Published in :
British Machine Vision Conference 2006
Volume One
Editors : M. J. Chantler, E. Trucco, R. B. Fisher
BMVA
ISBN : Full Set 1-904410-14-6
Pages : 287-296

Manuscript

Integrated Viewpoint Fusion and Viewpoint Selection for Optimal Object Recognition

Frank Deinzer *, Christian Derichs *, Heinrich Niemann
Chair for Pattern Recognition, Department of Computer Science,
University of Erlangen-Nuremberg, Martensstr. 3, 91058 Erlangen
{deinzer,derichs,niemann}@informatik.uni-erlangen.de

Joachim Denzler
Chair for Computer Vision, Friedrich-Schiller-University Jena
Ernst-Abbe-Platz 2, 07743 Jena, denzler@informatik.uni-jena.de

Abstract

In the past decades, most object recognition systems were based on passive approaches. But in the last few years a lot of research was done in the field of active object recognition, that is selectively moving a sensor/camera around a considered object in order to acquire as much information about it as possible. In this paper we present an active object recognition approach that solves the problem of choosing optimal views (viewpoint selection) and iteratively fuses the gained information for an optimal 3D object recognition (viewpoint fusion) in an integrated manner. Therefore, we apply a method for the fusion of multiple views with respect to the knowledge about the assumed camera movement between them.

For viewpoint selection we formally define the choice of additional views as an optimization problem. We show how to use reinforcement learning for this purpose and perform a training without user interaction. In this context we focus on the modeling of continuous states, continuous, one-dimensional actions and supporting rewards for an optimized recognition of real objects.

The experimental results show that our combined viewpoint selection and viewpoint fusion approach is able to significantly improve the recognition rates compared to passive object recognition with randomly chosen views.

1 Introduction

The results of 3D object classification and localization strongly depend on the images which have been taken of the object. Based on ambiguities between objects in the data set, some views might result in better recognition rates, others in worse. For difficult data sets, usually more than one view is necessary to decide reliably on a certain object class. Viewpoint selection tackles exactly the problem of finding a sequence of optimal views to increase classification and localization results by avoiding ambiguous views or by sequentially ruling out possible object hypotheses. The optimality is not only defined with respect to the recognition rate, but also with respect to the number of views necessary

*This work was funded by the German Science Foundation(DFG) under grant SFB 603/TP B2. Only the authors are responsible for the content.

to get reliable results. The number of views should be as small as possible to delimit viewpoint selection from randomly taking a large number of images.

In this paper, we apply an approach for active object recognition, namely viewpoint selection based on reinforcement learning, to the recognition of real objects (see figure 2). The approach shows some major benefits: First, the optimal sequence of views is learned automatically in a training step without any user interaction. Second, the approach performs a fusion of the generated views, where the fusion method does not depend on a special classifier. This makes it reasonable for a very wide range of applications. Reinforcement learning is usually done in discrete state and action spaces what is not profitable in our viewpoint selection problem. So third, we will show how to extend the classical reinforcement learning approaches to continuous state and action spaces. The viewpoint selection is the *active* (acting) part of the object recognition.

One important aspect besides the choice of the best viewpoint is the fusion of the classification and localization results of a sequence of viewpoints — the *object recognition* part of our approach. In the context of a viewpoint selection the problem arises how to fuse the collected views to finally return a classification and localization result. Also a sequence of views will improve the recognition rate in general if a decent fusion scheme is applied. In this paper we apply a fusion scheme which is based on the Condensation Algorithm [7]. There, [3] has proven that the latter is able to deal with multimodal distributions over the class and pose space of the objects and with the uncertainty of the camera movement during the viewpoint selection process. Viewpoint selection and combination of the resulting views has been investigated in the past in several applications. Examples are 3D reconstruction [11] or optimal segmentation of image data [9]. In object recognition, some active approaches have already been discussed as well. [12] plans the next view for a movable camera based on probabilistic reasoning. The active part is the selection of a certain area of the image for feature selection. The selected part is also called the receptive field [13]. Compared to our approach, no camera movement is performed, neither during training nor during testing. Thus, the modeling of viewpoints in a continuous 3D space is also avoided. Instead of following [4], where the optimal action is directly searched for by maximizing the mutual information between the observation and the state to be estimated, we use reinforcement learning for this purpose.

