Learning viewpoint planning in active recognition on a small sampling budget: a Kriging approach

Joseph Defretin, Julien Marzat, Hélène Piet-Lahanier

Learning Viewpoint Planning in Active Recognition on a Small Sampling Budget: a Kriging Approach

Joseph Defretin[†‡], Julien Marzat[‡], Hélène Piet-Lahanier[‡]
[†] *CMLA, ENS Cachan, CNRS, UniverSud, 61 Avenue du Président Wilson, F-94230 Cachan*
[‡] *ONERA, Chemin de la Hunière FR-91761 Palaiseau Cedex, France*
<firstname>.<lastname>@onera.fr

***Abstract*—This paper focuses on viewpoint planning for 3D active object recognition. The objective is to design a planning policy into a Q-learning framework with a limited number of samples. Most existing stochastic techniques are therefore inapplicable. We propose to use Kriging and Bayesian Optimization coupled with Q-learning to obtain a computationally-efficient viewpoint-planning design, under a restrictive sampling budget. Experimental results on a representative database, including a comparison with classical approaches, show promising results for this strategy.**

***Keywords*-Active Recognition, Reinforcement Learning, Q-learning, Kriging, Bayesian Optimization, Viewpoint Planning**

I. Introduction

Object recognition is a field of great interest for an autonomous vehicle, named agent in what follows, equipped with an on-board inexpensive camera. Since such sensors generate only 2D data, numerous approaches have been proposed to recognize objects from image patterns, but classifiers are generally tuned for a restricted object pose. Besides, intrinsic similarities between objects generate visual ambiguities, thus recognition accuracy strongly depends on the chosen viewpoint. As the agent has the potential to fully explore the viewpoint space, richer visual information may be available by changing the viewpoint. The identification of the object of interest might then be realized from a sequence of observations. This sequence must be chosen to ensure minimal ambiguity of the classification, which is the aim of active recognition. Several approaches could be defined to address the planning of this sequence of viewpoints. The easiest one, random planification, consists in selecting the viewpoints according to a uniform distribution. Another approach derives from an entropy measure, which is used to compute the information gain of new observations given the current state of the system [1], [2], [3]. Entropy-based viewpoint selection has been proven to perform better than random planification. However, both planification and classification use a probabilistic modeling of the objects that should represent theoretical intra-class variations. This requires a rich amount of learning observations, which is often difficult to obtain. The system may also learn directly from visual interactions with the environment into a reinforcement learning framework. This approach presents the great advantage that the planning procedure is independent from the classifier. Refined modeling of objects is no longer necessary, thus fewer training data are needed. The so called Q-learning [4], [5] derives from the mathematical framework of Markov Decision Processes [6]. A classical way of solving such a problem is to use a recursive stochastic estimation of the action value via Monte Carlo. Although it is well suited to this task as the convergence to the desired solution is ensured, it requires a large amount of trajectory samples, restricting its use to cost-free sensing applications. The aim of this work is to extend the Q-learning to viewpoint planning when Monte-Carlo estimations are too costly. We present a novel approach for viewpoint planning based on a coupled design of a Q-learning procedure and the use of Kriging for both fitting and global optimization. The objective is to find the best approximation of the action-value function within a small sampling budget. In [7], Kriging and Bayesian optimization have been used to address a simultaneous localization and mapping problem under time and energy constraints. In the present paper, similar strategies are investigated to tackle the problem of viewpoint planning for active recognition.

The optimal sampling strategy is achieved by recursively fitting a parameterized surrogate function on the samples. This function assumes an underlying Gaussian process, thus making it cheap to evaluate. An expected improvement measure is derived from the current sampling so as to select the next exploration path, and the surrogate function is updated according to the new sample. Kriging has several useful properties. First, this unbiased predictor minimizes the squared prediction error and thus provides a reliable estimate. Second, it can be used to achieve global optimization, by combining the exploration of unknown areas with the exploitation of current knowledge. Third, the Kriging predictor is linear on the available observations, involving a very reduced computational cost. Finally, all the underlying parameters may be estimated by maximum-likelihood to fit the available data.

