quad \forall_{\tau\in\{1,\ldots,T\}} \\ & \sum_{k=1}^{K} a_{t+\tau}^k |u_{t+\tau}^k| \le S_{\text{max}} A_p \quad \forall_{\tau\in\{1,\ldots,T\}} \end{aligned}$$

where T is the planning horizon, $E[\cdot]$ is the expectation operation, $r(\cdot)$ is the reward function, $\gamma \in]0,1]$ is a discount factor, $|u_{t+\tau}^k|$ is the area of the k search region, K_{max} is the maximum region-based activation capacity, A_p is the image pixel area and S_{max} is the relative maximum image area that the visual system may process per time-instant. The first constraint reflects short-term memory limitations and allows reducing the action space (assuming $K_{\text{max}} < K$), and thus the branching factor during planning. The second is motivated by computational effort and timing limitations that arise during visual processing and contributes to prune infeasible planning tree branches, by prioritizing resources to higher uncertainty targets.

E. Monte Carlo Tree Search (MCTS)

The MCTS algorithm relies on Monte-Carlo simulations to assess the nodes of a search tree in a best-first order, by prioritizing the expansion of the most promising nodes according to their expected reward.

In a nutshell, the algorithm runs Monte Carlo simulations from the current belief state (i.e. input root node), and progressively builds a tree of belief states and outcomes. In the end, the most promising action is returned. Each run comprises four phases (see Fig. 2):

1) Selection: In the selection step a sequence of actions are chosen within the search tree. Tree descending is performed from the root until a leaf node is reached. Action selection is typically carried out using an algorithm named Upper Confidence Bounds for Trees (UCT) (Kocsis and Szepesvári, 2006), which elegantly balances the exploration-exploitation trade-off, during action selection. On the one hand, based on the current accumulated simulated knowledge, the planning agent should select actions that may lead to the best immediate

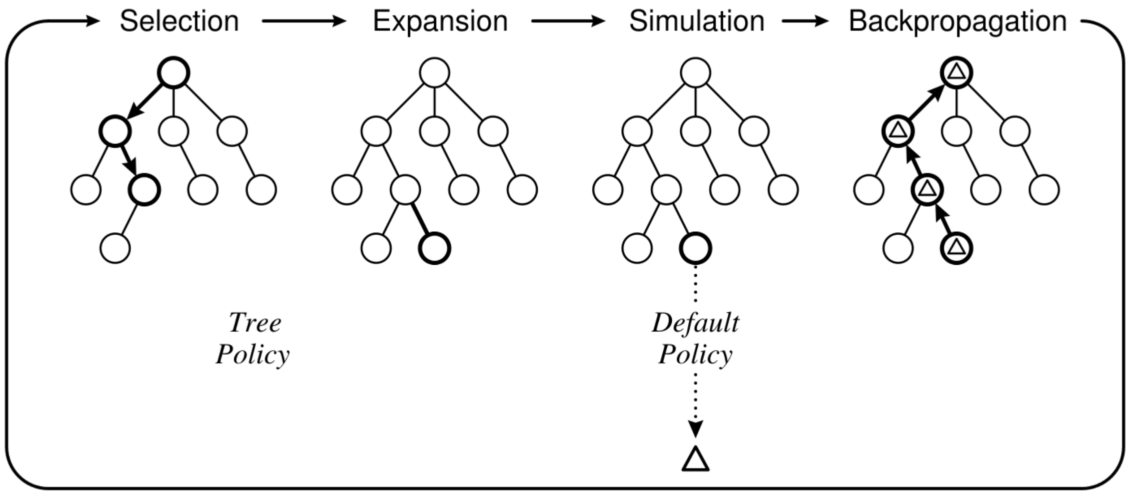

Fig. 2: Monte Carlo Tree Search (image taken from (Browne et al., 2012)].

payoffs (exploitation). On the other hand, the agent should select unexplored actions since they may yield better long-term outcomes;

2) Expansion: an action that leads to an unvisited node is selected and the resulting expanded leaf node is appended to the tree;
3) Simulation: From the expanded node, actions are taken randomly in a Monte-Carlo depth-first manner, until a predefined horizon or a terminal state is reached. Simulation depth (i.e. time horizon) is typically fixed, to deal with real-time constraints. Since sampling from a uniform distribution over actions may be suboptimal, problem specific knowledge should be incorporated to give larger sampling probabilities to more promising actions. In our specific problem, we bias this sampling distribution such that regions with higher entropy are prioritized.
4) Back-propagation: Finally, the simulation rewards are back-propagated to the root node. This includes updating the reward rate stored at each node along the way.

