entropy

MDPI

Article

Continuous Viewpoint Planning in Conjunction with Dynamic Exploration for Active Object Recognition

Haibo Sun [1,2,3,4], Feng Zhu [2,3,4,*], Yanzi Kong [2,3,4,5], Jianyu Wang [1,2,3,4] and Pengfei Zhao [2,3,4,5]

1 Faculty of Robot Science and Engineering, Northeastern University, Shenyang 110169, China; sunhaibo@sia.cn (H.S.); wangjianyu@sia.cn (J.W.)
2 Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China; kongyanzi@sia.cn (Y.K.); zhaopengfei@sia.cn (P.Z.)
3 Key Laboratory of Opto-Electronic Information Processing, Chinese Academy of Sciences, Shenyang 110016, China
4 Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110169, China
5 University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: fzhu@sia.cn

Abstract: Active object recognition (AOR) aims at collecting additional information to improve recognition performance by purposefully adjusting the viewpoint of an agent. How to determine the next best viewpoint of the agent, i.e., viewpoint planning (VP), is a research focus. Most existing VP methods perform viewpoint exploration in the discrete viewpoint space, which have to sample viewpoint space and may bring in significant quantization error. To address this challenge, a continuous VP approach for AOR based on reinforcement learning is proposed. Specifically, we use two separate neural networks to model the VP policy as a parameterized Gaussian distribution and resort the proximal policy optimization framework to learn the policy. Furthermore, an adaptive entropy regularization based dynamic exploration scheme is presented to automatically adjust the viewpoint exploration ability in the learning process. To the end, experimental results on the public dataset GERMS well demonstrate the superiority of our proposed VP method.

Keywords: active object recognition; continuous viewpoint planning; adaptive entropy regularization; dynamic exploration; proximal policy optimization

Citation: Sun, H.; Zhu, F.; Kong, Y.; Wang, J.; Zhao, P. Continuous Viewpoint Planning in Conjunction with Dynamic Exploration for Active Object Recognition. *Entropy* **2021**, *23*, 1702. https://doi.org/10.3390/e23121702

Academic Editors: Luis Javier García Villalba and Andrea Prati

Received: 8 November 2021
Accepted: 13 December 2021
Published: 20 December 2021

1. Introduction

Visual object recognition plays an important role in the fields of computer vision and robotics. It has been successfully applied into a large number of tasks, e.g., autonomous driving, manipulation and grasping, monitoring security, transportation surveillance [1], etc.

Most recognition systems exclusively focus on static image recognition, that is, the systems take a single snapshot as input and generate a category label estimate as output [2]. It is easy to produce recognition errors when the single-view image can not provide enough information. However, the vision behavior of people is exploratory, probing, and searching in order to better understand their surroundings. For example, you will go to the front of a person to confirm when you can not identify him from his back. Thus, if the viewpoint of an agent (e.g., an automatic mobile robot with a head mounted camera) is allowed to be changed, more detailed information will be collected to improve the performance of recognition.

The idea described above fits into the realm of active object recognition (AOR) [3–5], which gathers additional evidence to improve recognition performance by purposefully adjusting the viewpoint (position and orientation) of an agent. Many classic and latest AOR approaches are reviewed in [6,7]. The main focus of AOR research is viewpoint planning (VP) which means how to determine the next best viewpoint of the agent. A good VP policy can greatly ameliorate the recognition performance. In recent years, reinforcement learning has attracted growing research attention on viewpoint planning [8–12]. The agent

is able to learn a good VP policy under the guidance of hand-designed reward functions. The main algorithms involved in the learning process are dynamic programming [8] and Q-Learning [9–12]. Both dynamic programming based and Q-Learning based methods have made a great contribution to AOR. However, these VP methods explore discrete viewpoint space, which have to sample viewpoint space and may bring in significant quantization error.

To alleviate this problem, we propose a continuous viewpoint planning approach for AOR based on reinforcement learning in this work. The approach can effectively explore the continuous viewpoint space. To be specific, we employ recently presented proximal policy optimization (PPO) [13] framework to tackle the VP problem. The VP policy is represented by a Gaussian model that can be monotonically improved by the clipping mechanism of PPO. In addition, the standard deviation of the Gaussian model implies the viewpoint exploration ability, which represents the opportunity to try new viewpoints. As shown in Figure 1, the larger the standard deviation is, the stronger the exploration ability is. If the standard deviation is fixed in the whole policy learning process (fixed exploration), two unpleasant results will be produced: (1) the VP policy may stuck in local optimum due to insufficient exploration when the standard deviation is small; (2) the optimal VP policy can not be obtained when the standard deviation is large (because the optimal VP policy is a deterministic policy which is approximately equivalent to a Gaussian model with the small standard deviation). So, in the field of reinforcement learning, it generally hopes to have a higher exploration in the early stage of policy learning and gradually reduce it in the later in order to obtain a better policy [14]. Therefore, we develop a dynamic exploration scheme to automatically adjust viewpoint exploration in the learning process. The scheme is implemented by using separate neural networks for the representation of policy mean and standard deviation and training the mean and standard deviation at the same time. Moreover, entropy regularization [15] is introduced and improved to an adaptive version to prevent the exploration from shrinking prematurely. The experimental results on the public dataset GERMS [12] strongly support the effectiveness of our proposed VP method.

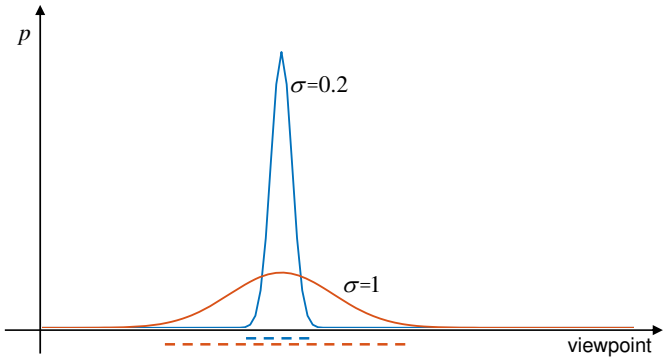

Figure 1. The illustration of viewpoint exploration ability. The exploration ability of the VP policy with the standard deviation $\sigma = 1$ is stronger than that of the VP policy with the standard deviation $\sigma = 0.2$. Because there are more possibilities to try new viewpoints when $\sigma = 1$.

The contributions of our work are as follows:

- A novel continuous viewpoint planning method for active object recognition based on proximal policy optimization is proposed to deal with the problem of quantization error of discrete viewpoint planning methods;
- An adaptive entropy regularization based dynamic exploration scheme is presented to automatically adjust viewpoint exploration in the learning process;
- Experiments are carried out on the public dataset GERMS, and the proposed method obtains rather promising results.