This paper is organized as follows. Section II reviews the main principles of active recognition and Q-learning. Section III describes the basics of Kriging and how it

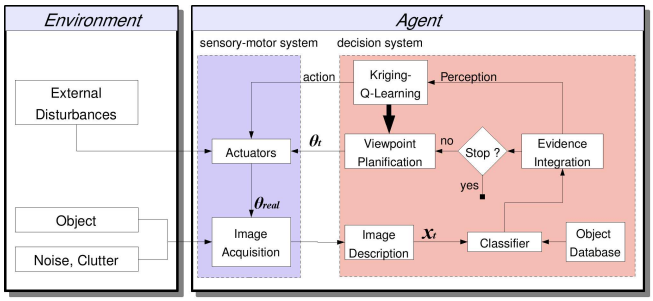

Figure 1. Closed-loop between the agent decision system and environment sensing. The decision system planifies on-line a sequence of observations from both the database ambiguity and its current state of knowledge

can be used to enhance viewpoint planning. Experimental results illustrate the proposed approach in Section IV, while conclusions and perspectives are reported in Section V.

II. Active Recognition

A. Multiple Observations Fusion

Let $\Omega = \{1, \dots K\}$ be a set of object classes. Given a sequence of T observations $\mathcal{X}_T = \{\mathbf{x}_1 \dots, \mathbf{x}_T\}$ such that $\mathbf{x} \in \mathbb{R}^m$, associated with their respective viewpoints $\mathcal{V}_T = \{\theta_1 \dots, \theta_T\}$, such that $\theta \in \mathbb{R}^d$, the class label ω should be inferred amongst Ω with minimum error. Since observations might be disturbed by intra class variation and noise (illumination changes, occlusions, clutter), recognition is expressed in a probabilistic framework. The state of the system at time t is $s_t = P(\omega \mid \mathcal{X}_t, \mathcal{V}_t)$. The state contains both the current class hypothesis and the viewpoints that have been visited. The decision of the object class is given by *a posteriori* maximizing

$$\omega* = \underset{\omega \in \Omega}{\operatorname{argmax}}\ P(\omega \mid \mathcal{X}_t, \mathcal{V}_t) \tag{1}$$

The integration of a next pair observation-viewpoint is defined as follows :

$$\begin{aligned} & P(\mathcal{X}_t, \mathcal{V}_t, \mathbf{x}_{t+1}, \theta_{t+1} \mid \omega) \\ & \quad = \quad P(\mathbf{x}_{t+1}, \theta_{t+1} \mid \mathcal{X}_t, \mathcal{V}_t, \omega) P(\mathcal{X}_t, \mathcal{V}_t \mid \omega) \\ & \quad = \quad P(\mathbf{x}_{t+1} \mid \theta_{t+1}, \mathcal{X}_t, \mathcal{V}_t, \omega) \\ & \qquad \times P(\theta_{t+1} \mid \mathcal{X}_t, \mathcal{V}_t, \omega) P(\mathcal{X}_t, \mathcal{V}_t \mid \omega) \end{aligned} \tag{2}$$

In the Markov assumption, the equality $P(\mathbf{x}_{t+1} \mid \theta_{t+1}, \mathcal{X}_t, \mathcal{V}_t, \omega) = P(\mathbf{x}_{t+1} \mid \theta_{t+1}, \omega)$ is applied. It yields

$$\begin{aligned} P(\mathcal{X}_t, \mathcal{V}_t, \mathbf{x}_{t+1}, \theta_{t+1} \mid \omega) &= P(\mathbf{x}_{t+1} \mid \theta_{t+1}, \omega) \\ \times P(\theta_{t+1} \mid \mathcal{X}_t, \mathcal{V}_t, \omega) & P(\mathcal{X}_t, \mathcal{V}_t \mid \omega) \end{aligned} \tag{3}$$

The evaluation of the probability

$$P(\mathbf{x}_{t+1} \mid \theta_{t+1}, \omega) \tag{4}$$

is performed by the classifier. A higher range in $\mathcal{X}$ and $\mathcal{V}$ could be considered without challenging the rest of the approach, but it would require more complexity in the design of the classifier, which is not the focus of the paper. The probability $P(\theta_{t+1} \mid \mathcal{X}_t, \mathcal{V}_t, \omega)$ indicates how to select the next viewpoint given the past observations of the object. Given the current state s_t at time t, an *estimation policy* π_e should be defined to favor actions leading to a fast and accurate estimation of s_t^*, which is the aim of the next section.