Finally, runs are repeated until a computational budget (i.e. a triggering timeout or a maximum number of iterations) is reached, and the best action from the root node is selected.

1) Upper Confidence Bounds for Trees (UCT)

The idea of using Upper Confidence Bounds (Auer et al., 2002) on rewards to deal with the exploration exploitation dilemma in the face of uncertainty, has been widely applied to reinforcement learning problems. In MCTS, Upper Confidence Bounds for Trees (UCT) are typically employed in the selection phase, while descending the tree. The upper confidence bound accounts for the currently estimated value of the action, and the estimated UCT variance, according to

$$UCT(a) = r + c\sqrt{\frac{\log n_v}{n_a}} \tag{25}$$

where r is the estimate for the value of the action based on the simulated payoffs, n_v is the number of times the node has been visited, and n_a is the number of times an action a has been tried from that node. The constant c is a problem-dependent parameter that balances the exploration-exploitation trade-off.

IV. Experiments

In order to evaluate the proposed resource-constrained tracking approach we performed a set of experiments on the TUD-Stadtmitte MOTChallenge dataset (Leal-Taixé et al., 2015), which allows to evaluate tracking performance with the CLEAR MOT metrics and known ground truth (Bernardin and Stiefelhagen, 2008). This dataset comprises a video sequence of 179 images, acquired with a static camera with 640×480 image resolution. An average of 8 pedestrians are present in the visual field, during the video. To quantitatively assess the performance of our methodologies we focused our evaluation in the time speed-up gains and in the multiple object tracking precision (MOTP), which is the total error in estimated position for matched object-hypothesis pairs over all frames, averaged by the total number of matches:

$$\text{MOTP} = \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \tag{26}$$

where $d_t^i \in [0, 100]$ quantifies the amount of overlap (in percentage) between the true object o_i and its associated hypothesis bounding boxes, and where c_t is the number of matches found for time t. The MOTP shows the ability of the tracker to keep consistent trajectories.

Our aim was to investigate the performance of the proposed methodologies dependency on the resource-constraints. We considered the following activation capacities $K_{\max} \in \{1, 2, 3, 4\}$ and maximum processing image areas $S_{\max} \in [0.1, 1.0]$. Since the MCTS method is randomized, we performed 100 trials for each combination of parameters. The region size parameters where found empirically and were set to $\alpha_c = \alpha_s = 1$. At each time step, the MCTS planning root node was set to the current tracking belief, and the algorithm was allowed to run for $10ms$. Finally, the simulation step depth was set to 3 and γ was set to 0.9. The association between detections and trackers was performed with the Hungarian Algorithm (Burkard et al., 2009) using the Mahalanobis distance. The tracking process is bootstrapped in the first frame, by applying the pedestrian detector to the whole image and instantiating a tracker for each detection. These trackers are kept during the entire video sequence, and every non-assigned detection is discarded, i.e., trackers are not further created.

The results presented in Figure 3 demonstrate that planning future resource allocations in a constrained setting, improves simultaneously detection times and tracking precision, when compared with the baseline, full-window detector.

As illustrated by the temporal gain plots (first row of Figure 3), our method achieves detection times around 12 times faster than the baseline detector applied to the full-window (0.02 against 0.24 seconds, for $S_{\max} = 0.1$), with comparable tracking performance. Furthermore, the MOTP metric results demonstrate that, on the one hand, constraining the attention to regions with high probability of pertaining a person, allows to improve detection accuracy and to reduce the possibility of erroneous detections in the targets' vicinities, which may lead to bad detection-tracker associations and hence degrade tracking precision. On the other hand, ignoring regions that are unlikely to contain a person allows to reduce the number of spurious wrong detections (i.e. False positives) that may also contribute to tracking performance degradation.