The remainder of the paper is laid out as follows. Section 2 reviews the related research. Section 3 formulates the problem. Section 4 details our continuous viewpoint planning

method. Section 5 shows the experiment results and analysis whereas we draw conclusions in Section 6.

2. Related Work

This section reviews related work about active object recognition and proximal policy optimization.

Active Object Recognition: Becerra et al. [8] model object detection as a Partially Observable Markov Decision Process problem, which is solved using Stochastic Dynamic Programming. In [9], researchers formally define the viewpoint selection as an optimization problem and use reinforcement learning for viewpoint training without user interaction. Malmir et al. [12] contribute a image-based AOR publicly dataset named GERMS and propose a deep Q-learning (DQL) system that learns to actively examine objects by minimizing overall classification error using standard back-propagation and Q-learning. Similarly, Liu et al. develop a hierarchical local-receptive-field-based extreme learning machine architecture to learn the state representation and utilize Q-learning to find the optimal policy [10]. In [11], researchers treat AOR as a Partially Observable Markov Decision Process and find corresponding action-values of training data using belief tree search. All above methods explore discrete viewpoint space, which may miss a few important object information owing to the quantization error of viewpoint. Therefore, we develop a continuous VP method for AOR to address this problem. The closest method to ours in this respect is [16] which resorts trust region policy optimization (TRPO) framework [17] to tackle the quantization error problem and shows better results on the dataset GERMS compared to the Q-Learning methods. However, in the TRPO-based AOR method, linear approximation of the optimization objective and quadratic approximation of the constraint are used to jointly direct policy update, leading to relatively high computation complexity. Although the researchers wisely employ extreme learning machine [18] to alleviate this problem, the learning speed is still unsatisfactory. Different from [16], we adopt a first-order optimization framework PPO [13] for continuous VP learning. It is computationally efficient and is able to guarantee monotonic performance improvement of VP policy. In addition, the VP policy standard deviation in [16] is fixed and small, which makes the viewpoint exploration insufficient during the learning process, resulting in the policy stuck in local optimum. However, we develop a dynamic exploration scheme in our work to automatically adjust the standard deviation in the learning process in order to obtain a better policy.

Proximal Policy Optimization: PPO has achieved significant successes in enormous applications. Gangapurwala et al. [19] introduce a guided constrained policy optimization framework based on PPO which guarantees the behavior of real quadruped robot within required safety constraints during training process. A centralized coordination scheme of automated vehicles at an intersection without traffic light using PPO is proposed to solve low computation efficiency suffered by state-of-the-art methods [20]. In [21], researchers apply PPO to the task of image captioning to establish a further improvement for the training phase of reinforcement learning. In [22], researchers propose an integrated metro service scheduling and train unit deployment with a PPO approach based on the deep reinforcement learning framework. A variant of PPO algorithm called memory proximal policy optimization is presented to solve quantum control tasks [23]. In [24], a PPO-based machine learning algorithm is implemented to decide on the replenishments of a group of collaborating companies. However, to our best knowledge, PPO has never been resorted for AOR task. In our work, it is firstly utilized for AOR to learn a continuous VP policy.

3. Problem Statement

In a visual AOR system, an agent will be automatically moved to capture images from different viewpoints to recognize an object. The current viewpoint is known to the agent in the recognition system. Specifically, at initial time $t = 0$, the viewpoint of agent is φ_0 and the captured image is I_{φ_0}. According to I_{φ_0}, we can predict the label of the object

to be recognized using a classifier. It is often that the single viewpoint image may be not sufficient to give a robust recognition result, we should move the agent to capture more images to improve the recognition performance. This requires us to plan an relative movement action a_t (i.e., VP) for the agent to obtain a new viewpoint that is $\varphi_{t+1} = \varphi_t + a_t$. Then, the new image $I_{\varphi_{t+1}}$ captured in the viewpoint φ_{t+1} will be used for the recognition again. The process like this will be repeated until a stop criteria is reached, such as the maximum of T steps.

An arbitrary action may lead to a worse view where the captured image does not provide useful information for recognition. Therefore, an effective VP policy is desirable. To this end, we consider the VP problem as a reinforcement learning one which is formulated as a six-element tuple $< S, A, r, \mathcal{P}, \gamma, \pi >$. S denotes the state space where every element s is generated by the images acquired from different viewpoints of an agent. A is the continuous action space where every action a is used to move the agent to a new viewpoint. $r : S \times A \rightarrow \mathbb{R}$ is a reward function designed to assess the value of one action in a certain state. $\mathcal{P} : S \times A \times S \rightarrow [0, 1]$ means the transition probability to the next state when an action is selected in the current state. $\gamma \in [0, 1]$ is a discount factor that represents the difference in importance between future rewards and present rewards. $\pi : S \times A \rightarrow [0, 1]$ is an continuous VP policy that describes the probability of selecting one action to produce a new viewpoint in a certain state. In the reinforcement learning setting, the VP problem is transformed to find the optimal policy π^*, which can move the agent to the best recognition viewpoints.

4. Proposed Method

To obtain the optimal continuous VP policy π^* for AOR, we employ PPO framework [13] to tackle this problem. Figure 2 shows our AOR pipeline based on PPO.

