Active Object Recognition via Monte Carlo Tree Search

Mikko Lauri, Nikolay Atanasov, George J. Pappas, and Risto Ritala

***Abstract*—This paper considers object recognition with a camera, whose viewpoint can be controlled in order to improve the recognition results. The goal is to choose a multi-view camera trajectory in order to minimize the probability of having misclassified objects and incorrect orientation estimates. Instead of using *offline* dynamic programming, the resulting stochastic optimal control problem is addressed via an *online* Monte Carlo tree search algorithm, which can handle various constraints and provides exceptional performance in large state spaces. A key insight is to use an active hypothesis testing policy to select camera viewpoints during the rollout stage of the tree search.**

I. INTRODUCTION

The goal of this paper is to choose a sequence of views for an RGB-D camera in order to identify the class and orientation of an object of interest (see Fig. 1). Unlike many existing approaches, which consider a next-best-view problem [1], [2], [3], we plan a multi-view camera trajectory to minimize the probability of having misclassified objects and incorrect orientation estimates. In previous work [4], we addressed a similar stochastic optimal control problem by casting it as a partially-observable Markov decision process. A point-based approximate solver [5] was used to obtain a non-greedy policy offline. Since repeated observations of the object from the same viewpoint provide redundant information, it is desirable to disallow viewpoint revisiting. The drawback of computing a policy offline is that revisiting and occlusion constraints are hard to incorporate and if the environment were to change, the computed policy would no longer be useful. The idea of this paper is to apply Monte Carlo tree search (MCTS, [6], [7]) to the active object recognition problem. MCTS is a best-first *online* planning approach which can handle various constraints and has exceptional performance in large challenging domains such as game solving [8], [9] and belief-space planning in robotics [10], [11], [12].

II. PROBLEM FORMULATION

Let the camera pose at time t be $x_t \in \mathcal{X} \subset SE(3)$, where $\mathcal{X}$ is a finite set of viewpoints on a sphere centered at the object's location (see Fig. 1). At time t, the camera can move to any of the viewpoints in $\mathcal{X}$ and pays a cost $g(x_{t-1}, x_t)$ which captures the energy expenditure. Let the true (unknown) class of the observed object be $c \in \mathcal{C}$. We formulate hypotheses about the class and orientation of the object:

$H(c, r)$: the object class is $c \in \mathcal{C}$ with orientation $r \in \mathcal{R}(c)$,

M. Lauri and R. Ritala are with the Department of Automation Science and Engineering, Tampere University of Technology, P.O. Box 692, FI-33101, Tampere, Finland, {mikko.lauri, risto.ritala}@tut.fi.

N. Atanasov and G. Pappas are with GRASP Lab, University of Pennsylvania, Philadelphia, PA 19104, USA, {atanasov, pappasg}@seas.upenn.edu. This work was supported by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

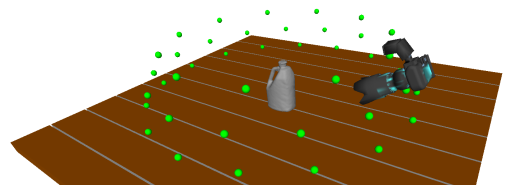

Fig. 1: Setup for the active object recognition problem. The camera position is restricted to a set of viewpoints (green) on a sphere centered at the object's location. The task is to choose a camera control policy, which minimizes the movement cost and the probability of misclassification.

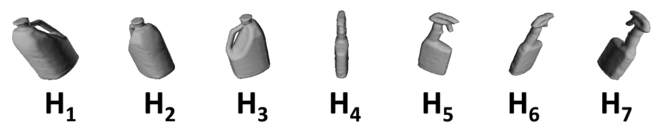

Fig. 2: An example set of hypotheses about the class and orientation of an unknown object

where $\mathcal{R}(c) \subset SO(3)$ is a small finite set of discrete[1] orientations for each class $c \in \mathcal{C}$. For notational convenience, let $i = 1, \ldots, M$ be an enumeration of the set $\{(c, r) \mid c \in \mathcal{C}, r \in \mathcal{R}(c)\}$ and denote the hypotheses by H_i (see Fig. 2).

