Unsupervised Active Visual Search with Monte Carlo Planning under Uncertain Detections

Unsupervised Active Visual Search With Monte Carlo Planning Under Uncertain Detections

Francesco Taioli, Francesco Giuliari, *Student Member, IEEE*, Yiming Wang, *Member, IEEE*, Riccardo Berra, Alberto Castellini, Alessio Del Bue, *Member, IEEE*, Alessandro Farinelli, Marco Cristani, *Member, IEEE*, and Francesco Setti, *Member, IEEE*

***Abstract*—We propose a solution for Active Visual Search of objects in an environment, whose 2D floor map is the only known information. Our solution has three key features that make it more plausible and robust to detector failures compared to state-of-the-art methods: *i)* it is unsupervised as it does not need any training sessions. *ii)* During the exploration, a probability distribution on the 2D floor map is updated according to an intuitive mechanism, while an improved belief update increases the effectiveness of the agent's exploration. *iii)* We incorporate the awareness that an object detector may fail into the aforementioned probability modelling by exploiting the success statistics of a specific detector. Our solution is dubbed POMP-BE-PD (Pomcp-based Online Motion Planning with Belief by Exploration and Probabilistic Detection). It uses the current pose of an agent and an RGB-D observation to learn an optimal search policy, exploiting a POMDP solved by a Monte-Carlo planning approach. On the Active Vision Dataset Benchmark, we increase the average success rate over all the environments by a significant 35% while decreasing the average path length by 4% with respect to competing methods. Thus, our results are state-of-the-art, even without any training procedure.**

***Index Terms*—Active vision dataset benchmark, active visual search, object goal navigation, online policy learning, partially observable Markov decision process, partially observable Monte Carlo planning.**

I. Introduction

AMONG the most interesting areas of robotics is the problem of Active Visual Search (AVS) [1], in which an intelligent robotic agent must autonomously find an object located far from it, moving and exploring its surroundings through egocentric visual sensors. AVS applies in many different contexts, such as the domotic field [2], [3], [4], personal assistance [5], search and rescue [6], [7], and the very intriguing Mars exploration [8]. In this paper we focus on the AVS problem in an indoor environment [9], where the only available knowledge is its 2D map. We propose a method for learning a motion planning policy that decides how to move an agent based on a perception module to visually detect and approach a specific object, i.e., the target (see Fig. 1).

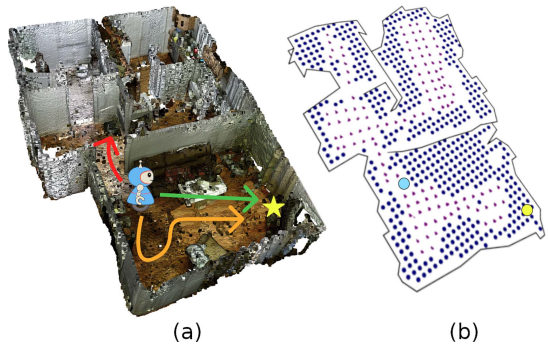


Fig. 1. An agent is initialised in a known environment with the task of visually searching for a target object, i.e., to localise the object and approach it. (a) 3D reconstruction of the environment; the agent has to navigate toward the target (yellow star) through the possible shortest path (highlighted in green) while avoiding longer trajectories (in orange) without missing entirely the target (in red). (b) Corresponding 2D grid map of the scene in our POMCP modelling: blue dots are the possible object locations, purple crosses are the possible robot poses.

AVS in real-world scenarios with egocentric camera views is a very challenging problem due to the unpredictable quality of the observations –i.e., object in the far field, motion blur and low resolution–, partial views and occlusions due to scene clutters and generalisation to new environment. This has an impact not only on the object detection but also on the planning policy. To address this challenge, recent efforts are mostly based on deep Reinforcement Learning (RL), e.g., deep recurrent Q-network (DRQN), fed with deep visual embedding [10], [11]. To train such DRQN models, a large amount of data is required, which are sequences of observations of various lengths, covering successful and unsuccessful search episodes from multiple real scenarios or simulated environments.

In this paper, we take a different perspective and propose an online reinforcement learning method for AVS. The basic idea is not to learn a complete policy by using a vast amount of training data, instead to use an advanced planning approach to learn a policy that can select the best action based on the environment configuration which is built starting from the

Received 2 March 2023; revised 12 April 2024; accepted 22 August 2024. Date of publication 29 August 2024; date of current version 5 November 2024. Recommended for acceptance by G. Farinella. *(Corresponding author: Francesco Taioli.)*

Francesco Taioli is with the Polytechnic of Turin, 10129 Turin, Italy, and also with the Department of Engineering for Innovation Medicine, University of Verona, 37129 Verona, Italy (e-mail: francesco.taioli@polito.it).

Francesco Giuliari is with the Pattern Analysis and Computer Vision (PAVIS) Research Line, Istituto Italiano Di Tecnologia (IIT), University of Genoa, 16126 Genova, Italy.

Yiming Wang is with the Deep Visual Learning (DVL) Unit, Fondazione Bruno Kessler (FBK), 38123 Trento, Italy.

Riccardo Berra, Marco Cristani, and Francesco Setti are with the Department of Engineering for Innovation Medicine, University of Verona, 37129 Verona, Italy.

Alberto Castellini and Alessandro Farinelli are with the Department of Computer Science, University of Verona, 37129 Verona, Italy.

Alessio Del Bue is with the Pattern Analysis and Computer Vision (PAVIS) Research Line, Istituto Italiano di Tecnologia (IIT), 16126 Genova, Italy.

Code at https://intelligolabs.github.io/unsupervised_active_visual_search/

Digital Object Identifier 10.1109/TPAMI.2024.3451994

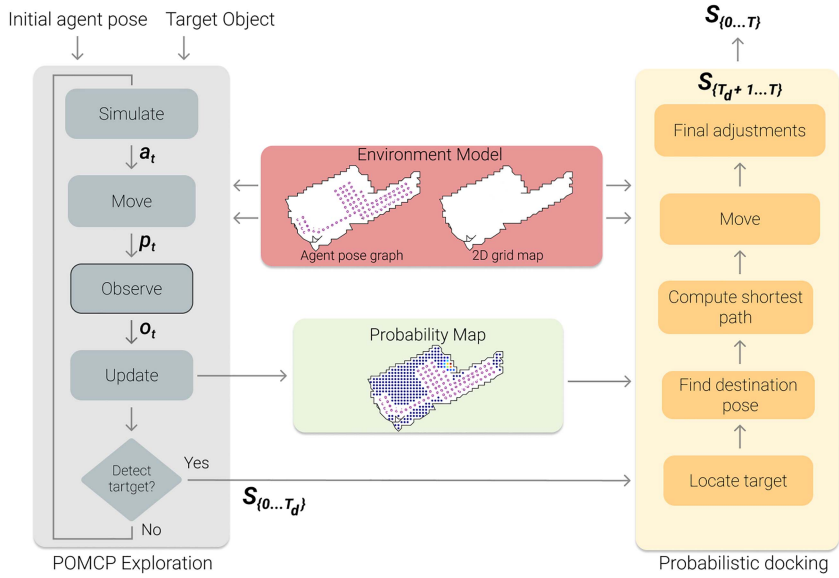


Fig. 2. Overall architecture of our proposed method POMP-BE-PD. The red box represents prior knowledge pushed into the POMCP module, the grey box represents the exploration strategy to detect the target object, the yellow box represents the probabilistic docking strategy to reach the destination pose and the green box represents the probability distribution over the locations. Math notation: state s_t, action a_t, pose p_t, observation o_t, POMCP state sequence $s_{\{0..T_d\}}$, docking state sequence $s_{\{T_{d+1}..T\}}$, complete state sequence $s_{\{0..T\}}$.

observations gathered in the environment. This fundamental shift in the methodology is carried out considering the Partially Observable Monte Carlo Planning (POMCP) method [12]. In the following, we will use the term *learning* referring to the process of Q-value approximation from simulations performed using the model of the environment, instead of referring to a process of function parameter tuning from data. POMCP has been applied to benchmark problems, such as *rocksample*, *battleship* and *pocman* (partially observable pacman) with impressive results, however, its use for robotic applications is an open and challenging research problem.

In our previous work [13], we introduced POMP, an online reinforcement learning method, that uses as input the current pose of an agent and an RGB-D frame to plan the next move that brings the agent closer to the target object. We modelled the problem as a Partially Observable Markov Decision Process solved by a Monte-Carlo planning approach, allowing us to explore the environment and search for the object at the same time. The main benefit of this approach is that POMP does not require a training phase, so being very agile in solving AVS in small and medium real scenarios. Despite achieving results close to the state-of-the-art without using any training data, POMP uses real object detectors that are inaccurate with false positives and miss-detections. As a consequence, an agent could terminate the exploration in wrong locations, fooled by the detector, thus decreasing the overall success rate.

To overcome these problems, in this paper we propose POMP-BE-PD, an extension of POMP that defines the observation model in probabilistic terms, allowing us to better handle the false positives of object detectors, as well as improving the effectiveness of the agent's explorations. A visual representation of our approach is shown in Fig. 2. At each time step, we feed our model with the agent pose –i.e., position and orientation– in a known 2D map and an RGB-D frame given by a sensor acquisition. An off-the-shelf object detector is applied to the RGB image to identify the bounding box of the target object, if present. The depth channel of the candidate target proposal is further exploited to obtain the candidate's position in the floor map. We use this information to build a probability distribution over all the candidate locations of the target object. The policy is learned online by Monte Carlo simulations, therefore the proposed framework is general and easy to deploy in any environment. The *POMCP exploration* terminates when one location within the belief space exceeds a threshold.