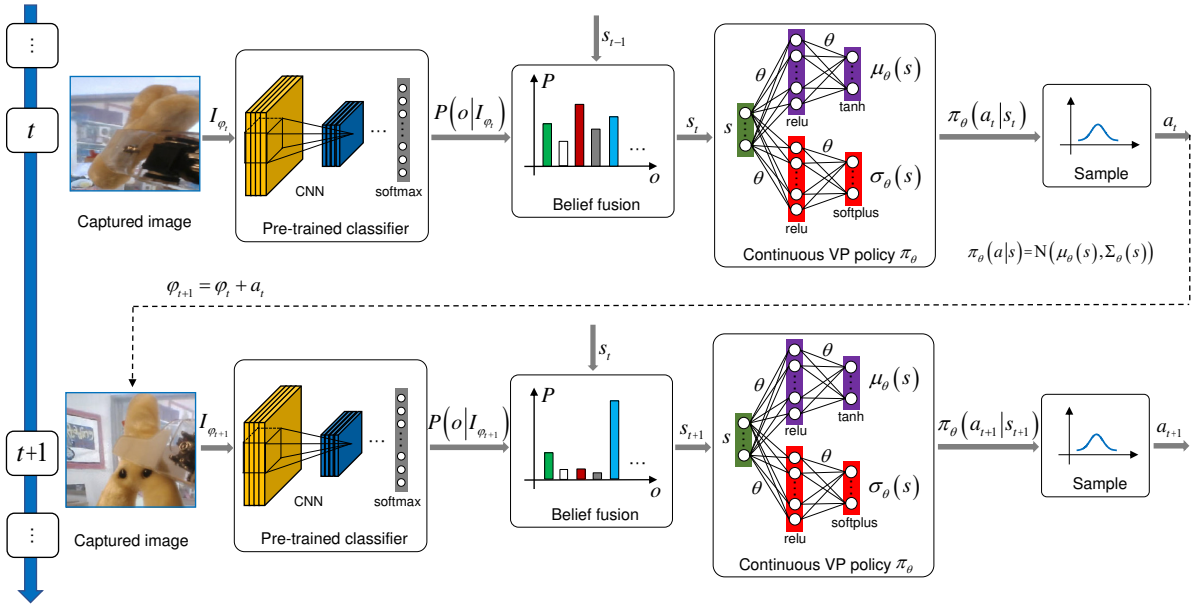

Figure 2. The proposed AOR pipeline. The pipeline adopts PPO framework [13] to learn the continuous VP policy π_θ that is denoted by a parameterized Gaussian model. In order to realize dynamic exploration, two separate neural networks are used for the representation of the policy mean and standard deviation of the Gaussian model and trained concurrently. During the training process, the policy π_θ is improved by collecting some sample trajectories $\{s_t, a_t, r(s_t, a_t)\}_{t=0}^{T}$ and optimizing the PPO objective.

During policy training process, at each time step t, an agent observes the state $s_t \in S$, takes an action $a_t \in A$ under current VP policy π (i.e., $a_t \sim \pi(a_t|s_t)$), generates a new state $s_{t+1} \sim \mathcal{P}(s_{t+1}|s_t, a_t)$, and receives a scalar reward $r(s_t, a_t)$. Starting from arbitrary initial state s_0 at time $t = 0$, the cumulative discounted reward function is

$$\eta(\pi) = \mathop{\mathbb{E}}_{\substack{a_t \sim \pi(a_t|s_t) \\ s_{t+1} \sim \mathcal{P}(s_{t+1}|s_t,a_t)}} [\sum_{t=0}^{T} \gamma^t r(s_t, a_t)], \tag{1}$$

where $\mathbb{E}[\cdot]$ denotes the expectation operator. T is the maximum number of planning. $\eta(\pi)$ is used to evaluate different VP polices. A better VP policy corresponds to a higher value of $\eta(\pi)$. We assume that VP policy π is parameterized by θ and denote it as π_θ. Thus, to find the optimal continuous VP policy π^* is to find the optimal parameter θ^* that can be solved by

$$\theta^* = \arg\max_\theta \eta(\pi_\theta). \tag{2}$$

The recent PPO framework [13] is adopted to address the optimization problem (2) in an iterative updating way. Let $\pi_{\theta_{old}}$ be the old policy, π_θ be the new policy after the policy update, and $\kappa(\theta)$ be the probability ratio $\kappa(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{old}}(a_t|s_t)$. In the PPO framework, θ^* in (2) can be achieved by maximizing a clipping surrogate objective (The detailed derivation process from (2) to (3) can refer to [13,17].):

$$\max_\theta L(\theta) = \mathop{\mathbb{E}}_{\pi_{\theta_{old}}} [\min(\kappa(\theta) A_{\pi_{\theta_{old}}}(s_t, a_t), clip(\kappa(\theta), 1-\epsilon, 1+\epsilon) A_{\pi_{\theta_{old}}}(s_t, a_t))], \tag{3}$$

where ϵ is a hyper-parameter to control the clipping ratio. $A_{\pi_{\theta_{old}}}(s_t, a_t)$ is advantage function under the old policy $\pi_{\theta_{old}}$, which is detailed in Section 4.4. In the following, we will elaborate the representation of state s_t, continuous VP policy π_θ, and reward function $r(s_t, a_t)$ in our PPO-based AOR pipeline and develop a training algorithm to solve the optimization problem in (3).

4.1. Belief Fusion for State Representation

As shown in Figure 2, the captured image I_{φ_t} is first transformed into a series of convolutional neural network (CNN) features. We then add a $softmax$ layer on the top of the CNN model to identify the concerned objects. The output of the $softmax$ layer is a vector that means the recognition belief over different objects. We denote the oth element of the belief vector as $P(o|I_{\varphi_t})$ where $o = 1, 2, ..., M$ is the object label. Like [25], the belief $P(o|I_{\varphi_t})$ is fused with the accumulated belief $P(o|I_{\varphi_0}, I_{\varphi_1}, ..., I_{\varphi_{t-1}})$ from previous images using Naive Bayes:

$$P(o|I_{\varphi_0}, I_{\varphi_1}, ..., I_{\varphi_t}) = \beta_t P(o|I_{\varphi_t}) P(o|I_{\varphi_0}, I_{\varphi_1}, ..., I_{\varphi_{t-1}}). \tag{4}$$

The fusion result $P(o|I_{\varphi_0}, I_{\varphi_1}, ..., I_{\varphi_t})$ is the new accumulated belief at time step t. β_t is a normalizing coefficient ($\beta_t = 1/\sum_o P(o|I_{\varphi_t}) P(o|I_{\varphi_0}, I_{\varphi_1}, ..., I_{\varphi_{t-1}})$) that makes $\sum_o P(o|I_{\varphi_0}, I_{\varphi_1}, ..., I_{\varphi_t}) = 1$ hold. In this work, the accumulated belief is used for the representation of the recognition state (i.e., $s_t = P(o|I_{\varphi_0}, I_{\varphi_1}, ..., I_{\varphi_t}), o = 1, 2, ..., M$) at each time step. It is worth noting that the parameters of the classifier (composed of the CNN model and the $softmax$ layer) are pre-trained with the images from different viewpoints of different objects and invariable during the training process of continuous VP policy.

4.2. Continuous VP Policy Network Combined with Dynamic Exploration

Similar to [16], the continuous VP policy is represented by a parameterized Gaussian distribution. However, ref. [16] only parameterizes the policy mean μ with a neural network, that is, $\pi_\theta(a|s) = \mathcal{N}(\mu_\theta(s), \sum)$ (Viewpoint is composed of orientation and position, so the planning action a may be a multi-dimensional vector. Therefore, the Gaussian model may be a multivariate form. It is usually assumed that the variables in a are independent of each other, so the covariance matrix $\sum$ is a diagonal matrix, i.e., $\sum = diag(\sigma_1^2, \sigma_2^2, ..., \sigma_d^2)$. σ is standard deviation and d is the dimension of a.). The standard deviations in the covariance matrix $\sum$ are small and invariable in the whole training process. As analyzed in Section 1, the standard deviation implies the viewpoint exploration ability, the fixed small standard deviation may make the VP policy stuck in local optimum due to insufficient

exploration. Therefore, an adaptive entropy regularization based dynamic exploration scheme is developed to automatically adjust the standard deviation in the training process in order to obtain a better policy. The research process and implementation details of the scheme are as follows.