Offline, a 3-D model database is used to train a viewpoint-pose tree [4] by extracting point clouds from views on a sphere around each model. A set of Fast Point Feature Histograms [13] is extracted from each point cloud and the clouds are arranged in a tree structure according to their feature similarity (see [4] for details). Given a query point cloud, the best-matching cloud from the tree carries information about the class and orientation of the observed object and about the quality of the feature match. Thus, the tree provides an observation $z_t \in \mathcal{Z}$, consisting of the class, orientation, and confidence score of the top match. The model database is used to learn the probability density function (pdf) $q(\cdot \mid x, H_i)$ of z conditioned on any camera viewpoint $x \in \mathcal{X}$ and any hypothesis H_i, $i = 1, \ldots, M$.

Problem. *Given a camera pose $x_0 \in \mathcal{X}$, a prior $p_0 \in [0,1]^M$ on the true hypothesis H_i, and a planning horizon $T < \infty$, choose a sequence of functions $\mu_t : (\mathcal{Z} \times \mathcal{X})^{t+1} \to \mathcal{X}$ for $t = 0, \ldots, T-1$, which minimizes the average movement cost and the probability of an incorrect hypothesis:*

$$\begin{aligned} \min_{\mu_{0:T-1}} \; & \frac{1}{T} \sum_{t=1}^{T} g(x_{t-1}, x_t) + \lambda Pe(T) \\ \textit{s.t. } & x_{t+1} = \mu_t(z_{0:t}, x_{0:t}), \quad t = 0, \ldots, T-1, \\ & x_{t+1} \notin \{x_0, \ldots, x_t\}, \quad t = 0, \ldots, T-1, \\ & z_t \sim q(\cdot \mid x_t, H_i), \qquad t = 0, \ldots, T, \\ & p_t = b(p_{t-1}, z_t, x_t), \quad t = 1, \ldots, T, \end{aligned} \tag{1}$$

[1] After a hypothesis is chosen, the discrete orientation estimate can be refined by aligning the observed object surface to the corresponding model in the training database, e.g., by using the iterative closest point algorithm.

where $\lambda \geq 0$ determines the relative importance of a correct decision versus cost of movement, $b(p,z,x) := \frac{p \odot q(z|x,\cdot)}{p^T q(z|x,\cdot)}$ is the Bayesian update, $\hat{i}_t := \underset{i \in \{1,\ldots,M\}}{\arg\max}\ p_t(i)$ is the maximum-likelihood estimate of the true hypothesis, and $Pe(t)$ is the probability of error:

$$Pe(t) := \mathbb{P}(i \neq \hat{i}_t) = \mathbb{E}_{z_{0:t}} \sum_{i=1}^{M} \left(\mathbb{1}_{\left\{ i \neq \underset{j \in \{1,\ldots,M\}}{\arg\max}\ p_t(j) \right\}} p_t(i) \right)$$
$$= \mathbb{E}_{z_{0:t}} \left(1 - \max_{i \in \{1,\ldots,M\}} p_t(i) \right).$$

III. Monte Carlo Tree Search

Monte Carlo tree search (Alg. 1) is an online simulation-based alternative to the exact dynamic programming solution of the active object recognition problem. MCTS constructs a tree sequentially in a best-first order. A node in the tree corresponds to a state (x_t, p_t) and contains a visitation count and the total cost accumulated over all simulations, both initialized to 0. Each simulation has two stages: a *tree policy* and a *rollout policy*. The tree policy (lines 12-18) is followed until reaching a leaf node. The real work of the tree policy is done by the SELECTCHILD function (line 12), which uses the UCT (Upper Confidence bounds applied to Trees) method [6] to select the next node as follows:

$$n' = \underset{n \in \text{CHILDREN}(\text{ node })}{\arg\min} \frac{n.\text{TotalCost}}{n.\text{Visits}} + \kappa \sqrt{\frac{\log(node.\text{Visits})}{n.\text{Visits}}},$$

where κ is an exploration parameter encouraging selection of rarely-visited viewpoints. Once a leaf is reached, the rollout policy (lines 9-11) computes the cost-to-go by choosing viewpoints and simulating measurements until the end of the planning horizon T. The most common choice for a rollout policy is one that picks successors uniformly at random [8]. While the tree policy has been subject to extensive research [6], [7], since it determines if MCTS converges asymptotically to the optimal policy, the choice of rollout policy has received less attention. MCTS converges for any choice of rollout policy, but the convergence speed may be affected [6], [7]. In the next section, we propose rollout policies which we expect to be particularly suited for object recognition problems.