B. Q-learning Framework for learning viewpoint selection

A solution to (1) both independent from the classifier and the objects to identify is looked for in this paragraph. The selection of the next best action for recognition is based on a previous series of actions and decisions performed during the learning stage. In Q-learning [8], a closed loop linking acting and sensing is defined (see Figure 1). An action is defined by $a_t = (\theta_{t+1} - \theta_t) \in \mathbb{A}(s_t)$ where $\mathbb{A}(s_t)$ is the set of available actions in state s_t. A quality criterion $Q(s_t, a_t)$ is associated to each state-action pair $(s_t, a_t) \in \mathbb{S} \times \mathbb{A}(s_t)$. Q should reflect how good it is for the agent to select a_t for the future. The expected return of the subsequent steps is defined as a function of $N+1$ actions $\mathbf{a}_{N+1} = [a_t^{\mathrm{T}}, \dots, a_{t+N}^{\mathrm{T}}] \in \mathbb{A}(s_t) \times \dots \times \mathbb{A}(s_{t+N})$. Thus, it yields

$$R_t(\mathbf{a}_{N+1}) = \sum_{n=0}^{N} \gamma^n r_{t+n+1}(a_{t+n}) \quad \text{with } \gamma \in [0;1] \tag{5}$$

where $r_{t+n+1} : \mathbb{S} \times \mathbb{A}(s_{t+n}) \rightarrow \mathbb{R}$ is the reward associated with action a_{t+n}. The *discount rate* γ is a constant coefficient that controls the influence of each subsequent step. Note that N is theoretically equal to infinity. As future rewards are not known in advance, the action-value is given by the expected return

$$\widetilde{R}_t(\mathbf{a}_{N+1}) = \mathbb{E}_{\mathbf{P}_\theta}\left[R_t(\mathbf{a}_{N+1})\right] \tag{6}$$

The expectation is taken with respect to the pose uncertainty

$$\begin{aligned} \mathbf{P}_\theta \quad = \quad & \prod_{n=0}^{N} p(\mathbf{x}_{t+n+1} \mid \theta_{t+n+1}, s_{t+n}) \\ & \times p(\theta_{t+n+1} \mid \theta_{t+n}, a_{t+n}) \end{aligned} \tag{7}$$

Equation (6) can be rewritten in terms of a one-step recursive estimation of Q

$$Q(s_t, a_t) = \mathbb{E}_{\mathbf{P}_\theta}\left[r_{t+1}\right] + \gamma \max_{a_{t+1} \in \mathbb{A}(s_{t+1})} Q(s_{t+1}, a_{t+1}) \tag{8}$$

At the end of the learning stage, Q is supposed to converge to the optimal quality criterion Q^*. The optimal action policy is then defined by

$$\pi_e^*(s) = \underset{a}{\operatorname{argmax}}\ Q^*(s, a) \tag{9}$$

Numerous approaches have been proposed to estimate Q^* (for a detailed description, see [9]). An *Off-Policy Control*

has been selected here to solve the Q-learning problem. The *behavior policy* used to sample trajectories is unrelated to the *estimation policy*. The optimal action value $Q^*(s_t, a_t)$ is chosen as the maximum expected return of the N subsequent steps following a_t. It is a natural choice since it corresponds to the subsequence of actions that should be performed.
For computational tractability, we assume recognition to be viewed as an episodic task. Thus, N is a finite number of actions and, under this assumption, Q_N is the corresponding expected return. The learning of $Q_N(s_t, a_t)$ is then defined in four steps.

1) given a state s_t, generate a sequence of K actions a_t according to a sampling *behavior policy* π_b,
2) for each action a_t, generate a series of K' subsequences of actions according to a *behavior policy* π'_b,
3) for each subsequence of actions, calculate the expected return,
4) find the subsequence $\mathbf{a}_N = [a_{t+1}^{\mathrm{T}}, \ldots, a_{t+N}^{\mathrm{T}}] \in \mathbb{A}(s_{t+1}) \times \ldots \times \mathbb{A}(s_{t+N})$ that maximizes the expected return and update $Q_N(s_t, a_t)$ by using equation (8) as follows :

$$Q_N(s_t, a_t) = \mathbb{E}_{\mathbf{P}_\theta}\left[r_{t+1}\right] + \max_{\mathbf{a}_N} \mathbb{E}_{\mathbf{P}_\theta}\left[\sum_{n=1}^{N} \gamma^n r_{t+n+1}\right] \quad (10)$$

C. Basic Sampling Approach

A straightforward way to estimate $Q^*(s_t, a_t)$ from experience consists in a Monte Carlo evaluation of the maximum expected return. The two policies π_b and π'_b use a uniform sampling respectively over $\mathbb{A}(s_t)$ and $\mathbb{A}(s_{t+1}) \times \ldots \times \mathbb{A}(s_{t+N})$. Given an action a_t and K' subsequences of actions $\{\mathbf{a}_N^{(1)}, \ldots \mathbf{a}_N^{(K')}\}$, the action value $Q_N(s_t, a_t)$ is approximated by

$$Q_N(s_t, a_t) = \mathbb{E}_{\mathbf{P}_\theta}\left[r_{t+1}\right] + \max_{\mathbf{a}_N^{(k)},\ k \in [1;K']} \mathbb{E}_{\mathbf{P}_\theta}\left[\sum_{n=1}^{N} \gamma^n r_{t+n+1}\right] \quad (11)$$