Parameterization of the Policy Mean and Standard Deviation: The scheme is first realized by concurrently parameterizing the policy mean and standard deviations with two separate neural networks ($\mu_\theta(s)$ or $\mu(s;\theta)$ and $\sigma_\theta(s)$ or $\sigma(s;\theta)$) and training them at the same time. As shown in Figure 2, $\mu_\theta(s)$ and $\sigma_\theta(s)$ are two single hidden-layer fully-connected neural networks which take state as input and output the mean vector and standard deviation vector. The parameters of them are collectively called θ. Consequently, the VP policy is recorded as $\pi_\theta(a|s) = \mathcal{N}(\mu_\theta(s), \sum_\theta(s))$ which is expanded to

$$\pi_\theta(a|s) = \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi}\sigma_i(s;\theta)} \exp^{-\frac{(a_i-\mu_i(s;\theta))^2}{2\sigma_i(s;\theta)^2}}. \tag{5}$$

The ith element of the mean vector and standard deviation vector are represented as $\mu_i(s;\theta)$ and $\sigma_i(s;\theta)$, respectively. d is the dimension of action a. During training, the update of parameter θ under the PPO framework will simultaneously affect the policy mean and standard deviations, leading to the dynamic exploration.

Entropy Regularization: As stated in Section 1, in reinforcement learning, it generally hopes to have a higher exploration in the early stage of policy learning and gradually reduce it in the later in order to obtain a better policy [14]. However, we find the standard deviations shrink prematurely and adjust in a small range in the training process. As shown in Figure 3, it is the change of standard deviation in the training process of GERMS dataset [12] which has a single action dimension (A shrinkage case with two action dimensions is shown in [14]). It shrinks rapidly to a small value soon after the beginning of training and always keeps in a small value range (the curve with $c = 0$ in Figure 3), which may also result in the insufficient exploration. To address this problem, we then introduce entropy regularization [15] to the PPO optimization objective (3) to prevent the exploration from shrinking prematurely. Therefore, (3) is transformed into:

$$\begin{aligned} \max_\theta L^{Ent}(\theta) = \mathop{\mathbb{E}}_{\pi_{\theta_{old}}} [&\min(\kappa(\theta)A_{\pi_{\theta_{old}}}(s_t,a_t), \\ &clip(\kappa(\theta), 1-\epsilon, 1+\epsilon)A_{\pi_{\theta_{old}}}(s_t,a_t)) + cH(\pi_\theta(\cdot|s_t))], \end{aligned} \tag{6}$$

where c is a constant coefficient and $H(\cdot)$ is entropy operator ($H(x) = -\int p(x)\log p(x)$ or $H(x) = -\sum p(x)\log p(x)$. The entropy of a multivariate normal distribution is $\frac{1}{2}\log(2\pi e)^d|\sum|$.).

Adaptive Entropy Regularization Coefficient: In our experiment, we find the constant coefficient c in (6) is a hyper-parameter that is difficult to tune. As shown in Figure 3, when c is less than or equal to 0.03, entropy regularization fails to prevent the premature decay of exploration; when c is greater than 0.03, the standard deviation increases explosively. Thus, to tackle this problem, we last propose an adaptive entropy regularization method that can adapt the coefficient to achieve the appropriate exploration ability in the training process. The coefficient c in (6) is improved to

$$c = \begin{cases} c_{div}, & \exists i\ \bar{\sigma}_i < \sigma_{L_i}(t) \\ -c_{div}, & \exists i\ \bar{\sigma}_i > \sigma_{H_i}(t), \\ 0, & otherwise \end{cases} \tag{7}$$

where c_{div} is a divergence coefficient such as 0.04, 0.05, 0.1, 0.3, and 0.5 in Figure 3. If the planning action is multidimensional, then c_{div} is a coefficient that makes the standard deviation of each dimension diverge. $\sigma_{H_i}(t)$ and $\sigma_{L_i}(t)$ are the i-dimensional upper and lower boundaries of the standard deviation you want to maintain in the training. They are the functions of training time node t. In our work, we model them as stage functions shown in Figure 4. To be specific, the stage functions in a certain dimension are defined as

$$\begin{aligned} \sigma_L(t) &= \sigma_0 - [max(t - T_M, 0)/T_W]\sigma_\Delta \\ \sigma_H(t) &= \sigma_L(t) + \sigma_S. \end{aligned} \tag{8}$$

T_W is the training duration of each stage. According to it, the total training time can be evenly divided into several stages. σ_0 is the initial standard deviation. σ_Δ is the increment of the standard deviation. σ_S is the boundary range. $[\cdot]$ is the rounding operator, e.g., $[1/3] = 0$. $max(t - T_M, 0)$ is to increase the training time of the first stage by T_M. As shown in Figure 3, this is because it takes some time to raise the standard deviation to the boundary value of the first stage at the beginning of training.

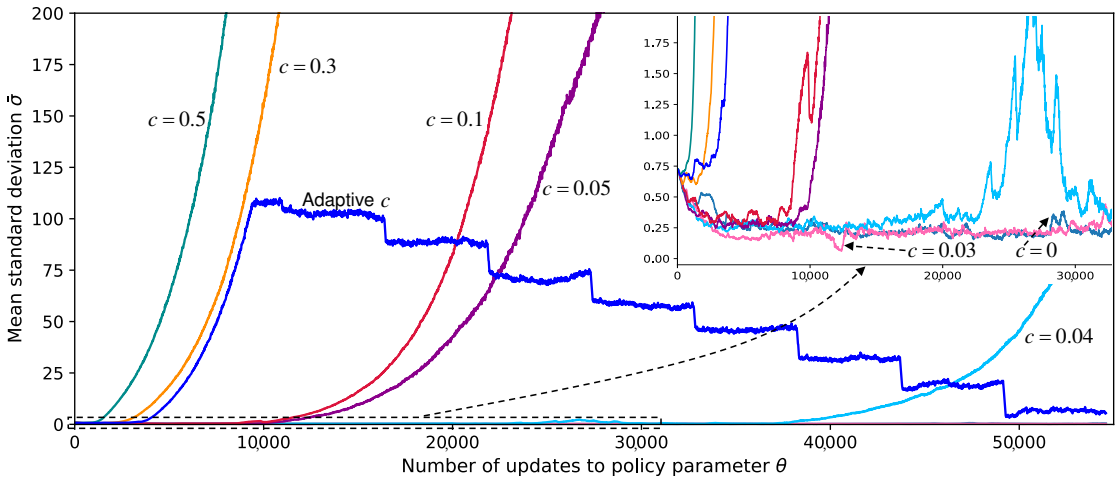

Figure 3. The changes of exploration ability in the training process of GERMS left arm dataset [12] under different dynamic exploration schemes. Because the standard deviation is a function of the state, the standard deviation representing the exploration ability refers to the average of the standard deviation $\bar{\sigma}$ corresponding to all states. However, there are infinite states, so $\bar{\sigma}$ can not be calculated. In the training, we use the average of the standard deviation of some sample states to approximately replace $\bar{\sigma}$. We implement three dynamic exploration schemes step by step: (1) the first is the simultaneous parameterization of policy mean and standard deviation with two separate neural networks (the curve with $c = 0$); (2) the second is to add the constant coefficient entropy regularization on the basis of (1) (the curves with $c = 0.03, 0.04, 0.05, 0.1, 0.3, or\ 0.5$); (3) the third is that the constant coefficient is improved into an adaptive version on the basis of (2) (the curve with Adaptive c). After experimental comparison, scheme (3) can meet our dynamic exploration need.