In conclusion, in the constrained setting the allocation of more computational resources yields better tracking precision, at the cost of increased computational effort. Therefore, depending on the application requirements, this trade-off can be easily balanced by carefully selecting the $K_{\max}$ and $S_{\max}$ resource-constraints.

V. Conclusions and Future Work

In this paper we have addressed the multiple object tracking (MOT) problem with constrained resources, which plays a fundamental role in the deployment of efficient mobile robots for real-time applications involved in HRI. We have framed the multiple object tracking within the POMDP domain and proposed a problem formulation that allows for on-line, real-time, planning with a state-of-the-art Monte Carlo Tree Search methodology. The results presented in this work show that directing the attentional focci to important image sub-regions allows for large detection speed-ups improvements on tracking precision.

The major limitation of our approach is still its incapacity of dealing with non-sparse targets. In the future, data association should also be considered during planning by integrating data association methodologies such as joint probabilistic data-association (JPDA) (Hamid Rezatofighi et al., 2015). Another shortcoming of our methodology is its incapacity of locating new pedestrians appearing on the scene, in an efficient manner. However, this can be easily overcome by considering proposals generated by bottom-up saliency methods.

Finally, we note that the targets' dynamics and the observation distributions are extremely non-linear and non-Gaussian. Therefore, a mixture of particle filters (Okuma et al., 2004) would be more appropriate for our particular problem, and hence improve tracking accuracy at the cost of some additional computational effort.

ACKNOWLEDGMENT

This work has been partially supported by the Portuguese Foundation for Science and

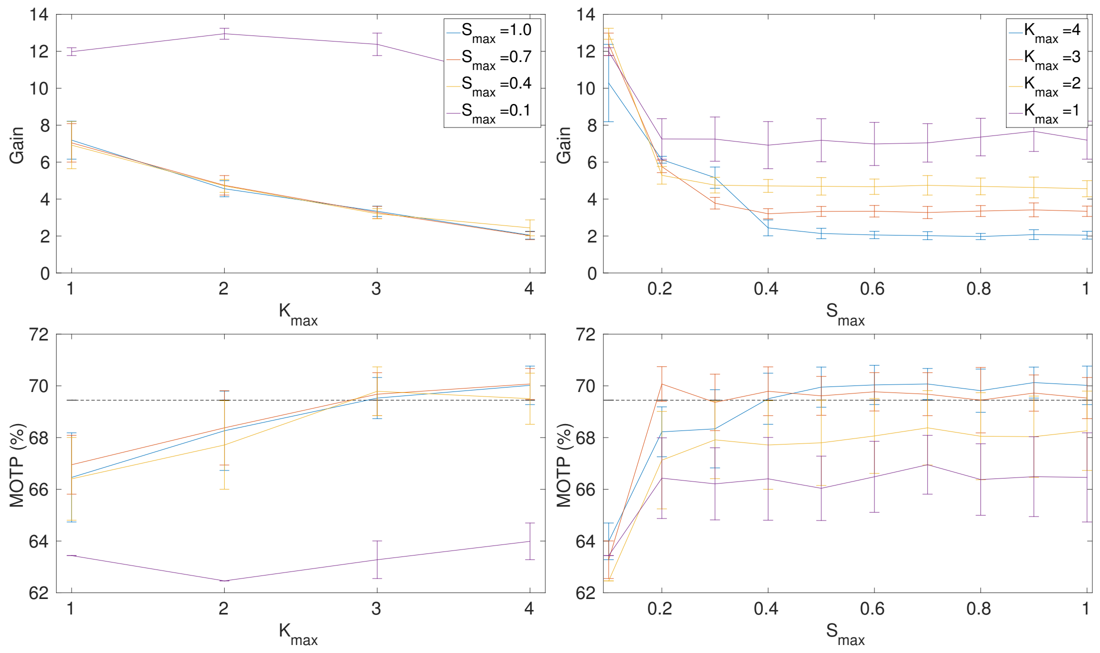

Fig. 3: Speed-up gains and resulting Multiple Object Tracking Precision (MOTP). Bottom-row: Dashed black line represents the baseline full-window detector.