For large values of K, K' and N, the quality criterion Q_N should converge to the optimal criterion Q^* (for a fixed value of γ), which is the idea underlying all Monte Carlo methods. As a reinforcement learning method, this Monte Carlo estimation process treats states and actions as discrete variables. To allow for a continuous estimation of Q, a natural way consists in defining a continuous function $\hat{Q}(s, a)$ by a weighted sum of the previously collected action-values as in [10]

$$\hat{Q}(s, a) = \frac{\sum_{(s', a') \in \Gamma(s)} d(a, a') Q(s', a')}{\sum_{(s', a') \in \Gamma(s)} d(a, a')} \quad (12)$$

where $\Gamma(s)$ defines the set of all state-action pairs whose state s' is equal to s. The term $d(a, a')$ is a distance function that measures how far is a from a'. It is generally computed by using a parametric kernel, usually Gaussian [10]. However, this approach suffers from several weaknesses. First, the method implies a large sampling number. An accurate computation of Q is thus very time-consuming. This turns out to be infeasible for learning the estimation policy when actions require a physical (thus slow and costly) move of the agent. Second, the interpolation involves the selection of both kernel and kernel parameter values. Optimal selection can be obtained by cross-validation with the learning objects, but it requires further simulation time.

III. Q-learning coupled with Kriging

To overcome the drawbacks of the Monte-Carlo method, we propose to use the potentialities of Kriging within the Q-learning framework in two points. First, a Kriging model with a smart sampling policy is used to obtain a dense estimate of Q during the learning stage and avoid the complex interpolation design from equation (12). This accounts for optimizing the behavior policy π_b. Second, a Kriging-based global optimization procedure furnishes a reliable estimation of the maximum expected future reward from equation (10). This accounts for optimizing the behavior policy π'_b. The design of the two policies are respectively indicated in Algorithms 1 and 2. The basics of Kriging and the underlying concepts of these two procedures are now described.

A. Basics of Kriging

Kriging has been given this name by the French geostatistician G. Matheron, to recognize the seminal influence of the work of D.G. Krige on the gold deposit of the Rand, in South Africa [11]. The Kriging approach is presented here with notations independent from those of the rest of the paper to remain generic.

Consider a process giving a scalar output y from inputs $\mathbf{u} \in \mathbb{U} \subset \mathbb{R}^d$. Given an initial small sample of size n, $\mathcal{U}_n = \{\mathbf{u}^{(1)}, ..., \mathbf{u}^{(n)}\}$ and the corresponding output results $\mathbf{y}_n = [y^{(1)}, ..., y^{(n)}]$, the aim of Kriging is to predict the value of $y(\cdot)$ at any unexplored point $\mathbf{u} \in \mathbb{U}$. For this purpose, the function $y(\cdot)$ is modeled as a Gaussian process $Y(\cdot)$ with mean function $\mathrm{avg}(\cdot)$ and covariance function $\mathrm{cov}(\cdot, \cdot)$. More specifically, $Y(\cdot)$ is written as

$$Y(\mathbf{u}) = \mathbf{f}^{\mathrm{T}}(\mathbf{u})\, \mathbf{b} + Z(\mathbf{u}) \quad (13)$$

where $\mathbf{f}(\mathbf{u})$ is some known regression function vector (usually chosen constant or polynomial in $\mathbf{u}$), $\mathbf{b}$ is a vector of unknown regression coefficients to be estimated, and $Z(\cdot)$ is a zero-mean Gaussian process with known (or parametrized) covariance function $\mathrm{cov}(\cdot, \cdot)$. Kriging is then the search for the *best linear unbiased predictor* (BLUP) of $Y(\cdot)$ [12].