We used MCTS with a uniform rollout policy and a computational budget of 1500 simulations to solve an active object recognition problem with $M = 16$ hypotheses, $|\mathcal{X}| = 42$ viewpoints, and $|\mathcal{Z}| = 7056$ possible observations, comprising all combinations of object classes, orientations, and discretized confidence scores. The movement cost g was the great circle distance between viewpoints on the sphere, scaled to the range $[0, 1]$. We compared the performance of the policy obtained by MCTS to that of a greedy policy, minimizing the sum of the movement cost and error probability at the next time step. A total of 12 classification tasks were executed with a planning horizon of $T = 3$. In each case, the true (unknown) hypothesis was chosen at random among the 16 possibilities. The results are summarized in Table I. We note that MCTS outperforms the greedy policy in classification accuracy, while accumulating a slightly greater movement cost. We expect that the average planning times per decision can be decreased with an optimized implementation.

Algorithm 1 Monte Carlo Tree Search (x_0, p_0)

1: root $\leftarrow$ TREENODE(x_0,p_0)
2: **while** within computational budget **do**
3: SIMULATE(root, 0)
4: best_node = $\underset{n \in \text{CHILDREN(root)}}{\arg\min} \frac{n.\text{TotalCost}}{n.\text{Visits}}$
5: **return** best_node.x
6:
7: **function** SIMULATE(node, t)
8: **if** $t = T$ **then return** $\lambda(1 - \max_i \text{node}.p(i))$
9: **if** ISLEAF(node) **then**
10: EXPANDTREE(node)
11: **return** ROLLOUT(node, t)
12: $n' \leftarrow$ SELECTCHILD(node)
13: $z \leftarrow$ SAMPLEOBSERVATION($n'.x$, p)
14: $n'.p \leftarrow b(p, z, n'.x)$
15: $J \leftarrow \frac{c(\text{node}.x, n'.x)}{T}$ + SIMULATE(n', $t+1$)
16: node.Visits $\leftarrow$ node.Visits $+1$
17: node.TotalCost $\leftarrow$ node.TotalCost $+J$
18: **return** J

TABLE I: Active object recognition results (average values)

Policy	Classification accuracy [%]	Movement cost	Planning time [s]
MCTS	41.7	0.0882	4.33
Greedy	25.0	0.0719	0.19

IV. Discussion and Future Work

In many applications, the performance of MCTS is improved significantly by using domain-specific knowledge to design a rollout policy [8]. Policies with asymptotic optimality guarantees based on Jensen-Shannon divergence [14], [15] and Chernoff information [16], [17] have been proposed for active hypothesis testing. Since object recognition is closely related, these policies are excellent candidates for rollout in MCTS (Alg. 1, line 11). Future work will focus on proving that such policies outperform the commonly-used uniform rollout policy. Another key questions is the particular choice of information measure (e.g. mutual information, probability of error, conditional entropy) in the problem formulation and during rollout. For example, a closed-form expression exists for the Chernoff information of density functions from the exponential family of order 1 (e.g., Binomial, Bernoulli, Laplacian, Rayleigh, etc.) [18] and is ideal for planning informative actions quickly. In similar vein, Charrow et al. [19] recently proposed an alternative to the classical mutual information, which can be computed in closed-form for some measurement distributions. Fano's inequality [20], $\mathbb{H}(H_i \mid z_{0:t}) \leq Pe(t) \log(M-1) + \mathbb{H}(Pe(t))$, provides a connection between the probability of error and conditional entropy, which in turn is related to the mutual information. Careful selection of a cost function and a rollout policy for MCTS may impact many robotic planning tasks that optimize information gathering. Examples include active localization and mapping [21], environmental monitoring [22], search and rescue [23], surveillance and reconnaissance [24], and many others.

References

[1] C. Potthast and G. Sukhatme, "A Probabilistic Framework for Next Best View Estimation in a Cluttered Environment," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, 2014.