Technology (FCT) project [UID/EEA/50009/2013]. Rui Figueiredo is funded by FCT PhD grant PD/BD/105779/2014.

REFERENCES

Ahmad, S. and Yu, A. J. (2013). Active sensing as bayes-optimal sequential decision making. *CoRR*, abs/1305.6650.

Araya, M., Buffet, O., Thomas, V., and Charpillet, F. (2010). A pomdp extension with belief-dependent rewards. In *Advances in Neural Information Processing Systems*, pages 64–72.

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256.

Bernardin, K. and Stiefelhagen, R. (2008). Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008(1):1–10.

Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B. (2016). Simple online and realtime tracking. *CoRR*, abs/1602.00763.

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. (2012). A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43.

Burkard, R., Dell'Amico, M., and Martello, S. (2009). *Assignment Problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Butko, N. J. and Movellan, J. R. (2010). Infomax control of eye movements. *Autonomous Mental Development, IEEE Transactions on*, 2(2):91–107.

Chong, E. K., Kreucher, C. M., and Hero, A. O. (2008). Monte-carlo-based partially observable markov decision process approximations for adaptive sensing. In *Discrete Event Systems, 2008. WODES 2008. 9th International Workshop on*, pages 173–180. IEEE.

Dollár, P., Appel, R., Belongie, S., and Perona, P. (2014). Fast feature pyramids for object detection. *PAMI*.

Gedikli, S., Bandouch, J., von Hoyningen-Huene, N., Kirchlechner, B., and Beetz, M. (2007). An adaptive vision system for tracking soccer players from variable camera settings. In *Proceedings of the 5th International Conference on Computer Vision Systems (ICVS)*.

Gelly, S., Kocsis, L., Schoenauer, M., Sebag, M., Silver, D., Szepesvári, C., and Teytaud, O. (2012). The grand challenge of computer go: Monte carlo tree search and extensions. *Communications of the ACM*, 55(3):106–113.

Hamid Rezatofighi, S., Milan, A., Zhang, Z., Shi, Q., Dick, A., and Reid, I. (2015). Joint probabilistic data association revisited. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3047–3055.

Kocsis, L. and Szepesvári, C. (2006). Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer.

Leal-Taixé, L., Milan, A., Reid, I., Roth, S., and Schindler, K. (2015). MOTChallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942 [cs]*. arXiv: 1504.01942.

Malloy, M. L. and Nowak, R. D. (2014). Near-optimal adaptive compressed sensing. *IEEE Transactions on Information Theory*, 60(7):4001–4012.

Mihaylova, L., Lefebvre, T., Bruyninckx, H., Gadeyne, K., and Schutter, J. D. (2002). Active sensing for robotics - a survey. In *in Proc. 5 th Intl Conf. On Numerical Methods and Applications*, pages 316–324.

Okuma, K., Taleghani, A., De Freitas, N., Little, J. J., and Lowe, D. G. (2004). A boosted particle filter: Multitarget detection and tracking. In *European Conference on Computer Vision*, pages 28–39. Springer.

Ross, S., Pineau, J., Paquet, S., and Chaib-Draa, B. (2008). Online planning algorithms for pomdps. *Journal of Artificial Intelligence Research*, 32:663–704.

Sommerlade, E. and Reid, I. (2010). Probabilistic surveillance with multiple active cameras. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 440–445. IEEE.

Spaan, M. T., Veiga, T. S., and Lima, P. U. (2015). Decision-theoretic planning under uncertainty with information rewards for active cooperative perception. *Autonomous Agents and Multi-Agent Systems*, 29(6):1157–1185.

Van Rooij, I. (2008). The tractable cognition thesis. *Cognitive science*, 32(6):939–984.

Wang, J. R. and Parameswaran, N. (2004). Survey of sports video analysis: research issues and applications. In *Proceedings of the Pan-Sydney area workshop on Visual information processing*, pages 87–90. Australian Computer Society, Inc.

Xu, Y. and Chun, M. M. (2009). Selecting and perceiving multiple visual objects. *Trends in cognitive sciences*, 13(4):167–174.

Zitnick, C. L. and Dollár, P. (2014). Edge boxes: Locating object proposals from edges. In *ECCV*.