Crucially, our approach exploits the model of the environment to consider the sensor's field of view and all the admissible moves of the agent in the area. For our active visual search scenario, such a model can be easily obtained by building a map of the environment to include the position of fixed elements, such as obstacles, walls or furniture. Our motion policy explicitly exploits the knowledge of the environment for the visibility modelling, instead other RL-based strategies [10], [11] implicitly encode such environment knowledge in a data-driven manner. Once the exploration phase is over, the *probabilistic docking* module guides the agent to approach the target location –i.e., the closest pose with a frontal-facing viewpoint to the target– as quickly as possible. First, we estimate the shortest path [14] on the graph of all possible robot poses, and then a path replanning is used to improve robustness.

With respect to our previous work [13], the main contributions we make in this paper are:

- a new strategy for improving the robustness with respect to false positives and miss-detection when using a real object detector in which we substitute the deterministic detection with a probabilistic one through a Bayesian inference considering a probability distribution over all possible object locations;
- a new strategy for the belief update of POMCP that allows us to lower the total path length of the exploration and increase the effectiveness with large state-space environments;

- a new approach for docking, considering the information gathered during the exploration to improve the robustness to the problems discussed above;
- a deeper experimental analysis and results discussion, providing results for all the scenarios of the Active Vision Dataset;
- an extended description of the POMCP visual search method, with more mathematical details and discussion.

II. Related Work

The two main research topics related to this work are AVS and planning with Partially Observable MDPs. The main works of both topics are briefly surveyed in the following and original elements of our contribution with respect to the state-of-the-art highlighted.

A. Active Visual Search

Active Visual Search, often referred to as Object Goal Navigation, is a specific task of Embodied AI research field. Embodied AI, which learns through interactions with the environment from an egocentric perspective, is an emerging field of study. Within this field, AVS is a task focused on detecting and approaching a specific object [15]. Early approaches exploit intermediate objects –e.g., the relation between a sofa and a television– to restrict the search area for the target object. Although intermediate objects are usually easier to detect because of their size, their spatial relation w.r.t.the target may be not systematic. A probabilistic approach is proposed in [16], where the likelihood of the target increases when objects which are expected to be co-occurring are detected.

AVS with deep learning is viable using Deep Reinforcement Learning techniques [10], [11], [17], where visual neural embeddings are often exploited for action policy training. Han et al. [17] proposed a deep Q-network (DQN) where the agent state is given by CNN features describing the current RGB observation and the bounding box of the detected object. However, this work assumes that the object must be detected initially. To address the search task, EAT [10] performs feature extraction from the current RGB observation, and the candidate target crop generated by a region proposal network (RPN). The features are then fed into the Action Policy network. Similarly, GAPLE [11] uses deep visual features enriched by 3D information, from the depth channel, for policy learning. Although GAPLE claims to be generalised, expensive training is the cost to pay as GAPLE is trained with 100 scenes rendered using a simulator House3D based on the synthetic SUNCG dataset. This limitation is shared with other approaches that learn optimal policies using Asynchronous Advantage Actor Critic (A3C) algorithm [18], Long Short Term Memories (LSTM) architectures [19], and Transformer networks coupled with deep Q-Learning [20].

Recent efforts from the community include also benchmarking the AVS task. Challenges including Habitat ObjectNav [21] encourage methods for enabling an agent initialised at a random starting pose in an *unknown* environment to find a given instance of an object category using only sensory inputs to navigate, where large-scale datasets of 3D real-world spaces with densely annotated semantics are also made available to facilitate training and testing the models [22]. As the scene map is unknown, most methods aim to learn the policy by joining the objectives of both semantic exploration and object search [23]. On top of such semantic exploration, the perception skills in terms of where to look and the navigation can be further disentangled [24] for an improved success rate. Moreover, spatial relations among objects have also been formulated as graphs and embedded via Graph Convolutional Networks to guide the navigation policy [25], where external commonsense knowledge has also shown advantages for the object localisation via spatial graph learning [3]. In general, RL-based strategies are dependent on training with a large amount of data in order to encode the environmental modelling and motion policy. Differently, our proposed POMCP-based method makes explicit use of the available scene knowledge and performs efficient planning for the agent's path online without additional offline training.

B. Monte Carlo Planning

As for optimal policy computation, *Partially Observable Markov Decision Processes* (POMDPs) are a popular framework for representing dynamical processes in uncertain environments and solving related sequential decision making problems [26]. Computing exact solutions for non-trivial POMDPs is often computationally intractable [27], but in the recent years impressive progress was made to develop approximate solvers. One of the most recent and efficient strategies for solving POMDPs in an approximate way is *Monte Carlo Tree Search* (MCTS) [28], [29], [30]. The main advantage of using MCTS for solving POMDPs is scalability. MCTS-based strategies compute the policy online, i.e., only for the specific states (or beliefs, in case of partially observable environments) the agent visits in its trajectories. This is fundamental in domains with very large state spaces and in partially observable environments where the dimension of the belief space is infinite, since beliefs are probability distributions over states. In MCTS system states are represented as nodes of a tree, and actions/observations as edges. Monte Carlo simulations are performed to generate the tree using specific action selection strategies, such as the algorithm called Upper Confidence bounds applied to Trees (UCT) [31], that efficiently balances exploration and exploitation.

The most influential solver for POMDPs which takes advantage of MCTS is *Partially Observable Monte Carlo Planning (POMCP)* [12] which combines a particle filter representation of the belief, a MCTS-based strategy for computing action Q-values, and an efficient way to update the agent's belief. Several extensions of POMCP have been realised. BA-POMCP [32] extends POMCP to Bayesian Adaptive POMDPs, allowing the model of the environment to be learned during execution. A version of POMCP for scalable planning in multiagent POMDPs is presented in [33], which it introduced model learning in POMDPs considering also the uncertainty about model parameters in the belief. A scalable extension of POMCP for dealing with cost constraints is presented in [34]. Very recent work focused on the introduction of prior knowledge about the environment and the policy in POMCP. In [35] known state-variable relationships are used to improve the performance of POMCP, in [36] policy improvement is performed with safety guarantees. In [37] logical rules representing parts of the POMCP policy are generated using Satisfiability Modulo Theory (SMT) to improve the explainability of the policy and identify anomalous action selections due to wrong parameter tuning. Again, in the research line of merging probabilistic planning and symbolic

approaches, [38], [39] allows to generate shields based on logical rules to improve the safety of POMCP. The technique has been further improved in [40], [41] where active approaches and methods based on Inductive Learning of Answer Set Programs are used to learn the logic rules. In [42], the authors presented a probabilistic technique for solving the learning motion planning problem in static environments. A dynamic probabilistic environment is considered in [43], incorporating perception uncertainty and incompleteness into the planning process through a probabilistic approach.

Applications of POMCP can be found in several domains. A few of them are related to the exploration of partially known environments [44] and the find-and-follow of people [45] with robots. Others [46], [47] concern robot navigation using only POMCP or hierarchical methods approaches with POMCP for high-level control and neural networks for low-level control. Popular MCTS-based approaches have been recently used also for developing agents with superhuman performance in the game of Go [48], [49]. The approach proposed in this work differentiates from all works mentioned above because it specialises POMCP to the AVS domain and introduces methodological improvements to belief update, probabilistic detection of objects and docking that are not present in the literature. To the best of our knowledge, the only works available in the literature about AVS with POMCP are [13], [50]. The differences between these works are substantial since [13] uses standard belief update, assumes exact object detection and employs a naïve docking procedure, while [50] focuses on completely unknown environments.

III. Method

We consider an agent navigating through known environment with the goal of locating and approaching a specific object. The agent explores the environment to identify the target object, determines its location on the map and then moves closer to it. To be coherent with the related literature, the agent's *pose* at time step t is $p_t = \{x_t, y_t, \theta_t\}$, where x and y are the coordinates on the floor plane, and θ is the orientation. At each time step the agent takes an action a_t from a predefined set A: specifically, the agent can `move_forward`, `move_backward`, `rotate_clockwise`, `rotate_counter_clockwise`. Rotations are defined by a fixed angle. When the agent reaches a new pose p_t, it receives an observation which is the output of an object detector applied to the image acquired by an RGB-D camera. We model the search space as a grid map (see Fig. 1(b)), in which each cell can be either: *i)* "*visual occlusion*", if the cell is occupied by obstacles, such as a wall or a piece of furniture, that prevent the agent to see through; *ii)* "*empty*", if the agent is allowed to enter the cell and thus no objects can be located in there; or *iii)* "*candidate*", if none of the above, thus the cell is a possible object location for the target object.