The actual covariance $\mathrm{cov}(\cdot, \cdot)$ is most often unknown. It is expressed as

$$\mathrm{cov}(Z(\mathbf{u}^{(i)}), Z(\mathbf{u}^{(j)})) = \sigma_Z^2 C(\mathbf{u}^{(i)}, \mathbf{u}^{(j)}) \quad (14)$$

where σ_Z^2 is the process variance and $C(\cdot, \cdot)$ is a parametric correlation function. Both σ_Z^2 and the parameters of $C(\cdot, \cdot)$

must be chosen or estimated from the available data. Under a stationarity assumption, $C\left(\mathbf{u}^{(i)}, \mathbf{u}^{(j)}\right)$ depends only on the displacement vector $\mathbf{u}^{(i)} - \mathbf{u}^{(j)}$, denoted by $\mathbf{h}$ in what follows. A frequent choice of correlation function, also adopted in the present paper, is the *power exponential correlation function*

$$C\left(\mathbf{h}\right) = \exp\left(-\sum_{k=1}^{d}\left|\frac{h_k}{\beta_k}\right|^{p_k}\right) \tag{15}$$

where $0 < p_k \leq 2$, and h_k is the k-th component of $\mathbf{h}$. Note that with this choice, $C\left(\mathbf{h}\right)$ tends to 1 when $\mathbf{h}$ tends to $\mathbf{0}$. The β_k may be estimated from the data by maximum likelihood, to get what is known as *empirical Kriging*. A wide range of other choices for the correlation function is available [13].

Define $\mathbf{C}$ as the $n \times n$ matrix such that its (i, j) element $\mathbf{C}_{ij}$ is

$$\mathbf{C}_{ij} = C(\mathbf{u}^{(i)}, \mathbf{u}^{(j)}) \tag{16}$$

and $\mathbf{c}(\mathbf{u})$ as the n vector

$$\mathbf{c}\left(\mathbf{u}\right) = \left[C(\mathbf{u}, \mathbf{u}^{(1)}), ..., C(\mathbf{u}, \mathbf{u}^{(n)})\right]^{\mathrm{T}} \tag{17}$$

and $\mathbf{F}$ as the $(n \times \dim\, \mathbf{b})$ matrix

$$\mathbf{F} = [\mathbf{f}(\mathbf{u}^{(1)}), \ldots, \mathbf{f}(\mathbf{u}^{(n)})]^{\mathrm{T}} \tag{18}$$

The maximum-likelihood estimate $\widehat{\mathbf{b}}$ of the regression coefficients $\mathbf{b}$ from the available data $\{\mathcal{U}_n, \mathbf{y}_n\}$ is

$$\widehat{\mathbf{b}} = \left(\mathbf{F}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{F}\right)^{-1}\mathbf{F}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{y}_n \tag{19}$$

The predictor of the mean of the Gaussian process, at $\mathbf{u} \in \mathbb{U}$, is then given by

$$\widehat{Y}(\mathbf{u}) = \mathbf{f}^{\mathrm{T}}\left(\mathbf{u}\right)\widehat{\mathbf{b}} + \mathbf{c}\left(\mathbf{u}\right)^{\mathrm{T}}\mathbf{C}^{-1}\left(\mathbf{y}_n - \mathbf{F}\widehat{\mathbf{b}}\right) \tag{20}$$

This predictor is linear in $\mathbf{y}_n$ and interpolates the training data, as $\widehat{Y}(\mathbf{u}^{(i)}) = y^{(i)}$. Another interesting property of Kriging, which is crucial regarding the reliability of the estimate and global search for a maximum, is the possibility to compute the *variance of the prediction error* at $\mathbf{u} \in \mathbb{U}$ by

$$\widehat{\sigma}^2\left(\mathbf{u}\right) = \sigma_Z^2\left(1 - \mathbf{c}\left(\mathbf{u}\right)^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{c}\left(\mathbf{u}\right)\right) \tag{21}$$

B. Estimating Q by Kriging

The Kriging predictor is used to compute, for a given s_t, the value $Q_N(s_t, a_t)$ for any action $a_t \in \mathbb{A}(s_t)$, with a reduced sampling budget of K samples. The fitting proceeds in two main steps. An initialization step consists in choosing randomly by Latin Hypercube Sampling (LHS) n points in $\mathbb{A}(s_t)$ $(n < K)$ and computing their corresponding expected return. A Kriging predictor is then fitted on these data to obtain a first estimator of Q_N. The second step recursively finds the next sampling point for which the prediction error (21) is high, until the exhaustion of the sampling budget K. This way, the fitting minimizes the global prediction error (this is a natural property of Kriging), but also ensures that the local prediction error is small, in order to have a high-quality prediction with a reduced number of points. Algorithm 1 summarizes the procedure. At Step 2 and 6, the values $Q_N(s_t, a_t)$ of the sampled actions are computed by Algorithm 2, which achieves global optimization by Kriging, and which is now described.