[2] B. Browatzki, V. Tikhanoff, G. Metta, H. Bulthoff, and C. Wallraven, "Active Object Recognition on a Humanoid Robot," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[3] H. Borotschnig, L. Paletta, M. Prantl, and A. Pinz, "Appearance-based Active Object Recognition," *Image and Vision Computing*, vol. 18, no. 9, 2000.

[4] N. Atanasov, B. Sankaran, J. Le Ny, G. Pappas, and K. Daniilidis, "Nonmyopic View Planning for Active Object Classification and Pose Estimation," *IEEE Trans. on Robotics*, vol. 30, no. 5, pp. 1078–1090, 2014.

[5] H. Kurniawati, D. Hsu, and W. Lee, "SARSOP: Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces," *Robotics: Science and Systems*, 2008.

[6] L. Kocsis and C. Szepesvári, "Bandit based Monte-Carlo Planning," in *European Conference on Machine Learning (ECML)*, 2006.

[7] D. Silver and J. Veness, "Monte-Carlo Planning in Large POMDPs," in *Neural Information Processing Systems (NIPS) 23*, 2010, pp. 2164–2172.

[8] C. Browne, E. Powley, D. Whitehouse, S. Lucas, P. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A Survey of Monte Carlo Tree Search Methods," *IEEE Trans on Computational Intelligence and AI in Games*, vol. 4, no. 1, 2012.

[9] S. Gelly and D. Silver, "Combining Online and Offline Knowledge in UCT," in *Int. Conf. on Machine Learning*, 2007, pp. 273–280.

[10] K. Hauser, "Randomized Belief-Space Replanning in Partially-Observable Continuous Spaces," in *Algorithmic Foundations of Robotics IX*, ser. Springer Tracts in Advanced Robotics, D. Hsu, V. Isler, J.-C. Latombe, and M. Lin, Eds. Springer Berlin Heidelberg, 2011, vol. 68, pp. 193–209.

[11] J. Nguyen, N. Lawrance, and S. Sukkarieh, "Nonmyopic Planning for Long-term Information Gathering with an Aerial Glider," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014.

[12] M. Lauri and R. Ritala, "Stochastic Control for Maximizing Mutual Information in Active Sensing," in *IEEE Int. Conf. on Robotics and Automation (ICRA) Workshop on Robots in Homes and Industry*, 2014.

[13] R. Rusu, "Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments," Ph.D. dissertation, Technische Universität München, 2009.

[14] M. Naghshvar and T. Javidi, "Active Sequential Hypothesis Testing," *arXiv:1203.4626*, 2012.

[15] ——, "Sequentiality and Adaptivity Gains in Active Hypothesis Testing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 5, pp. 768–782, 2013.

[16] S. Nitinawarat and V. Veeravalli, "Controlled Sensing for Sequential Multihypothesis Testing with Controlled Markovian Observations and Non-Uniform Control Cost," *arXiv:1310.1844*, 2013.

[17] S. Nitinawarat, G. Atia, and V. Veeravalli, "Controlled Sensing for Multihypothesis Testing," *IEEE Trans on Automatic Control (TAC)*, vol. 58, no. 10, 2013.

[18] F. Nielsen, "Chernoff Information of Exponential Families," *arXiv preprint:1102.2684*, 2011.

[19] B. Charrow, S. Liu, V. Kumar, and N. Michael, "Information-Theoretic Mapping Using Cauchy-Schwarz Quadratic Mutual Information," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015.

[20] T. Cover and J. Thomas, *Elements of Information Theory*, 2nd ed. Wiley-Interscience, 2006.

[21] N. Atanasov, J. Le Ny, K. Daniilidis, and G. Pappas, "Decentralized Active Information Acquisition: Theory and Application to Multi-Robot SLAM," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015.

[22] H. Choi, "Adaptive Sampling and Forecasting With Mobile Sensor Networks," Ph.D. dissertation, MIT, 2009.

[23] V. Kumar, D. Rus, and S. Singh, "Robot and Sensor Networks for First Responders," *IEEE Pervasive Computing*, vol. 3, no. 4, 2004.

[24] P. Rybski, S. Stoeter, M. Erickson, M. Gini, D. Hougen, and N. Papanikolopoulos, "A Team of Robotic Agents for Surveillance," in *Int. Conf. on Autonomous Agents*, 2000.