A. Partially Observable Markov Decision Processes

We formulate the AVS problem as a Partially Observable Markov Decision Process (POMDP), which is a standard framework for modeling sequential decision processes under uncertainty in dynamical environments [26]. A POMDP is a tuple $(S, A, O, T, Z, R, \gamma)$, where S is a finite set of partially observable states, A is a finite set of actions, Z is a finite set of observations, T: $S \times A \to \Pi(S)$ is the *state-transition model*, O: $S \times A \to \Pi(Z)$ is the *observation model*, R: $S \times A \to \mathbb{R}$ is the reward function and $\gamma \in [0, 1)$ is a discount factor. Agents operating POMDPs aim to maximise their expected total discounted reward $E[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)]$, by choosing the best action a_t in each state s_t, where t is the time instant; γ reduces the weight of distant rewards and ensures the (infinite) sum's convergence. The partial observability of the state is modelled by considering at each time-step a probability distribution over all the states, called the *belief* B. POMDP solvers are algorithms that compute, in an exact or approximate way, a *policy* for POMDPs, namely a function π: $B \to A$ that maps beliefs to actions.

B. Partially Observable Monte Carlo Planning

POMCP [12] is an online Monte-Carlo based solver for POMDPs. It uses Monte-Carlo Tree Search (MCTS) for selecting, at each time step, an action which approximates the optimal one. Given the current belief, represented by an unweighted particle filter, the Monte Carlo tree is generated by performing a certain number of simulation from the current belief. These simulations generate, in an efficient way, estimates of the Q-values of all actions from the current belief. The action with the highest estimated Q-value is selected and performed in the environment. A big advantage of POMCP is that it enables to scale to large state spaces because it never represents the complete policy but it generates only the part of the policy related to the belief states actually seen during the plan execution. Moreover, the local policy approximation is generated online using a simulator of the environment, namely a function that given the current state and an action provides the new state and an observation according to the POMDP transition and observation models.

The main phases of the POMCP algorithm are summarized in the following. *i) Particle initialization.* POMCP starts with a MCTS containing only the root node representing the empty history h (i.e., no action performed and no observation observed). The belief state is represented by a particle filter initialized with a certain number of particles. The particles in the root node are initialized by a procedure that selects random hidden states (e.g., target object positions in our application domain) from a uniform distribution over all possible hidden states; *ii) Simulations and node statistics update.* At each step, POMCP performs a certain number of simulations from the current history h to generate (online) a policy for that specific step (i.e., belief). A particle, representing a state s of the system, is randomly chosen from the particle filter of node h which represents the belief state of the agent. From state s a set of simulation steps[1] is performed. At each step, an action a is selected using UCT when the current history is inside the tree, and a *uniform policy* when the current history is outside the tree. A black-box model $\mathcal{M}(s, a)$ is used to perform each simulation step, returning a simulated observation and a simulated reward. When all simulation steps are performed, the total reward of the simulation is used to update *node statistics* about the average return of all simulations passing through h; *iii) Action selection in the environment.* The action that maximizes the Q-value of the initial node h is selected and performed in the environment; *iv) Belief update.* The observation o obtained from the environment after performing action a is

[1] The term *step* is used to identify steps in the environment; the term *simulation step* to identify steps in the simulation phase.

used to update the belief. In particular, the next history node $h' = hao$ is selected in the tree with the related particle filter, and the rest of the tree is pruned; *v) Particle reinvigoration.* If a lack of particles is experienced in h' (because some particles moved to other branches of the tree rooted in h), then new particles are generated in h' by computing local transformations on current particles and using a rejection sampling strategy to decide if the new particles are compatible with the belief in h'. These particles must contain states reachable from the previous belief h after performing action a and observing observation o.

C. Exploration, Localisation and Approach

The methodology here proposed is a specialisation of POMCP for the AVS problem. It is based on three main elements, defined in the following, that are used altogether by POMCP to perform the search of an object in the environment. We assume that n is the number of possible poses that the agent can assume in the environment, m is the number of objects in the environment, and k is the number of locations in which each object can be positioned.

i) The first element of the proposed framework is a *pose graph* $\mathcal{G}$ in which nodes represent the n possible poses of the agent and edges connect only poses reachable by the agent with a single action. Thus, $\mathcal{G}$ is used to constrain the actions that cannot be performed in the environment. *ii)* The second element is the set $\mathcal{H} = \{1, \ldots, k\}$ of all possible indices of locations that each object can take in the environment. Each index in $\mathcal{H}$ corresponds to a specific position in the topology of the environment where the search is made. *iii)* The third element is a matrix of object observability $\mathbf{L} = (l_{i,j}) \in \{0, 1\}^{n \times k}$, where $l_{i,j} = 1$ if the location j is visible from pose i –i.e., location j is in the field of view of the agent positioned in i. Matrix $\mathbf{L}$ can be deterministically derived from $\mathcal{G}$ and $\mathcal{H}$ by a visibility function f_L which computes the visibility of each object from each agent pose, considering the physical properties of the environment. This matrix is used in the observation model employed in the simulations. Namely, the observation model returns 1 if the target object is observed from the current pose, 0 otherwise. More formally, it returns 1 if the agent is in position $\hat{i} \in \mathcal{G}$ and the target object is in a position $\hat{j} \in \mathcal{H}$ for which $l_{\hat{i},\hat{j}} = 1$. Notice that the position of the target object in each specific simulation is known because it is defined in the particle sampled at the beginning of the simulation.

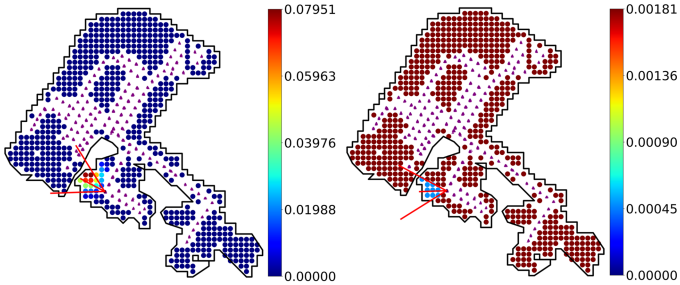
On the other hand, observations in the environment are based on the object detector. Both for the environment and in the simulator, we give a positive reward if the object is observed; otherwise, a negative reward is provided (corresponding to the energy spent to perform the movement) and the POMCP-based search is continued. To prevent the agent to visit the same poses more than once, the agent maintains an internal memory vector that collects all the poses already visited during the current run. Every time the agent re-visits a pose it receives a high negative reward. After every step in the environment, the agent receives from the object detector an observed value 1 if the target object has been observed, 0 otherwise.

The belief of the agent at each time step is an approximated probability distribution over all the candidate object locations in the environment, that represents the POMCP hidden state. If the object is not observed within a fixed amount of moves, the method terminates and reports a search failure.

1) Belief Update: In our original formulation of POMP [13], belief is updated using the standard POMCP strategy. A problem with this approach is related to the cardinality of our state space. In AVS, the state space describes both the agent's pose and the target's location. Because the object can be in any location, and the number of simulations is limited, it may happen that some states are not considered during the simulation phase and can only be recovered during reinvigoration. If they are not recovered, they are removed from the belief and cannot be recovered anymore, even if they are valid positions. Another issue with this approach is that, during reinvigoration, the new particles are sampled from the previous belief. This creates a feedback loop in which particles that survive the belief update, have a higher chance of being chosen during reinvigoration. Thus, in situations where the number of simulations is limited, it is possible for the belief to become confined to a sub-space within the state space. In POMP-BE-PD, we change the belief update and resampling procedure to overcome these issues. The belief is initially generated by sampling particles from a uniform distribution over all states –i.e., candidate object locations. An auxiliary variable pp stores the current list of object locations not been observed yet, where $pp = \{j \in \mathcal{H} \mid j \text{ not yet observed}\}$. The set is initialised as $pp = \mathcal{H}$. At each time step, the agent acquires observations about the locations within the current FOV through the object detector, and it updates pp removing the observed positions that do not contain the searched object. The new belief is sampled from a uniform distribution over locations satisfying the pp constraint –namely, having the searched object in positions belonging to $\mathcal{H} \setminus pp$. This way to update the belief is beneficial in terms of performance, as shown in our experiments below.

Notice that the belief update strategy used in POMP (which is inherited by the standard strategy used in POMCP [12]) is based on rejection sampling [51]. This strategy is known to be suboptimal in large state spaces [52]. In fact, rejection sampling in POMP reinvigoration considers only the states related to the ones in the belief of the previous step. Local transformations of those states are generated and only new states compatible with the previous belief are kept in the particle filter. However, during the reinvigoration process, the random nature of the sampling procedure can lead to losing certain particles' belief (i.e., possible object positions). To cope with this issue, alternative sampling strategies were designed [52], whose effectiveness was measured by empirical analysis (since general mathematical analyses are nontrivial, being the relationship between sampling and outcomes very complicated). For example, in [52], importance sampling [51] is used instead of rejection sampling. Our new approach, instead, re-samples at each step from a set of states explicitly considering all possible positions of the object *not* observed so far. This novel sampling mechanism, which is not based on state transformations but on the explicit consideration of all possible positions not yet observed, gives a general improvement in most of the experiments. The rationale is that the agent considers all possible unobserved object positions to select an optimal action, instead of focusing on the positions that are related to previously visited states.