Algorithm 1: Design of π_b by Kriging

Initialize: $s_t, \gamma, K, n < K, N$
Output: Fitted planning function

1 Choose $\mathcal{A}_n = \{a_t^{(1)}, ..., a_t^{(n)}\}$ by LHS in $\mathbb{A}(s_t)$;
2 Compute $\mathcal{Q}_n = \{Q_N(s_t, a_t^{(1)}), ..., Q_N(s_t, a_t^{(n)})\}$ using Algorithm 2;
3 **while** $n \leq K$ **do**
4 Fit the Kriging model on the known data points $\{\mathcal{A}_n, \mathcal{Q}_n\}$ according to equations (15)→(20);
5 Find $a^{(n+1)} = \arg\max_a \widehat{\sigma}^2\left(a\right)$;
6 Compute $Q_N(s_t, a_t)$, append it to $\mathcal{Q}_n$ and append $a^{(n+1)}$ to $\mathcal{A}_n$;
7 $n \leftarrow n + 1$;
8 **end**

Algorithm 2: Design of π_b' by Kriging

Initialize: K', n' and use initialized variables from Algorithm 1, notably the current action a_t
Output: estimation of the maximum expected return $\widetilde{R}_t^{\max}$

1 Choose $\boldsymbol{\mathcal{A}}_{n'} = \{\mathbf{a}_N^{(1)}, \ldots, \mathbf{a}_N^{(n')}\}$ by LHS;
2 $\mathbf{a}_{N+1} = [a_t^{\mathrm{T}}, \mathbf{a}_N]$;
3 Compute, according to (6), $\mathcal{R}_{n'} = \{\widetilde{R}_t(\mathbf{a}_{N+1}^{(1)}), \ldots, \widetilde{R}_t(\mathbf{a}_{N+1}^{(n')})\}$;
4 **while** $n' < K'$ **do**
5 Fit the Kriging model on the known data points $\{\boldsymbol{\mathcal{A}}_n, \mathcal{R}_{n'}\}$ according to equations (15)→(20);
6 Find $\widetilde{R}_t^{\max} = \max_{i=1\ldots n'}\{\widetilde{R}_t(\mathbf{a}_{N+1}^{(i)})\}$;
7 Find $\mathbf{a}_N^{(n'+1)} = \max_{\mathbf{a}_N}\{EI(\mathbf{a}, R_t^{\max})\}$;
8 Compute $\widetilde{R}_t(\mathbf{a}_{N+1}^{(n'+1)})$, append it to $\mathcal{R}_{n'}$ and append $\mathbf{a}_N^{n'+1}$ to $\boldsymbol{\mathcal{A}}_{n'}$;
9 $n' \leftarrow n' + 1$;
10 **end**

C. Bayesian Optimization for best sequence of actions

For a given pair state-action (s_t, a_t), a global optimization procedure should be employed to find the subsequence of N actions $\mathbf{a}_N = \left[a_{t+1}^{\mathrm{T}}, ...a_{t+N}^{\mathrm{T}}\right] \in \mathbb{A}(s_{t+1}) \times \ldots \times \mathbb{A}(s_{t+N})$ that maximizes the expected return $\widetilde{R}_t(\mathbf{a}_{N+1})$ where $\mathbf{a}_{N+1} = [a_t^{\mathrm{T}}, \mathbf{a}_N]$. For that purpose, we propose to use a global optimization algorithm based on Kriging and Expected Improvement, called EGO for *efficient global optimization* [13]. This algorithm uses the Kriging predictor (20) as a surrogate to find a better approximation of the global maximum of the expected return, taking advantage of the knowledge of the prediction error (21). The recursive procedure maximizes the Expected Improvement, whose principles are now outlined.

After an initial sampling of n' subsequences and corresponding computations of $\widetilde{R}_t$, the best available estimate for the global maximum is

$$R_t^{\max} = \max_{i=1\ldots n} \left\{ \widetilde{R}_t(\mathbf{a}_{N+1}) \right\} \tag{22}$$

The Expected Improvement is expressed in closed-form as

$$\mathrm{EI}(\mathbf{a}, R_t^{\max}) = \widehat{\sigma}(\mathbf{a}) \left[u\Phi(u) + \phi(u) \right] \tag{23}$$

where $u = \left(\widehat{Y} - R_t^{\max}(\mathbf{a}) \right) / \widehat{\sigma}(\mathbf{a})$. Φ is the cumulative distribution function and ϕ the probability density function of the normalized Gaussian distribution $\mathcal{N}(0,1)$. Maximizing Expected Improvement achieves a trade-off between local search (numerator of u) and the exploration of unknown areas (where $\widehat{\sigma}$ is high) and is therefore well suited for global optimization.