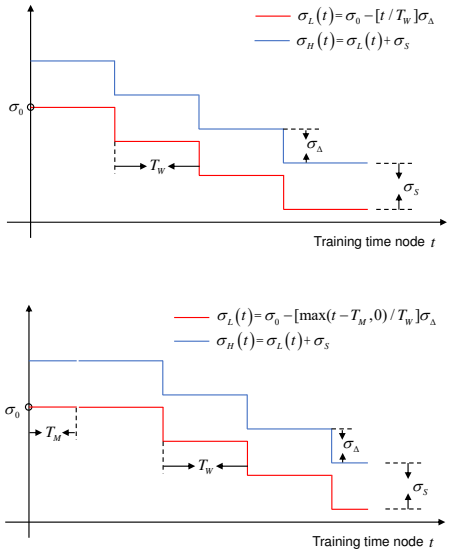

Figure 4. The diagram of upper and lower boundary functions of standard deviation.

After experimental verification, the dynamic exploration with adaptive entropy regularization can meet our exploration requirement.

4.3. Reward Setting

Reward function $r(s_t, a_t)$ plays an important role in encouraging effective viewpoint selection. In Section 4.1, the recognition state ($s_t = P(o|I_{\varphi_0}, I_{\varphi_1}, ..., I_{\varphi_t}), o = 1, 2, ..., M$) describes a probability distribution over different objects. The flatter the distribution is, the stronger the recognition ambiguity is. Here, we resort information entropy [26,27] to quantify the ambiguity. Then the ambiguity in state s_t is represented as $H(s_t)$. The goal of AOR is to eliminate this ambiguity to improve recognition performance by viewpoint planning. A beneficial viewpoint attempt can reduce the current ambiguity. Therefore, we design the reward function according to the ambiguity in different states after viewpoint selection. Let $\hat{o}_{t+1}$ be the predicted result and o^* be the label of the image in the new viewpoint ($I_{\varphi_{t+1}} = I_{\varphi_t} + a_t$). Among them, $\hat{o}_{t+1} = argmax_o P(o|I_{\varphi_0}, I_{\varphi_1}, ..., I_{\varphi_{t+1}})$. If the predicted result $\hat{o}_{t+1}$ is right and the information entropy $H(s_{t+1})$ is smaller than $H(s_t)$ in state s_t, it means that the VP action a_t in state s_t is useful for recognition. Then the agent will receive a positive reward. Otherwise, the reward is non positive when the entropy does not decrease or the prediction is wrong. To sum up, the reward function is formulated as

$$r(s_t, a_t) = \begin{cases} -1, & \hat{o}_{t+1} \neq o^* \\ 0, & \hat{o}_{t+1} = o^*, H(s_{t+1}) \geqslant H(s_t), \\ 1, & \hat{o}_{t+1} = o^*, H(s_{t+1}) < H(s_t) \end{cases} \tag{9}$$

where $r(s_t, a_t)$ can be denoted as r_t for simplicity.

4.4. Training the Policy Network

To solve the optimization problem in (6), we develop a training algorithm to iteratively update θ in the policy network. The algorithm shown in Algorithm 1 is Actor–Critic style [15].

To replace the expectation operator in (6), we apply Monte Carlo method [28] to deal with it in an approximate manner. Specifically, we repeat N times to run the old policy $\pi_{\theta_{old}}$ for T time steps to collect a trajectory $\{s_t, a_t, r_t, s_{t+1}\}_{t=0}^{T}$. With N trajectories, (6) can be approximated as:

$$\begin{aligned} \max_{\theta} \hat{L}^{Ent}(\theta) = \frac{1}{N(T+1)} \sum_{i=1}^{N} \sum_{t=0}^{T} [\min(\kappa^{(i)}(\theta) A^{(i)}_{\pi_{\theta_{old}}}(s_t, a_t) \\ , clip(\kappa^{(i)}(\theta), 1-\epsilon, 1+\epsilon) A^{(i)}_{\pi_{\theta_{old}}}(s_t, a_t)) + cH(\pi^{(i)}_{\theta}(\cdot|s_t))]. \end{aligned} \tag{10}$$

The advantage function $A_{\pi_{\theta_{old}}}(s_t, a_t)$ can be estimated using the technology of generalized advantage estimation (GAE) [29]:

$$\begin{aligned} & A_{\pi_{\theta_{old}}}(s_t, a_t) = \delta_t + (\gamma\lambda)\delta_{t+1} + ... + (\gamma\lambda)^{T-t}\delta_T, \\ & where \quad \delta_t = r_t + \gamma V_{\pi_{\theta_{old}}}(s_{t+1}) - V_{\pi_{\theta_{old}}}(s_t). \end{aligned} \tag{11}$$

$V_{\pi_{\theta_{old}}}(\cdot)$ is state value function under the old VP policy $\pi_{\theta_{old}}$. It is approximately represented by a two-layer fully connected network with parameter ω. The network maps the state s_t to the function value $V(s_t; \omega)$. We update ω to obtain the state value function corresponding to different VP policies. We use the N trajectories (sampled by $\pi_{\theta_{old}}$) again to fit the state value function $V_{\pi_{\theta_{old}}}(s_t; \omega)$ of the old policy $\pi_{\theta_{old}}$ by solving the optimization problem:

$$\min_{\omega} \hat{L}(\omega) = \frac{1}{N(T+1)} \sum_{i=1}^{N} \sum_{t=0}^{T} (V^{(i)}_{Target}(s_t) - V^{(i)}_{\pi_{\theta_{old}}}(s_t; \omega))^2. \tag{12}$$

Algorithm 1: Training the continuous VP policy network

Input: Parameters: L, N, T, K_A, K_C.
Output: Parameter θ.