2) Probabilistic Detection: We equipped our agent with the *Target Driven Instance Detector* (TDID) presented in [53], an architecture designed to recognise and classify specific instances of object classes. Given an image, TDID returns a list of coordinates representing the associated bounding box (if any), a



(a) Object inside FOV (b) Object outside FOV

Fig. 3. The two cases considered when creating the vector D. Example derived from Home_003_2. In case (a) the objective is to determine the location of the object and assign probabilities in the form of a multivariate normal distribution. In (b), we assign low probabilities to the locations inside the FOV, and high probabilities to the locations outside it. Note: we assign different scales to the colorbar for ease of visualisation.

score $s \in [0, 1]$ and the corresponding class c. In our work, we consider only detections with an associated score greater than 0.9. Moreover, given the rate of *TP* (True Positive), *FP* (False Positive) and *FN* (False Negative), we define:

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

On top of that, we define the F_1 score as:

$$F_1\text{-score} = 2\,\frac{Precision \times Recall}{Precision + Recall}\,,$$

where $F_1 \in [0, 1]$ can be interpreted as the harmonic mean of $Precision$ and $Recall$.

In POMP [13], the planner terminates the exploration phase when the object detector identifies the target object inside the FOV. As a consequence, even a single false positive would terminate the exploration, thus not bringing the agent to the target object. In POMP-BE-PD we aim to reduce the impact of this problem by considering the current observation, as well as incorporating the whole history of observations.

We first define a vector $\mathcal{D} = \{d_1, \ldots, d_k\}$ for the probabilities to have the target object in location j considering the output of the object detector with the observation at the *current time step*. In other words, at each step in the environment, we reset all d_j with $j = 1, \ldots, k$. We then consider two cases: *i)* if the object is found by the object detector inside the current FOV, we set $d_j = 0$ in locations j outside the FOV and we set the probabilities d_j of locations inside the FOV according to a multivariate normal distribution with mean in the location j where the object is localised by the detector (see Fig. 3(a)); *ii)* instead, if the object is not found by the object detector inside the FOV, then we set $d_j = F_1$ in locations j inside the FOV and $d_j = 1 - F_1$ in locations j outside the FOV (see Fig. 3(b)). In both cases, we normalise $\mathcal{D}$ so that $\sum_{j=1}^{k} d_j = 1$. Notice that F_1 is class specific, i.e., it accounts for the performance of the object detector for the specific object class.

As a second auxiliary data structure, we define a vector $\mathcal{R} = \{p_1, \ldots, p_k\}$ of probabilities to have the target object in location j considering *the whole history of observations*, i.e., this represents a global probability using information also from previous steps. We initialize a uniform probability at time $t = 0$ as $p_j^0 = 1/n$, where n is the number of candidate object locations. For all the subsequent time steps $t \geq 1$, the probability is updated according to the following rule:

$$p_j^t = \frac{p_j^{t-1} \cdot d_j^t}{\sum_{i=1}^{k} p_i^{t-1} \cdot d_i^t} \tag{1}$$

for all $j \in \mathcal{H}$. Finally, we define a threshold $\tau = \frac{c}{n}$, where $c \in \mathbb{N}$ is a constant that allows us to increase the confidence of our probabilistic detection. We terminate the POMCP exploration phase when in the FOV of the current pose we have an object location j whose probability p_j exceeds the threshold. More formally, the following exit condition must be verified:

$$(p_j \geq \tau) \wedge (L_{i,j} = 1). \tag{2}$$

Updating the probabilities in a Bayesian way (see (1)) falls into the general case of Bayesian inference, in which the parameters of a distribution are estimated by considering subsequent observations of the environment. Our formulation does not assume any specific form of the probability distribution, thus the parametrization is the distribution itself, i.e., the values of the *pdf* in each potential location of the object. In this setup, Bayesian inference is proved to be optimal in the sense that it guarantees to minimize the overall risk of making incorrect decisions. According to this procedure, if the object is not in the current FOV, we assume that it must be in some other location, thus we increase the corresponding probabilities. Instead, if it is in the current FOV, we increase the probabilities of the locations near the 3D position of the object and lower the other ones. Moreover, we do not rely merely on the object detector output, we rather accumulate knowledge over time by leveraging the old and current state of the environment. In Fig. 4 we report an episode in which we can appreciate the evolution of the probabilities inside an environment.

3) Probabilistic Docking: Given the object location $j \in \mathcal{H}$ satisfying the exit condition of (2), we first identify the *destination pose* –i.e., the agent's pose $\hat{i} \in \mathcal{G}$ that is closest to the target location and points towards it. Then we use the Dijkstra algorithm [14] to compute the shortest path between the current pose $i \in \mathcal{G}$ and the estimated destination pose $\hat{i} \in \mathcal{G}$. While the agent navigates towards the destination pose, the object detector is not used since we are confident enough that the target object is in location j. This strategy achieves better performance than the *Robust Visual Docking* introduced in [13]. A key distinction is that in Robust Visual Docking the object detector is used along the path, thus, in case of poor performing detectors, miss-detections and false positives can easily distract the agent from the final goal, having fatal consequences in the approaching phase.

IV. Experiments

We tested our approach on the Active Vision Dataset Benchmark [9], a public benchmark for active visual search that contains more than 30,000 RGBD images taken in 15 different indoor environments and 33 different target objects. Consistently with [10], we classify each scene in the dataset as simple, medium, or hard for the visual search task, where we define a simple environment consisting of a single small room, a medium difficult apartment with a large room or with an additional small room –e.g., a bathroom or an open space–, and finally a hard apartment with multiple large rooms. We use in our experiments two simple (Home_005_2 and Home_015_1), three medium (Home_001_2, Home_016_1, Home_014_2), and three

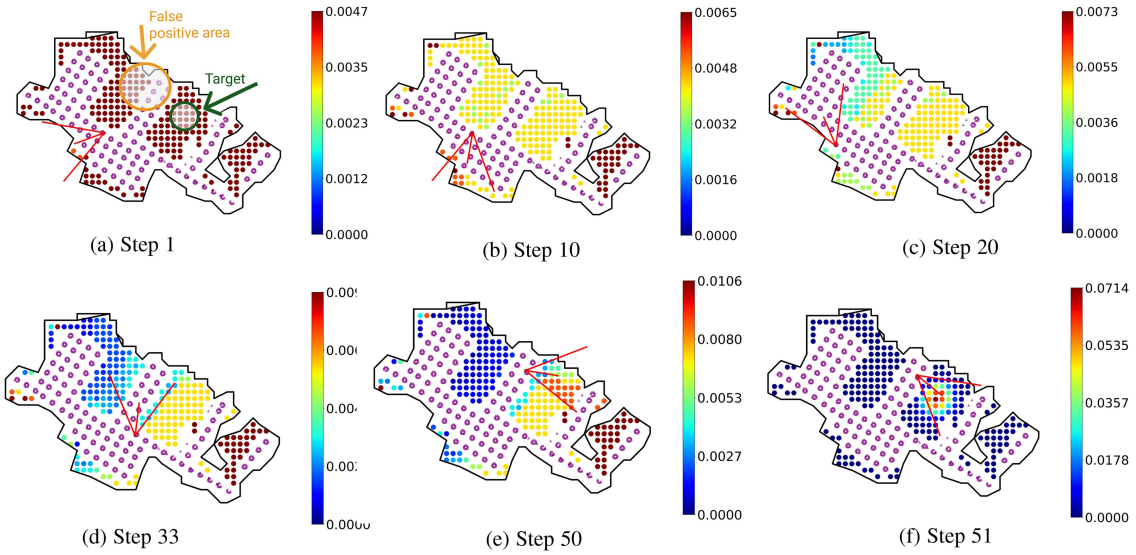


Fig. 4. Evolution of the probabilities p_j inside Home_016_1 using the proposed approach POMP-BE-PD. In step (a) we initialise the agent in the environment; we highlight the target position and a false positive area. From step (b) to (c) the robot explores the top area; in step (d) we show the robustness of our approach to a false positive; finally, in step (e) we identify high probable locations, locating the target in step (f).

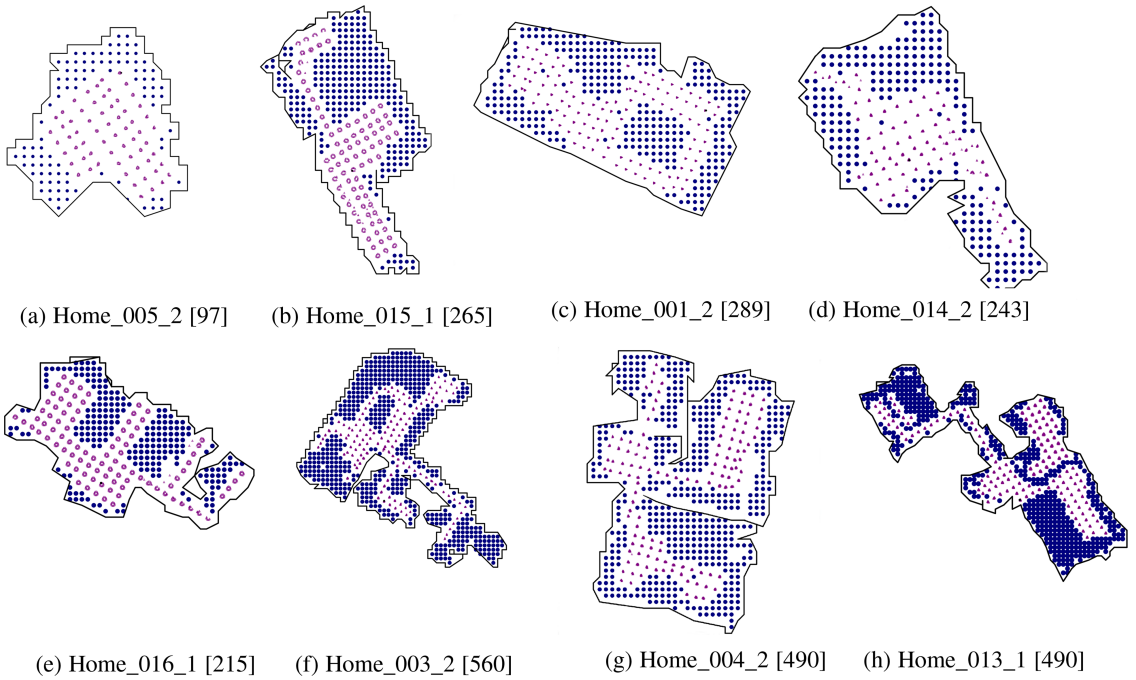


Fig. 5. Corresponding 2D floor maps (not in scale) for the test scenes from AVBD of 3 different difficulty levels (as in [10]). For each environment, we report the name and, in parenthesis, the number of possible object locations. As the difficulty increases, we can note an increment of possible object location and more difficult spatial layouts.

hard apartments (Home_003_2, Home_004_2, Home_013_1). Some examples of these scenarios are shown in Fig. 5.