Our implementation of these algorithms is based on Sasena's toolbox SuperEGO [14] and uses the DIRECT optimization algorithm [15] to achieve Step 5 of Algorithm 1 and Step 6 of Algorithm 2.

IV. Experimental Results

A series of experiments has been performed to validate the Kriging approach. The illustrative environment, represented in Figure 2, allows a one-degree-of-freedom displacement along the azimuth, leading to a 1-dimensional action space. The database used for recognition is composed of 8 models of cars, as shown in Figure 3. For convenience, all observations have been collected in advance: objects have been presented on a turntable to a calibrated camera and images have been acquired at video rate, giving approximately 1000 images per object. For each object, 2 datasets have been considered, namely a learning set for training the planning policy and the classifier, and a test set for experiments. Each set corresponds to a 360-degree rotation. Note that this step does not challenge the use of our sampling approach since a motion cost could be defined for each observation. Each image has been centered into a sub-window of 100*100 pixels and annotated with the object class and the object pose relative to the camera. The classifier is based on the GLOH appearance descriptor of the objects [16]. Each descriptor is normalized by its sum in order to reduce the effects of illumination change. The dimension have been reduced by PCA to obtain a 5-d image descriptor. A Gaussian mixture density has then been computed for each class from the training set, and the probability (4) has been derived. This choice of classifier (which should have statistical properties) is independent from the rest of the process.

The learning of the estimation policy has been achieved by computing, for each class ω^*, a viewpoint planning function in order to disambiguate this class amongst the database. The reward r_t is defined as the difference between the posterior

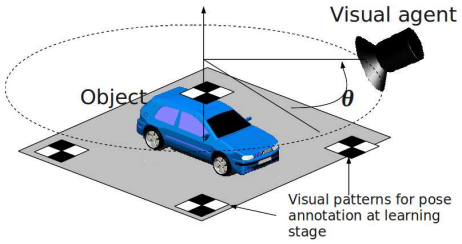

Figure 2. Experimental setup for active recognition.

of ω^* and the best current posterior,

$$r_t = P(\omega^* \mid \mathcal{X}_t, \mathcal{V}_t) - \max_{\omega \in \Omega,\ \omega \neq \omega^*} P(\omega \mid \mathcal{X}_t, \mathcal{V}_t) \tag{24}$$

The planning horizon is set to $N = 2$, but higher values could be considered without additional constraint. The parameters of π_b and π'_b (Algorithms 1 and 2) are $n = n' = 10$ and $K = K' = 15$. The advantage of Kriging interpolation over classical Gaussian kernel interpolation could be seen in Figure 4. For the same sampling budget, Kriging interpolation is far more sharply with no additional parameter to tune. For each action a_t, the expected return could be estimated by averaging the return of a set of trajectories generated according to the pose uncertainty distribution, as in [7]. The return of a single trajectory is considered here, since no pose uncertainty is assumed during the learning stage. The discount rate, γ, can be chosen experimentally to minimize the average number of observations needed to identify the object class (see Figure 5). The value $\gamma = 0.4$ has thus been chosen for the recognition task.

During the recognition stage, the first viewpoint is randomly chosen. At each step, the agent plans the next viewpoint using the planning function associated to the current object hypothesis. The pose uncertainty is modeled by a Gaussian distribution centered on the selected viewpoint, with standard deviation of 5 degrees. Recognition ends as soon as one of the posteriors exceeds a threshold $P_{\max} = 0.9$ or when the maximum number of allowed observations $O_{\max} = 20$ is reached. Figure 6 compares the recognition performance using the Kriging approach, the stochastic approach and random planification. These results are averaged over 50 tests for each class. The Kriging-based planning policy is shown to converge much faster and provides a significantly higher maximum performance rate.

V. Conclusions

We have presented a computationally-efficient approach for viewpoint planning in active recognition under a restricted sampling budget. Kriging sampling policies have been defined, achieving a trade-off between exploration and capitalization of the current best solution. Active recognition experiments on a database of 8 classes show that the method is significantly beneficial, providing higher performance and

Figure 3. Illustration of the database used for the experiments. Objects are represented by their 2D appearance.