1 Create a new policy network, an old policy network and a state value network with parameters θ, θ_{old}, and ω, respectively. The new and old policy network has the same network structure. Initialize the parameters θ, θ_{old}, and ω randomly.
2 **for** ***episode*** $\leftarrow 1$ ***to*** L **do**
3 $\quad \theta_{old} = \theta$
4 $\quad$ **for** $i \leftarrow 1$ ***to*** N **do**
5 $\quad\quad$ Run policy $\pi_{\theta_{old}}(a|s)$ for T time steps, collecting a trajectory $\{s_t, a_t, r_t, s_{t+1}\}_{t=0}^{T}$ where $a_t \sim \pi(a_t|s_t), s_{t+1} \sim \mathcal{P}(s_{t+1}|s_t, a_t)$.
6 $\quad$ For each t in every trajectory, estimate advantage function $A_{\pi_{\theta_{old}}}(s_t, a_t)$ according to (11).
7 $\quad$ **for** $i \leftarrow 1$ ***to*** K_A **do**
8 $\quad\quad$ Optimize $\hat{L}^{Ent}(\theta)$ in (10) w.r.t. θ with $N(T+1)$ size or $M \leqslant N(T+1)$ minibatch size samples.
9 $\quad$ **for** $i \leftarrow 1$ ***to*** K_C **do**
10 $\quad\quad$ Optimize $\hat{L}(\omega)$ in (12) w.r.t. ω with $N(T+1)$ size or $M \leqslant N(T+1)$ minibatch size samples.
11 $\quad$ Adapt the entropy regularization coefficient c in (10) in the light of (7) with the new policy network π_θ and $N(T+1)$ samples.
12 **return** θ

$V_{Target}(s_t)$ is not involved in the optimization procedure. It is calculated using $V_{Target}(s_t) = r_t + \gamma r_{t+1} + ... + \gamma^{T-t-1} r_{T-1} + \gamma^{T-t} V_{\pi_{\theta_{old}}}(s_T; \omega)$ in advance.

The iterative update process of (10) and (12) is shown in lines 7–10 of Algorithm 1.

Once the optimal parameter θ^* is obtained, it can be used for the practical AOR task. In state s_t, the planned action is $a_t \sim \mathcal{N}(\mu_{\theta^*}(s_t), \sum_{\theta^*}(s_t))$, and the next best viewpoint is $\varphi_{t+1} = \varphi_t + a_t$.

5. Experiments

5.1. Experimental Setup

Dataset and Metric: The GERMS dataset [12] shown in Figure 5 is collected in the context of the RUBI project whose intention is to develop a robot that interact with toddlers in early childhood education. It is composed of 1365 video tracks of give-and-take trials using 136 different soft toy objects. The tracks are divided according to the arm of the robot, with roughly half the training and testing tracks being the left arm and the other half the right arm. Each trial generates a track that records the robot putting the grasped object in its center of view, rotating it by 180° and then returning it. During the trial, the robot continuously saves images from its head-mounted camera at 30 frames per second, as shown in Figure 6. Meanwhile, the joint position and object label are recorded. These data are stored in a track, a series of which constitutes the dataset. On average, each track contains 150 images, Table 1 outlines the number of images in the dataset. These joint positions in each track allow researchers to simulate different VP methods in one dimensional action space. The performance of different VP methods is evaluated using recognition accuracy that is the average of the entire test set.

Figure 5. The GERMS dataset [12]. The objects are soft toys describing various human cell types, microbes and disease-related organisms.

Table 1. GERMS dataset statistics (mean ± std).

	Number of Tracks	Images/Track	Total Number of Images
Train	816	157 ± 12	76,722
Test	549	145 ± 19	51,561

Figure 6. The images from different viewpoints in different tracks.

Implementation Details: In this work, we employ the Tensorflow platform [30] to implement the proposed method. The CNN model used in the pre-trained classifier is VGG-net provided in [12], which can transform each image in GERMS into a 4096-dimensional feature vector. The number of neurons in the last $softmax$ layer of the pre-trained classifier is 136. In the policy $\mu_\theta(s)$ network, the number of neurons and the activation function in the hidden layer are 1024 and $relu$; The last layer uses $tanh$ activation function and has one neuron. In order to match the viewpoint range of GERMS, we multiply the output of $tanh$ by 512, so that the next relative VP action range is $[-45^\circ, 45^\circ]$. In the policy $\sigma_\theta(s)$ network, the configuration of the hidden layer is consistent with that in $\mu_\theta(s)$; The number of neurons and the activation function in the last layer are 1 and $softplus$. The configuration of the hidden layer in the state value network $V_\omega(s)$ is same as that in $\mu_\theta(s)$. The reward discount factor γ is 0.96, and the GAE parameter λ is 0.95. The clipping ration parameter ϵ is empirically set as $\epsilon = 0.2$ in the light of the original implementation of PPO [13]. The VP policy converges after 4200 episodes in the training process, therefore, we set $L = 4200$. N and the minibatch size M are all 128. K_A and K_C are 1 and 10. The maximum step T for recognition is set as $T = 12$. The Adam optimizer [31] is used for the optimization of the policy network and the state value network. The learning rates of them are 0.0001 and 0.0002. In the dynamic exploration, the parameters c_{div}, σ_0, T_M, T_W, σ_Δ and σ_S are 0.3, 106, 3, 3, 14, and 14, respectively.

5.2. Ablation Study

To investigate the effectiveness of our dynamic exploration scheme, we intend to conduct the variant experiments with different components ablation. Table 2 shows the abbreviations and interpretations of different components. In the variant experiments, the components AERC, ER, and SSDN are gradually removed.

Table 2. Abbreviations and interpretations for different components in our dynamic exploration scheme.

Abbreviation	Interpretation
BL	Baseline PPO framework [13] with a fixed exploration scheme (i.e., the standard deviation σ is a constant)
SSDN	Separate standard deviation network
ER	Entropy regularization (with a fixed coefficient)
AERC	Adaptive entropy regularization coefficient

The experimental results are presented in Figure 7, where the recognition accuracy is a function of the number of planned actions. From Figure 7, we can notice that the performance degrades heavily after removing the component AERC. The results of the experiments BL($\sigma = 0.1$), BL+SSDN, and BL+SSDN+ER($c = 0.03$) are similar. This is because their exploration ability is all at a low level. Although the experiment BL($\sigma = 100$) has a high exploration ability, the VP policy can not converge to the optimal. So its result is slightly worse. The result of experiment BL+SSDN+ER($c = 0.3$) is the most unsatisfactory, because its standard deviation increases explosively as shown in Figure 3. This study validates the effectiveness of our proposed adaptive entropy regularization based dynamic exploration scheme.

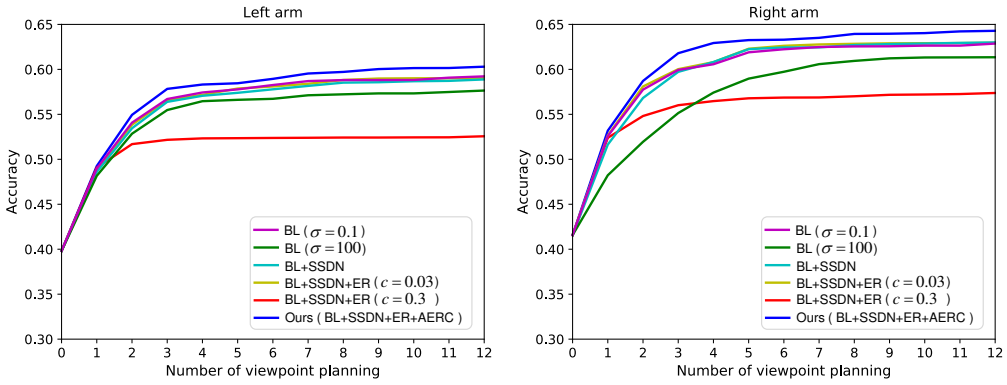

Figure 7. The performance comparison results of ablation experiments.