We consider three metrics: *Success Rate* (SR) [56] is considered the main metric of this work, and it is defined as the percentage of times the agent successfully reaches one of the destination poses (as provided in AVDB) over the total number of trials (a larger value indicates a more effective search); *Average Path Length* (APL) defined as the total number of poses visited by the agent, among the successful episodes, divided by the number of successful episodes (a lower value indicates a higher absolute efficiency); and *Success weighted by Path Length* (SPL) [56] defined as:

$$SPL = \frac{1}{N}\sum_{i=1}^{N} S_i \frac{l_i}{\max(p_i, l_i)}, \tag{3}$$

where N are the test episodes, l_i is the length of the shortest path between the goal and the target for an episode, p_i is the length of the path taken by an agent in an episode and S_i is a binary indicator of success in episode i . In general, a larger value indicates a higher absolute efficiency. In this work, the term "efficiency" refers to the effectiveness of the agent's exploration strategy, aimed at finding the target goal with the shortest path

TABLE I
RESULTS ON THREE SCENES FROM AVDB USING GT OBJECTS ANNOTATIONS

Method	Easy (Home_005_2)			Medium (Home_001_2)			Hard (Home_003_2)			Avg.		
	SR ↑	**APL ↓**	**SPL ↑**	**SR ↑**	**APL ↓**	**SPL ↑**	**SR ↑**	**APL ↓**	**SPL ↑**	**SR ↑**	**APL ↓**	**SPL ↑**
Random Walk	0.32	74.00	0.06	0.11	74.48	0.02	0.10	79.27	0.02	0.18	75.91	0.03
EAT [10]	0.77	12.20	0.42	0.73	16.20	0.56	0.58	22.10	0.41	0.69	16.80	0.46
DQN(*) [54]	*1.00*	*11.06*	-	*0.69*	*18.15*	-	-	-	-	-	-	-
DQN-TAM(*) [55]	*0.98*	*17.85*	-	*0.60*	*24.19*	-	-	-	-	-	-	-
POMP [13]	0.98	13.60	0.71	0.73	17.10	0.58	0.56	20.50	0.40	0.76	17.07	0.56
POMP-BE-PD	**0.98**	11.93	0.71	**0.80**	17.86	0.60	**0.92**	24.52	0.58	**0.90**	18.10	0.63

All methods are compared using the protocol defined in [10]. The asterisk (*) indicates that the evaluation is performed on a different subset of objects.
The bold values indicate state-of-the-art (SOTA) results considering the main metric.

possible. An episode is considered successful if the agent reaches the destination pose given by AVDB in a fixed number of steps (200 in our experiments), using the initial pose definition given by [9].

A. Quantitative Results

We compare our proposed approach POMP-BE-PD against a random walk baseline –i.e., we allow the agent to randomly select an action among all the feasible ones at each time step–, and four state-of-the-art approaches, namely: EAT [10], DQN [54], DQN-TAM [55], and our previous work POMP [13]. The latter is the only unsupervised method, while the former three need training data to learn the policy. Since no official code for published methods is available, except for EAT, we are only able to compare results with them following the protocol proposed in [10]. With respect to the standard protocol defined in the benchmark paper [9], this protocol provides results only using GT annotations for object detection, and on a limited number of scenes. Moreover, it uses only a subset of target objects. DQN and DQN-TAM use only two scenes (one easy and one medium), thus the average column is not meaningful for a fair comparison. Additionally, DQN and DQN-TAM use a different subset of objects in their evaluation. From results reported in Table I we can clearly see that our approach POMP-BE-PD outperforms EAT in terms of SR, with a little increment in APL, which is reasonable since we are now considering more challenging situations, as we will deeply explore in Section IV-C. As for the comparison between POMP-BE-PD and DQN, we note that the DQN approach outperforms our method in the easy scenario, but in the medium case we outperform the competitor in SR with a comparable APL. It is worth noting that for achieving these results both DQN approaches require 13 scenarios for training the best policy, while our method requires no training at all.

Results using the object detector provided by [53] are reported in Table III for POMP and POMP-BE-PD. Again, we can appreciate a strong increment of 35% of both SR and SPL, mostly due to the ability of our proposed method to handle more complex cases.

B. Ablation Studies

We provide also an ablation study to deeply analyse the contributions of the different terms in our model. In the following we will answer some questions by comparing the proposed method with some partial versions of it considering only the new belief update (called POMP-BE) and considering only the probabilistic detection (called POMP-PD).

TABLE II
RESULT OF DIFFERENT VERSIONS OF IMPROVED POMP WITH MORE SCENES PER DIFFICULTY LEVEL IN AVD

Difficulty	Scene	POMP[13]			POMP-BE		
		SR ↑	**APL ↓**	**SPL ↑**	**SR ↑**	**APL ↓**	**SPL ↑**
Easy	Home_005_2	0.94	12.96	0.73	0.93	12.26	0.72
	Home_015_1	0.75	23.66	0.45	0.73	17.04	0.52
	Avg.	**0.84**	18.31	0.59	0.83	14.65	0.62
Medium	Home_001_2	0.80	18.20	0.57	0.81	19.95	0.55
	Home_014_2	0.76	41.07	0.38	0.90	19.99	0.55
	Home_016_1	0.71	29.64	0.39	0.83	36.55	0.50
	Avg.	0.76	29.64	0.45	**0.85**	25.50	0.53
Hard	Home_003_2	0.43	21.90	0.27	0.79	31.93	0.45
	Home_004_2	0.45	66.20	0.17	0.57	47.71	0.28
	Home_013_1	0.55	49.72	0.27	0.74	53.11	0.41
	Avg.	0.48	45.94	0.24	**0.70**	44.25	0.38
Average		0.67	32.92	0.40	**0.79**	29.82	0.50

POMP-BE is POMP with the improved Belief Update. Result using the ground truth annotations instead of the detector, using 2^{10} simulations during the planning phase. The new Belief Update consistently increase the efficiency of the exploration phase, thus reducing the Average Path length, and increasing the SR and SPL.
The bold values indicate state-of-the-art (SOTA) results considering the main metric.

Belief update: Does the new belief update reduce the episode length? What are the benefits of the new belief update when navigating difficult scenarios?

In Fig. 6 we aggregate the episodes by their difficulty, grouped by the minimum path length for the episode to reach the target. In Fig. 6(a) we aggregated the results for the easy scenario (Home_005_2 and Home_015_1); in Fig. 6(b) for the hard scenario (Home_003_2, Home_004_2, Home_013_1); in Fig. 6(c) for the medium one (Home_001_2, Home_014_2 and Home_016_1); finally for Fig. 6(d) we aggregated the results for all the scenario present in AVDB. From these charts we can derive that, by removing from the belief update locations already observed, we can optimize the exploration phase, thus increasing the effectiveness. Furthermore, Table II analyzes the impact of the new belief update isolating all the possible causes of error, i.e., we swap the object detector with the ground truth annotations eliminating the source of false positive and miss detection both during the planning and docking phases. In the easy scenario, we have a marginal reduction in SR (0.01) while reducing the APL and increasing the efficiency (i.e., SPL). We hypothesise that the simple layout (Fig. 5(a) and (b)) and the concentration of minimum path length (Fig. 6(a)) do not allow the belief update to be beneficial. However, the more difficult the scenario, with more possible object locations and complicated spatial layouts, the higher the improvement in performance.