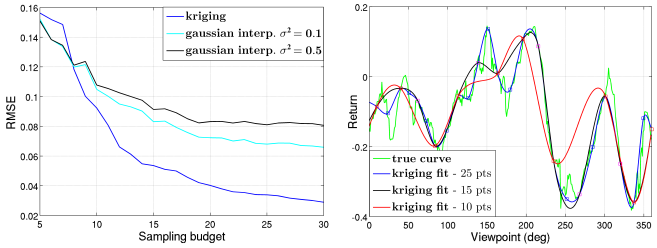

Figure 4. *Right:* Root mean square error between the true planning function and interpolated curves. *Left:* Fitting of a planning function (for $\gamma = 0$) by Kriging with different sampling budgets.

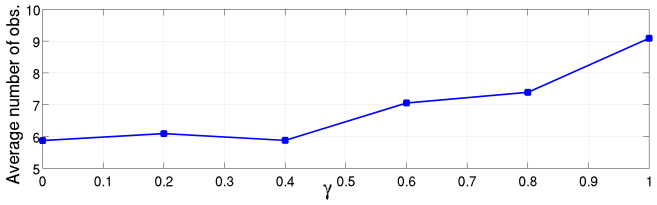

Figure 5. Influence of γ on the average number of observations needed for accurate recognition (obtained with kriging-based viewpoint planning).

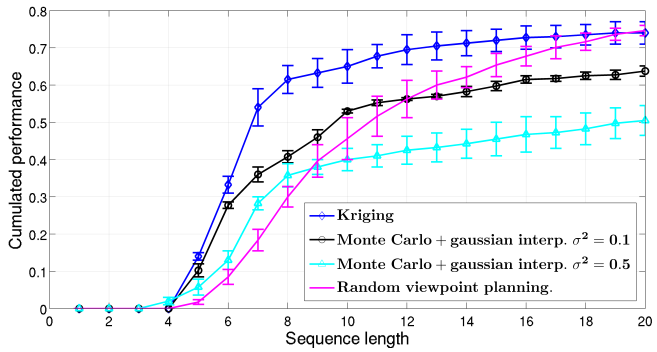

Figure 6. Average cumulated performance as a function of the sequence length for different planning strategies ($\gamma = 0.4$).

better estimation than classical stochastic methods. Future work will study the influence of the sampling budget on recognition accuracy, take into account pose uncertainty during the learning stage, optimize the decision threshold, and test other sampling strategies for viewpoint planning.

References

[1] J. Denzler and C. M. Brown, "Information theoretic sensor data selection for active object recognition and state estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 2, pp. 145–157, 2002.

[2] F. Callari and F. Ferrie, "Autonomous recognition: Driven by ambiguity," in *CVPR96*, 1996, pp. 701–707.

[3] T. Arbel and F. P. Ferrie, "Viewpoint selection by navigation through entropy maps," in *Seventh Int'l Conf. Computer Vision*, 1999.

[4] L. Paletta and A. Pinz, "Active object recognition by view integration and reinforcement learning," *Robotics and Autonomous Systems*, vol. 31, pp. 71–86, 2000.

[5] F. Deinzer, C. Derichs, H. Niemann, and J. Denzler, "Integrated viewpoint fusion and viewpoint selection for optimal object recognition," in *BMVC06*, 2006, p. I:287.

[6] S. D. Whitehead and D. H. Ballard, "Learning to perceive and act by trial and error," *Machine Learning*, vol. 7, pp. 45–83, 1991.

[7] R. Martinez-Cantin, N. de Freitas, E. Brochu, J. Castellanos, and A. Doucet, "A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot," *Autonomous Robots*, vol. 27, no. 2, pp. 93–103, 2009.

[8] C. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3, pp. 279–292, 1992.

[9] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. The MIT Press, March 1998.

[10] F. Deinzer, J. Denzler, and H. Niemann, "Classifier independent viewpoint selection for 3-d object recognition," *Mustererkennung*, vol. 22, pp. 237–244, 2000.

[11] G. Matheron, "Principles of geostatistics," *Economic Geology*, vol. 58, no. 8, p. 1246, 1963.

[12] J. Lefebvre, H. Roussel, E. Walter, D. Lecointe, and W. Tabbara, "Prediction from wrong models: the Kriging approach," *IEEE Antennas and Propagation Magazine*, vol. 38, no. 4, pp. 35–45, 1996.

[13] D. Jones, M. Schonlau, and W. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[14] M. Sasena, *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*. PhD thesis, University of Michigan, USA, 2002.

[15] D. Jones, C. Perttunen, and B. Stuckman, "Lipschitzian optimization without the Lipschitz constant," *Journal of Optimization Theory and Applications*, vol. 79, no. 1, pp. 157–181, 1993.

[16] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, 2005.