5.3. Dynamic Exploration Study

In our dynamic exploration scheme, the standard deviation σ is adapted by updating the VP policy parameters θ during the training. Another natural idea (i.e., independent linear decaying dynamic exploration, ILDDE) is to adjust σ independently of parameters θ. The idea is realized as

$$\sigma(t) = \frac{(\sigma_L - \sigma_0)}{T_L} t + \sigma_0 \quad (\sigma_L < \sigma_0), \tag{13}$$

where σ is a linear decaying function of the training time node t. σ_0 and σ_L are the initial and final σ values, respectively. T_L is the total training time. Therefore, the VP policy can be represented as $\pi_\theta(a|s) = \mathcal{N}(\mu_\theta(s), \sum(t))$ where $\sum(t) = diag(\sigma_1^2(t), \sigma_2^2(t), ..., \sigma_d^2(t))$. In the training, the update of parameters θ only affects the policy mean, the policy standard deviation is independently adapted by (13). We experiment with this idea and compare it with our scheme. In the experiment, except that the independent network $\sigma_\theta(s)$ in Figure 2 is removed and replaced with $\sigma(t)$ in (13), everything else is exactly the same. From the

presented results in Figure 8, we can notice that the performance of our scheme is much better than that of ILDDE. This is because the VP policy corresponding to ILDDE is affected by two parameters: θ and t. However, t does not participate in the optimization process, which may make the learned policy worse and worse. However, in our scheme, the policy mean and standard deviation are only related to θ, and participate in the whole optimization process.

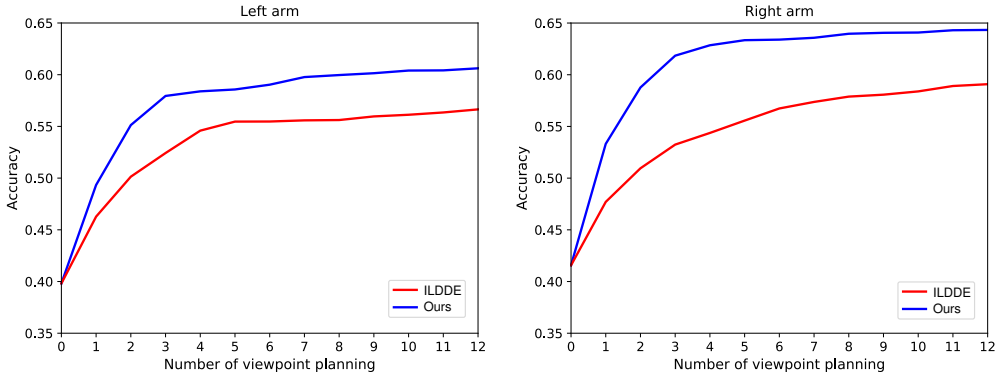

Figure 8. The performance comparison results of continuous VP policies combined with different dynamic exploration schemes. The parameters σ_0, σ_L, and T_L involved in ILDDE are 120, 0.1, and 4200.

5.4. Comparison with the State-of-the-Art Methods

In this subsection, several baselines [10] and state-of-the-art VP approaches [11,12,16] are employed for experiment comparison with our continuous VP method, which are showed as follows:

- Random policy [10] plans a random action from the action space $\{\pm\frac{\pi}{64}, \pm\frac{\pi}{32}, \pm\frac{\pi}{16}, \pm\frac{\pi}{8}, \pm\frac{\pi}{4}\}$ with uniform probability;
- Sequential policy [10] moves the agent to the next immediate position in the same direction;
- DQL policy [11,12] exploits deep Q-Learning algorithm to learn a discrete VP policy. The discrete action space is $\{\pm\frac{\pi}{64}, \pm\frac{\pi}{32}, \pm\frac{\pi}{16}, \pm\frac{\pi}{8}, \pm\frac{\pi}{4}\}$;
- E-TRPO policy [16] develops a continuous VP method which is implemented by trust region policy optimization [17] and extreme learning machine [18]. It represents the VP policy as a Gaussian model and learns the policy with a fixed exploration scheme.

For a fair comparison, the classifiers of different methods are the same in the experiment. The evaluation results of our VP model against other approaches are presented in Figure 9, from which we have the following observations: (1) Our proposed method achieve better performance compared with the state-of-the-art methods; (2) The performance of active policy is significantly better than that of passive policy. Random policy and Sequential policy are essentially passive VP policies. They do not actively plan the next viewpoint according to the information obtained from the previous viewpoints. However, DQL policy, E-TRPO policy, and the proposed method use the previous information to plan the next viewpoint, so they are active VP policies; (3) The performance of continuous VP policy outperforms that of discrete VP policy. DQL policy is a discrete VP policy while E-TRPO policy and our method are continuous VP policies. The continuous VP policy explores in the continuous viewpoint space and will not miss some important viewpoints; (4) Compared with the continuous VP method E-TRPO, our continuous VP model has better performance. This is mainly because we present an effective dynamic exploration scheme, which can explore more new viewpoints and find better solutions.

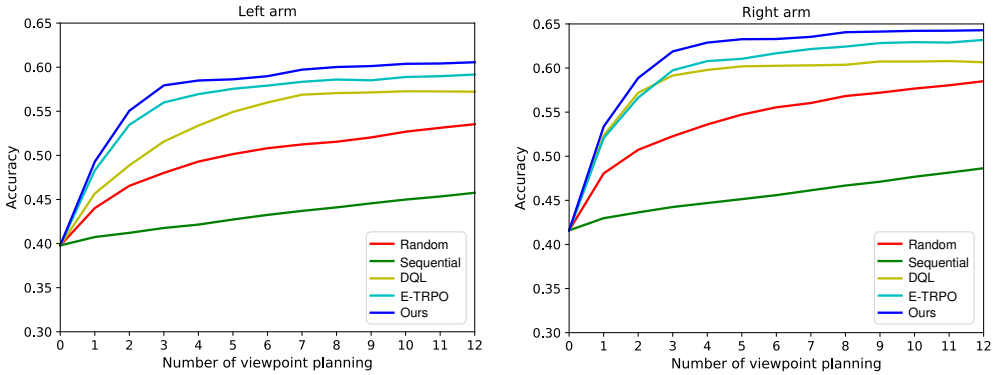

Figure 9. Performance comparison between our proposed continuous VP method and several competing approaches.