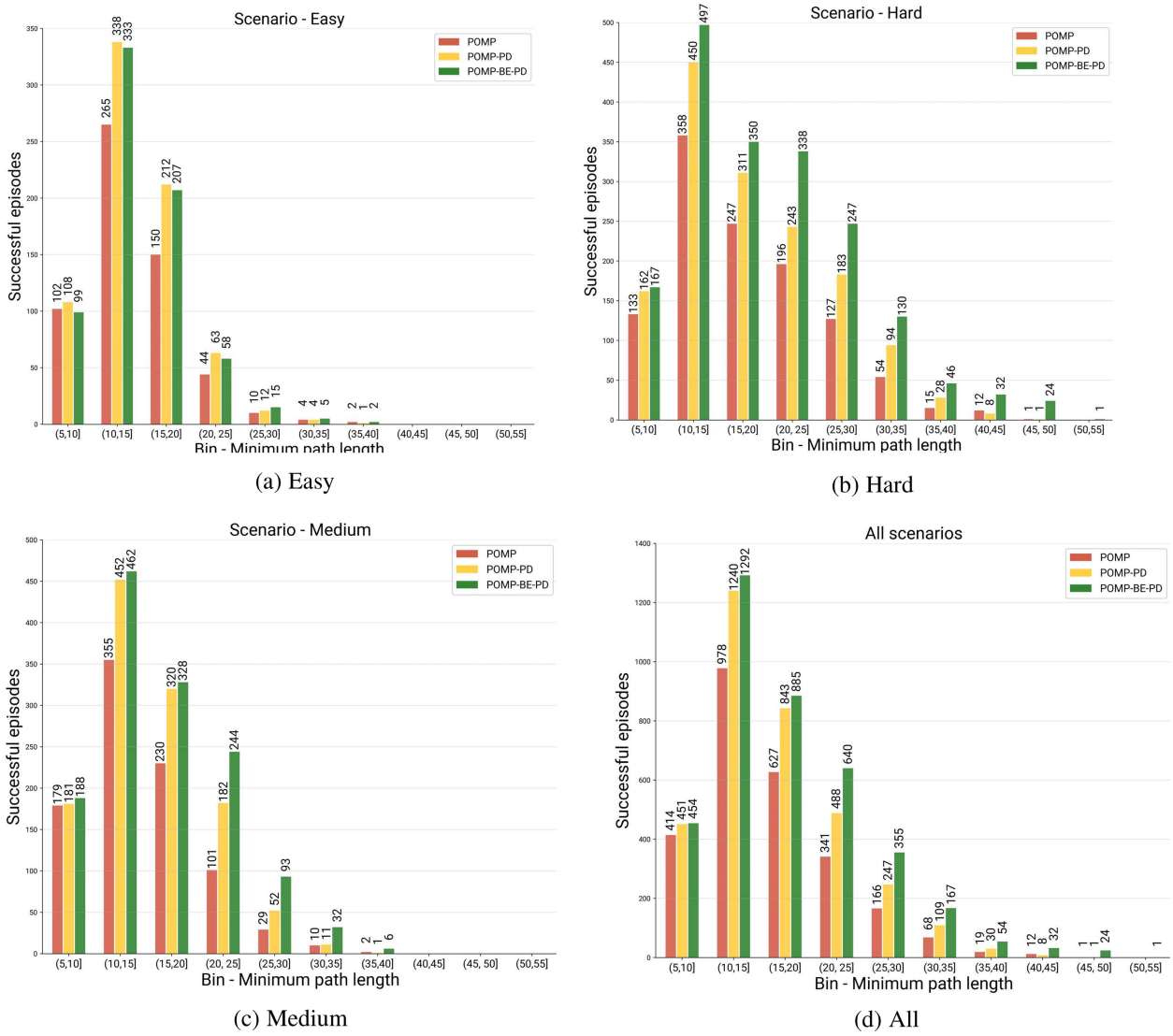


Fig. 6. We aggregated the episodes by the minimum number of steps to reach the object, thus incorporating the difficulty of the episode. In Figure (a) the results for the Easy scenarios; in Fig. (b) Hard Scenarios; in Fig. (c) the Medium ones and finally, in Fig. (d), the sum of all scenarios. Results using the object detector provided by [53], both during planning and docking. Focusing on the POMP-PD method (yellow bar), we can observe the increment of efficiency and efficacy due to the introduction of the Belief Update (green bar), since both methods do not change the exit condition during planning (Probabilistic Detection).

Starting from the medium scenario, we increase the SR from 0.76 to 0.85 while decreasing the APL from 29.64 to 25.50, with an increment of the total efficiency from 0.45 to 0.53. A similar result can be seen for the hard scenario.

Probabilistic Detection & Docking: Does Probabilistic Detection reduce the number of false positives? Is there a way to improve the docking, also considering the knowledge gathered during the planning?

To answer the first question, we analyze the different types of failure that can occur during the episodes. More specifically, we consider three types of errors: *i) Localisation*, if during POMCP exploration the exit condition is verified, but the target object is not actually present in the FOV. *ii) Docking*, if in the last pose of the POMCP exploration the agent correctly detects the target, but it fails to reach the *successful destination poses* defined by AVDB; *iii) Other*, if the error is not categorized as Docking or Localisation, and if the agent is unable to detect the target object within the time limit or the agent performs action not allowed during the path. In Fig. 7 we provide the percentage of error for each planner, averaged over all scenarios. First of all, we can notice a significant reduction (∼32% decrease) of false positives when introducing the Probabilistic Detection approach, thus increasing the robustness of our method. The new Belief Update, instead, greatly reduces the error categorized as "Other" (∼30% decrease), thus increasing the efficiency of our method. Moreover, using the knowledge gathered during the planning provides a reliable mechanism to increase the robustness during the docking phase. Indeed, if we look at Fig. 7 we can appreciate a ∼35.7% decrease of *Docking* error.

To measure the impact of Probabilistic Detection, in Table III we conduct an ablation study isolating the Probabilistic approach from the belief update, both using the detector during the planning and docking. In all the scenarios, POMP-PD increments the SR by a large margin (19% - 25%) over our previous formulation POMP, while maintaining, on average, the same SPL. However, we note an increment of the APL. This is not surprising: indeed,

TABLE III
RESULTS OF POMP AND VARIATIONS OF POMP-BE-PD WITH MORE SCENES PER DIFFICULTY LEVEL IN AVD USING THE REAL DETECTOR PROVIDED BY [53]

Difficulty	Scene	POMP[13]			POMP-BE			POMP-PD			POMP-BE-PD (Proposed)		
		SR ↑	APL ↓	SPL ↑	SR ↑	APL ↓	SPL ↑	SR ↑	APL ↓	SPL ↑	SR ↑	APL ↓	SPL ↑
Easy	Home_005_2	0.60	17.90	0.40	0.58	16.18	0.41	0.81	26.08	0.42	0.79	22.70	0.45
	Home_015_1	0.49	34.76	0.22	0.45	38.76	0.23	0.55	35.34	0.23	0.54	30.50	0.26
	Avg.	0.54	26.33	0.31	0.52	27.47	0.32	**0.68**	30.71	0.33	0.67	26.60	0.35
Medium	Home_001_2	0.40	20.73	0.24	0.39	19.36	0.24	0.50	31.00	0.24	0.57	28.50	0.31
	Home_014_2	0.53	47.60	0.25	0.60	18.52	0.38	0.60	45.79	0.24	0.66	21.38	0.37
	Home_016_1	0.29	50.23	0.12	0.28	47.05	0.13	0.36	57.73	0.12	0.41	53.26	0.16
	Avg.	0.41	39.52	0.20	0.42	28.31	0.25	0.49	44.84	0.20	**0.55**	34.38	0.28
Hard	Home_003_2	0.19	26.60	0.10	0.33	30.53	0.18	0.39	62.36	0.13	0.48	42.86	0.20
	Home_004_2	0.42	69.84	0.15	0.55	47.31	0.26	0.44	70.26	0.14	0.54	61.93	0.20
	Home_013_1	0.25	61.41	0.12	0.31	77.09	0.14	0.26	62.80	0.09	0.34	54.38	0.15
	Avg.	0.29	52.62	0.12	0.40	51.64	0.19	0.36	65.14	0.12	**0.45**	53.06	0.18
Average		0.40	41.13	0.20	0.44	36.85	0.25	0.49	48.92	0.20	**0.54**	39.44	0.27

The bold values indicate state-of-the-art (SOTA) results considering the main metric.

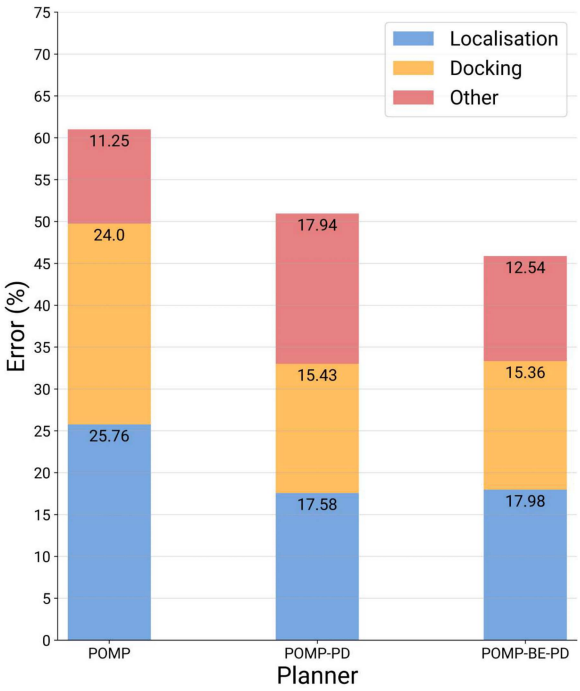


Fig. 7. Percentage of error of POMP, POMP-PD and POMP-BE-PD, averaged over all scenarios. The errors are categorised into three types: Localisation, Docking and Other. We used the object detector provided by [53], during both planning and docking.

if the agent needs to be more confident and robust against false positives, it must require more steps to increase the probability of the target location, and bring that to be $\geq$ the threshold τ.

C. Qualitative Results

In Fig. 4 we visualise an episode by our proposed approach, in which it is possible to appreciate the evolution of the probability distribution over the locations and the robustness to a false positive. The starting pose is defined in Fig. 4(a). From Fig. 4(b) and (c) the agent explores the top part of the environment without success. In Fig. 4(d) the robot encounters a false positive: the update rule defined in (1) with the generated probability map are providing a robust framework for not stopping the exploration. Indeed, the standard POMP defined in [13], in the same situation, would have stopped the episode. Moreover, we can note that in the area unexplored by the agent (the right part of the environment) we are raising the probabilities: if we do not locate the target elsewhere, the object must be in this area. Finally, in Fig. 4(e) we locate a zone with a high probability of containing the target, and in Fig. 4(f) we locate the searched object.

V. CONCLUSION

In this paper we presented POMP-BE-PD, our proposed approach to solve Active Visual Search (AVS) in known environments. Based on a POMCP planner, POMP-BE-PD learns the policy online by efficiently exploiting the information of the 2D floor map of the environment; as a consequence, our method does not require any expensive training, both in time and computational resources. To cope with imperfect object detectors, with a high number of false positives and miss-detection that could have a dramatic effect on the overall success rate, we transitioned from a deterministic detection to a *Probabilistic* one. After every step in the environment, a Bayesian inference, combined with a probability distribution over all possible object locations, allows us to reduce the false positives error by 32%. Consequently, to handle the restricted *belief space* of the original POMCP in the AVS domain, we introduce a new belief update considering, at each time step, all the possible positions that have not been observed yet. We evaluate extensively our method, following the AVDB benchmark, achieving state-of-the-art results. On top of that, with several ablation studies, we demonstrated the strength of our method. On average over all the environments, we increase the success rate by a significant 35% while decreasing the average path length by 4% with respect to our previous formulation POMP.

REFERENCES

[1] J. K. Tsotsos, "On the relative complexity of active versus passive visual search," *Int. J. Comput. Vis.*, vol. 7, no. 2, pp. 127–141, Jan. 1992, doi: 10.1007/bf00128132.
[2] K. Sjöö, A. Aydemir, and P. Jensfelt, "Topological spatial relations for active visual search," *Robot. Auton. Syst.*, vol. 60, no. 9, pp. 1093–1107, Sep. 2012, doi: 10.1016/j.robot.2012.06.001.

[3] F. Giuliari, G. Skenderi, M. Cristani, Y. Wang, and A. Del Bue, "Spatial commonsense graph for object localisation in partial scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 19518–19527, doi: 10.1109/cvpr52688.2022.01891.

[4] F. Taioli, F. Cunico, F. Girella, R. Bologna, A. Farinelli, and M. Cristani, "Language-enhanced RNR-map: Querying renderable neural radiance field maps with natural language," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2023, pp. 4669–4674, doi: 10.48550/arXiv.2308.08854.

[5] J. Park, T. Yoon, J. Hong, Y. Yu, M. Pan, and S. Choi, "Zero-shot active visual search (ZAVIS): Intelligent object search for robotic assistants," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 2004–2010, doi: 10.1109/icra48891.2023.10161345.

[6] A. Rasouli, P. Lanillos, G. Cheng, and J. K. Tsotsos, "Attention-based active visual search for mobile robots," *Auton. Robots*, vol. 44, no. 2, pp. 131–146, Aug. 2019, doi: 10.1007/s10514-019-09882-z.

[7] M. Leslie, "Robots tackle DARPA underground challenge," *Engineering*, vol. 13, pp. 2–4, Jun. 2022, doi: 10.1016/j.eng.2022.04.003.

[8] P. Pouya and A. Madni, "Performing active search to locate indication of ancient water on mars: An online, probabilistic approach," in *Proc. Annu. Conf. ASCEND*, 2021, Art. no. 4024, doi: 10.2514/6.2021-4024.vid.

[9] P. Ammirato, P. Poirson, E. Park, J. Kosecka, and A. C. Berg, "A dataset for developing and benchmarking active vision," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 1378–1385, doi: 10.1109/icra.2017.7989164.

[10] J. F. Schmid, M. Lauri, and S. Frintrop, "Explore, approach, and terminate: Evaluating subtasks in active visual object search based on deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 5008–5013, doi: 10.1109/iros40897.2019.8967805.

[11] X. Ye, Z. Lin, J.-Y. Lee, J. Zhang, S. Zheng, and Y. Yang, "GAPLE: Generalizable approaching policy learning for robotic object searching in indoor environment," *IEEE Trans. Robot. Autom.*, vol. 4, no. 4, pp. 4003–4010, Oct. 2019, doi: 10.1109/lra.2019.2930426.

[12] D. Silver and J. Veness, "Monte-Carlo planning in large POMDPs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 2164–2172.

[13] Y. Wang et al., "POMP: Pomcp-based online motion planning for active visual search in indoor environments," in *Proc. Brit. Mach. Vis. Conf.*, 2020, pp. 1–1, doi: 10.48550/arXiv.2009.08140.

[14] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, Dec. 1959, doi: 10.1007/bf01386390.

[15] D. Batra et al., "ObjectNAV revisited: On evaluation of embodied agents navigating to objects," 2020, *arXiv: 2006.13171*, doi: 10.48550/arXiv.2006.13171.

[16] L. Kunze, K. K. Doreswamy, and N. Hawes, "Using qualitative spatial relations for indirect object search," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 163–168, doi: 10.1109/icra.2014.6906604.

[17] X. Han, H. Liu, F. Sun, and X. Zhang, "Active object detection with multistep action prediction using deep Q-network," *IEEE Trans. Ind. Inform.*, vol. 15, no. 6, pp. 3723–3731, Jun. 2019, doi: 10.1109/tii.2019.2890849.

[18] P. Mirowski et al., "Learning to navigate in complex environments," in *Proc. Int. Conf. Learn. Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=SJMGPrcle

[19] A. Mousavian, A. Toshev, M. Fiser, J. Kosecka, A. Wahid, and J. Davidson, "Visual representations for semantic target driven navigation," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 8846–8852, doi: 10.1109/icra.2019.8793493.

[20] K. Fang, A. Toshev, L. Fei-Fei, and S. Savarese, "Scene memory transformer for embodied agents in long-horizon tasks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 538–547, doi: 10.1109/cvpr.2019.00063.

[21] K. Yadav, "Habitat challenge 2022," 2022. [Online]. Available: https://aihabitat.org/challenge/2022/

[22] K. Yadav, "Habitat-matterport 3D semantics dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 4927–4936, doi: 10.1109/cvpr52729.2023.00477.

[23] D. S. Chaplot, D. Gandhi, A. Gupta, and R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," in *Proc. Neural Inf. Process. Syst.*, 2020, pp. 4247–4258.

[24] S. K. Ramakrishnan, D. S. Chaplot, Z. Al-Halah, J. Malik, and K. Grauman, "PONI: Potential functions for objectgoal navigation with interaction-free learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18890–18900, doi: 10.1109/cvpr52688.2022.01832.

[25] D. A. Sasi Kiran et al., "Spatial relation graph and graph convolutional network for object goal navigation," in *Proc. IEEE 18th Int. Conf. Automat. Sci. Eng.*, 2022, pp. 1392–1398, doi: 10.1109/case49997.2022.9926534.

[26] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, no. 1–2, pp. 99–134, May 1998, doi: 10.1016/s0004-3702(98)00023-x.

[27] C. Papadimitriou and J. Tsitsiklis, "The complexity of Markov decision processes," *Math. Oper. Res.*, vol. 12, no. 3, pp. 441–450, 1987.

[28] S. Thrun, "Monte Carlo POMDPs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 1064–1070.

[29] R. Coulom, *Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search*. Berlin, Germany: Springer, 2007, pp. 72–83, doi: 10.1007/978-3-540-75538-8_7.

[30] C. Browne et al., "A survey of Monte Carlo tree search methods," *IEEE Trans. Comp. Intell. AI Games*, vol. 4, no. 1, pp. 1–43, Mar. 2012.

[31] L. Kocsis and C. Szepesvári, *Bandit Based Monte-Carlo Planning*. Berlin, Germany: Springer, 2006, pp. 282–293, doi: 10.1007/11871842_29.

[32] S. Katt, F. A. Oliehoek, and C. Amato, "Learning in POMDPs with Monte Carlo tree search," in *Proc. 34th Int. Conf. Mach. Learn.*, PMLR, 2017, pp. 1819–1827.

[33] C. Amato and F. Oliehoek, "Scalable planning and learning for multiagent POMDPs," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 1995–2002, doi: 10.1609/aaai.v29i1.9439.

[34] J. Lee, G.-H. Kim, P. Poupart, and K.-E. Kim, "Monte-Carlo tree search for constrained POMDPs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7934–7943.

[35] A. Castellini, G. Chalkiadakis, and A. Farinelli, "Influence of state-variable constraints on partially observable Monte Carlo planning," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 5540–5546, doi: 10.24963/ijcai.2019/769.