6. Conclusions

In this work, we develop a continuous viewpoint planning method for active object recognition based on reinforcement learning. More specifically, the viewpoint planning policy is represented as a parameterized Gaussian model and learned using the proximal policy framework. We also design a dynamic exploration scheme based on adaptive entropy regularization to automatically adjust the viewpoint exploration ability in the learning process. Experiments on the public dataset GERMS show the superiority of our method.

Author Contributions: Conceptualization, H.S.; methodology, F.Z. and H.S.; software, H.S. and Y.K.; validation, F.Z. and H.S.; formal analysis, H.S.; investigation, H.S.; resources, F.Z.; data curation, Y.K.; writing—original draft preparation, H.S.; writing—review and editing, J.W. and P.Z.; visualization, H.S.; supervision, J.W. and P.Z.; project administration, F.Z.; funding acquisition, F.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant no. U1713216.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used in this work are available at https://sites.google.com/a/eng.ucsd.edu/mmalmir/code-software-datasets, accessed on 1 November 2021.

Conflicts of Interest: The authors declare that they have no competing interests.

References

1. Pal, S.K.; Pramanik, A.; Maiti, J.; Mitra, P. Deep learning in multi-object detection and tracking: State of the art. *Appl. Intell.* **2021**, *51*, 6400–6429. [CrossRef]
2. Jayaraman, D.; Grauman, K. End-to-End Policy Learning for Active Visual Categorization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1601–1614. [CrossRef]
3. Patten, T.; Zillich, M.; Fitch, R.; Vincze, M.; Sukkarieh, S. Viewpoint evaluation for online 3-D active object classification. *IEEE Robot. Autom. Lett.* **2015**, *1*, 73–81. [CrossRef]
4. Potthast, C.; Breitenmoser, A.; Sha, F.; Sukhatme, G.S. Active multi-view object recognition: A unifying view on online feature selection and view planning. *Robot. Auton. Syst.* **2016**, *84*, 31–47. [CrossRef]
5. Wu, K.; Ranasinghe, R.; Dissanayake, G. Active recognition and pose estimation of household objects in clutter. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 4230–4237.
6. Andreopoulos, A.; Tsotsos, J.K. 50 Years of object recognition: Directions forward. *Comput. Vis. Image Underst.* **2013**, *117*, 827–891. [CrossRef]
7. Zeng, R.; Wen, Y.; Zhao, W.; Liu, Y.J. View planning in robot active vision: A survey of systems, algorithms, and applications. *Comput. Vis. Media* **2020**, *6*, 225–245. [CrossRef]
8. Becerra, I.; Valentin-Coronado, L.M.; Murrieta-Cid, R.; Latombe, J.C. Appearance-based motion strategies for object detection. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 6455–6461.

9. Deinzer, F.; Denzler, J.; Derichs, C.; Niemann, H. Aspects of optimal viewpoint selection and viewpoint fusion. In Proceedings of the Asian Conference on Computer Vision, Hyderabad, India, 13–16 January 2006; pp. 902–912.
10. Liu, H.; Li, F.; Xu, X.; Sun, F. Active object recognition using hierarchical local-receptive-field-based extreme learning machine. *Memetic Comput.* **2018**, *10*, 233–241. [CrossRef]
11. Malmir, M.; Cottrell, G.W. Belief tree search for active object recognition. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 4276–4283.
12. Malmir, M.; Sikka, K.; Forster, D.; Movellan, J.R.; Cottrell, G. Deep Q-Learning for Active Recognition of GERMS: Baseline Performance on a Standardized Dataset for Active Learning. In Proceedings of the British Machine Vision Conference (BMVC), Swansea, UK, 7–10 September 2015; pp. 161.1–161.11.
13. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
14. Hämäläinen, P.; Babadi, A.; Ma, X.; Lehtinen, J. PPO-CMA: Proximal policy optimization with covariance matrix adaptation. In Proceedings of the 2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP), Espoo, Finland, 21–24 September 2020; pp. 1–6.
15. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1928–1937.
16. Liu, H.; Wu, Y.; Sun, F. Extreme trust region policy optimization for active object recognition. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 2253–2258. [CrossRef] [PubMed]
17. Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; Moritz, P. Trust region policy optimization. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 1889–1897.
18. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
19. Gangapurwala, S.; Mitchell, A.; Havoutis, I. Guided constrained policy optimization for dynamic quadrupedal robot locomotion. *IEEE Robot. Autom. Lett.* **2020**, *5*, 3642–3649. [CrossRef]
20. Guan, Y.; Ren, Y.; Li, S.E.; Sun, Q.; Luo, L.; Li, K. Centralized cooperation for connected and automated vehicles at intersections by proximal policy optimization. *IEEE Trans. Veh. Technol.* **2020**, *69*, 12597–12608. [CrossRef]
21. Zhang, L.; Zhang, Y.; Zhao, X.; Zou, Z. Image captioning via proximal policy optimization. *Image Vis. Comput.* **2021**, *108*, 104126. [CrossRef]
22. Ying, C.S.; Chow, A.H.; Wang, Y.H.; Chin, K.S. Adaptive Metro Service Schedule and Train Composition with a Proximal Policy Optimization Approach Based on Deep Reinforcement Learning. *IEEE Trans. Intell. Transp. Syst.* **2021**, *6*, 1–12. [CrossRef]
23. August, M.; Hernández-Lobato, J.M. Taking gradients through experiments: LSTMs and memory proximal policy optimization for black-box quantum control. In Proceedings of the International Conference on High Performance Computing, Frankfurt, Germany, 24–28 June 2018; pp. 591–613.
24. Vanvuchelen, N.; Gijsbrechts, J.; Boute, R. Use of Proximal Policy Optimization for the Joint Replenishment Problem. *Comput. Ind.* **2020**, *119*, 103239. [CrossRef]
25. Paletta, L.; Pinz, A. Active object recognition by view integration and reinforcement learning. *Robot. Auton. Syst.* **2000**, *31*, 71–86. [CrossRef]
26. Zhao, D.; Chen, Y.; Lv, L. Deep reinforcement learning with visual attention for vehicle classification. *IEEE Trans. Cogn. Dev. Syst.* **2016**, *9*, 356–367. [CrossRef]
27. Liu, H.; Sun, F.; Zhang, X. Robotic material perception using active multimodal fusion. *IEEE Trans. Ind. Electron.* **2018**, *66*, 9878–9886. [CrossRef]
28. Hammersley, J. *Monte Carlo Methods*; Springer Science and Business Media: Berlin/Heidelberg, Germany, 2013.
29. Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; Abbeel, P. High-dimensional continuous control using generalized advantage estimation. *arXiv* **2015**, arXiv:1506.02438.
30. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A System for Large-Scale Machine Learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
31. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.