[36] A. Castellini, F. Bianchi, E. Zorzi, T. D. Sim ao, A. Farinelli, and M. T. J. Spaan, "Scalable safe policy improvement via Monte Carlo tree search," in *Proc. 40th Int. Conf. Mach. Learn.*, PMLR, 2023, pp. 3732–3756. https://proceedings.mlr.press/v202/castellini23a.html

[37] G. Mazzi, A. Castellini, and A. Farinelli, "Identification of unexpected decisions in partially observable Monte Carlo planning: A rule-based approach," in *Proc. ACM 20th Int. Conf. Auton. Agents Multiagent Syst.*, 2021, pp. 889–897.

[38] G. Mazzi, A. Castellini, and A. Farinelli, "Rule-based shielding for partially observable Monte-Carlo planning," in *Proc. Int. Conf. Automated Plan. Scheduling*, 2021, pp. 243–251, doi: 10.1609/icaps.v31i1.15968.

[39] G. Mazzi, A. Castellini, and A. Farinelli, "Risk-aware shielding of partially observable Monte Carlo planning policies," *Artif. Intell.*, vol. 324, Nov. 2023, Art. no. 103987, doi: 10.1016/j.artint.2023.103987.

[40] G. Mazzi, A. Castellini, and A. Farinelli, "Active generation of logical rules for POMCP shielding," in *Proc. 21th Int. Conf. Auton. Agents Multiagent Syst.*, 2022, pp. 1696–1698.

[41] D. Meli, G. Mazzi, A. Castellini, and A. Farinelli, "Learning logic specifications for soft policy guidance in POMCP," in *Proc. 22th Int. Conf. Auton. Agents Multiagent Syst.*, 2023, pp. 373–381.

[42] P. Svestka and M. H. Overmars, "Motion planning for carlike robots using a probabilistic learning approach," *Int. J. Robot. Res.*, vol. 16, no. 2, pp. 119–143, 1997, doi: 10.1177/027836499701600201.

[43] C. Fulgenzi, A. Spalanzani, and C. Laugier, "Probabilistic motion planning among moving obstacles following typical motion patterns," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St. Louis, MO, USA, 2009, pp. 4027–4033, doi: 10.1109/IROS.2009.5354755.

[44] M. Lauri and R. Ritala, "Planning for robotic exploration based on forward simulation," *Robot. Auton. Syst.*, vol. 83, pp. 15–31, Sep. 2016, doi: 10.1016/j.robot.2016.06.008.

[45] A. Goldhoorn, A. Garrell, R. Alquezar, and A. Sanfeliu, "Continuous real time POMCP to find-and-follow people by a humanoid service robot," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 741–747, doi: 10.1109/humanoids.2014.7041445.

[46] A. Castellini, E. Marchesini, and A. Farinelli, "Partially observable Monte Carlo planning with state variable constraints for mobile robot navigation," *Eng. Appl. Artif. Intell.*, vol. 104, Sep. 2021, Art. no. 104382, doi: 10.1016/j.engappai.2021.104382.

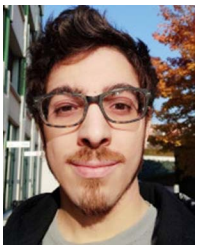
[47] M. Zuccotto, M. Piccinelli, A. Castellini, E. Marchesini, and A. Farinelli, "Learning state-variable relationships in POMCP: A framework for mobile robots," *Front. Robot. AI*, vol. 9, Jul. 2022, Art. no. 819107, doi: 10.3389/frobt.2022.819107.

[48] D. Silver et al., "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016, doi: 10.1038/nature16961.

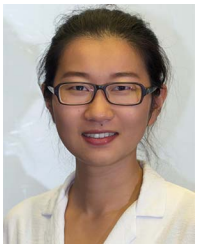
[49] D. Silver et al., "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, Dec. 2018, doi: 10.1126/science.aar6404.

[50] F. Giuliari et al., "POMP++: Pomcp-based active visual search in unknown indoor environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 1523–1530, doi: 10.1109/iros51168.2021.9635866.
[51] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Germany: Springer-Verlag, 2006.
[52] S. Katt, F. A. Oliehoek, and C. Amato, "Bayesian reinforcement learning in factored POMDPs," in *Proc. 18th Int. Conf. Auton. Agents MultiAgent Syst.*, 2019, pp. 7–15, doi: 10.5555/3306127.3331668.
[53] P. Ammirato, C.-Y. Fu, M. Shvets, J. Kosecka, and A. C. Berg, "Target driven instance detection," 2018, *arXiv: 1803.04610*, doi: 10.48550/arXiv.1803.04610.
[54] S. Liu and G. Tian, "A high-efficient training strategy for deep Q-learning network used in robot active object detection," in *Proc. IEEE 12th Int. Conf. CYBER Technol. Automat. Control Intell. Syst.*, 2022, pp. 202–206, doi: 10.1109/cyber55403.2022.9907473.
[55] S. Liu, G. Tian, Y. Cui, and X. Shao, "A deep Q-learning network based active object detection model with a novel training algorithm for service robots," *Front. Inf. Technol. Electron. Eng.*, vol. 23, no. 11, pp. 1673–1683, Sep. 2022, doi: 10.1631/fitee.2200109.
[56] P. Anderson et al., "On evaluation of embodied navigation agents," 2018, *arXiv: 1807.06757*.

Francesco Taioli received the MSc degree in computer engineering, graduating from the University of Verona, in 2022. He is currently working toward the PhD degree in the National PhD program in artificial intelligence with the Polytechnic of Turin, collaborating with the University of Verona. Currently supervised by Prof. Marco Cristani and Prof. Alessandro Farinelli, his main research interests are in computer vision and deep learning, with a focus on improving the autonomy of intelligent agents.



Francesco Giuliari (Student Member, IEEE) received the MSc degree in computer science from the University of Verona in 2018. He is currently working toward the PhD degree with the University of Genoa. He is currently affiliated with Istituto Italiano di Tecnologia under the supervision of Dr. Alessio Del Bue. His main research interests are in computer vision, scene understanding and vision-based agent navigation.



Yiming Wang (Member, IEEE) received the PhD degree in electric engineering from the Queen Mary University of London (U.K.) in 2018. She is a researcher with the Deep Visual Learning (DVL) unit, Fondazione Bruno Kessler (FBK). She works on topics related to active robotic perception and 3D scene understanding. She has organised a couple of workshops on related topics and she is actively serving as a reviewer for top-tier conferences and journals in both the Computer Vision and Robotics domains. She is a member of ELLIS.

Riccardo Berra is a computer science graduate from the University of Verona, currently working toward the master's degree program in computer engineering for Robotics and Smart Industry. He has collaborated several times in publications in the field of robotics and computer vision under the supervision of Prof. Marco Cristani and prof. Francesco Setti.

Alberto Castellini is assistant professor with the University of Verona, Dept. of Computer Science, since 2018. Before he was research fellow with Potsdam University/Max Planck Institute and University of Verona. His research interests include probabilistic planning under uncertainty, reinforcement learning, statistical learning and data analysis for intelligent systems. He published in artificial intelligence journals and conferences (*IEEE International Systems, Engineering Applications of Artificial Intelligence*, *Robotics/Automated Systems*, IJCAI, AAMAS, ICAPS, IROS).

Alessio Del Bue (Member, IEEE) is a tenured senior researcher leading the Pattern Analysis and computer VISion (PAVIS) Research Line of the Italian Institute of Technology (IIT), Genoa, Italy. He is a coauthor of more than 100 scientific publications in refereed journals and international conferences on computer vision and machine learning topics. His current research interests include 3D scene understanding from multi-modal input (images, depth, and audio). He is a member of the technical committees of major computer vision conferences (CVPR, ICCV, ECCV, and BMVC). He serves as an associate editor for *Pattern Recognition and Computer Vision and Image Understanding* journals. He is a member of ELLIS.

Alessandro Farinelli is a full professor with the University of Verona, Department of Computer Science, since 2019. His research interests focus on developing novel methodologies for Artificial Intelligence systems applied to robotics and cyber-physical systems. In particular, he focuses on multi-agent coordination, decentralised optimisation, reinforcement learning and data analysis for cyber-physical systems. He was principal Investigator for several national and international research projects in the broad area of Artificial Intelligence.

Marco Cristani (Member, IEEE) is full professor (Professore Ordinario) with the Department of Engineering for Innovation Medicine, University of Verona, associate member with the National Research Council (CNR), External Collaborator at the Italian Institute of Technology (IIT). His main research interests are in statistical pattern recognition and computer vision, mainly in deep learning and generative modeling, with application to social signal processing and fashion modeling. On these topics he has published more than 200 papers. He has organised 11 international workshops. He is or has been the Principal Investigator of several national and international projects, including PRIN and H2020 projects. He is an IAPR fellow.

Francesco Setti (Member, IEEE) is assistant professor with the University of Verona, Department of Engineering for Innovation Medicine and associate researcher of the Institute of Cognitive Science and Technology (ISTC-CNR). His research interests include computer vision, machine learning, and artificial intelligence applied to robotics and manufacturing. He is a co-author of more than 70 papers in international peer-reviewed journals and conferences. He is a member of IAPR, and BMVA, and he serves as a reviewer for all the major machine learning conferences and journals (including CVPR, ICCV, ECCV, and *IEEE Transactions on Pattern Analysis and Machine Intelligence*).

Open Access funding provided by 'Politecnico di Torino' within the CRUI CARE Agreement