In section 2 we will show how the viewpoint fusion of multiple views can be done based on recursive density propagation in a continuous state space. This section describes the *object recognition* part of our solution. The reinforcement learning approach for viewpoint selection is presented in section 3. It is the *active* component of our object recognition. The experimental results in section 4 show that the presented approach is able to learn an optimal strategy for viewpoint selection that recordes only the minimal number of images. The paper concludes with a summary and an outlook to future work in section 5.

2 Fusion of Multiple Views

In active object recognition, a series of observed images $\langle \boldsymbol{f} \rangle_t = \boldsymbol{f}_t, \boldsymbol{f}_{t-1}, \ldots, \boldsymbol{f}_0$ of an object is given together with the camera movements $\langle \boldsymbol{a} \rangle_{t-1} = \boldsymbol{a}_{t-1}, \ldots, \boldsymbol{a}_0$ between these images. Based on these observations of images and movements one wants to draw conclusions for a non-observable object state $\boldsymbol{q}_t$ which must contain both the *discrete* class Ω_κ and the *continuous* pose $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_J)^T$ of the object, leading to the definition $\boldsymbol{q}_t = (\Omega_\kappa, \phi_1^t, \ldots, \phi_J^t)^T$. The actions $\boldsymbol{a}_t$ consist of the *relative* camera movement with J degrees of freedom, in the following written as $\boldsymbol{a}_t = (\Delta\phi_1^t, \ldots, \Delta\phi_J^t)$.

Our solution to that problem is based on a probabilistic framework: In the context of a Bayesian approach, the knowledge on the object state is given in form of the a posteriori density

$$p(\boldsymbol{q}_t | \langle \boldsymbol{f} \rangle_t, \langle \boldsymbol{a} \rangle_{t-1}) \quad . \tag{1}$$

That requires to have all observed actions $\langle \boldsymbol{a} \rangle_{t-1}$ and images $\langle \boldsymbol{f} \rangle_t$ available. For practical applications, density (1) is not suitable since one would prefer a form that allows for a continuous integration of new images and actions into the present knowledge. This is possible with the following recursive formulation of (1) :

$$p\left(\boldsymbol{q}_t | \langle \boldsymbol{f} \rangle_t, \langle \boldsymbol{a} \rangle_{t-1}\right) = \frac{p\left(\boldsymbol{q}_t, \langle \boldsymbol{f} \rangle_t, \langle \boldsymbol{a} \rangle_{t-1}\right)}{p\left(\langle \boldsymbol{f} \rangle_t, \langle \boldsymbol{a} \rangle_{t-1}\right)} \tag{2}$$

$$= \frac{p\left(\boldsymbol{f}_t | \boldsymbol{q}_t, \langle \boldsymbol{f} \rangle_{t-1}, \langle \boldsymbol{a} \rangle_{t-1}\right) p\left(\boldsymbol{q}_t | \langle \boldsymbol{f} \rangle_{t-1}, \langle \boldsymbol{a} \rangle_{t-1}\right)}{p\left(\boldsymbol{f}_t | \langle \boldsymbol{f} \rangle_{t-1}, \langle \boldsymbol{a} \rangle_{t-1}\right)} \tag{3}$$

$$= \frac{1}{c} \cdot p\left(\boldsymbol{f}_t | \boldsymbol{q}_t\right) p\left(\boldsymbol{q}_t | \langle \boldsymbol{f} \rangle_{t-1}, \langle \boldsymbol{a} \rangle_{t-1}\right) \tag{4}$$

$$= \frac{1}{c} \cdot p\left(\boldsymbol{f}_t | \boldsymbol{q}_t\right) \int p\left(\boldsymbol{q}_t | \boldsymbol{q}_{t-1}, \boldsymbol{a}_{t-1}\right) \cdot p\left(\boldsymbol{q}_{t-1} | \langle \boldsymbol{f} \rangle_{t-1}, \langle \boldsymbol{a} \rangle_{t-2}\right) d\boldsymbol{q}_{t-1} \quad . \tag{5}$$

The definition of the conditional probability $p(A|B) = p(AB)/p(B)$ leads directly to (2) and the multiplication theorem for probability densities to (3). The denominator in (3) is constant c since we assume arbitrary images to appear with an equally distributed probability. In (4) the Markov assumption $p(\boldsymbol{f}_t | \boldsymbol{q}_t, \langle \boldsymbol{f} \rangle_{t-1}, \langle \boldsymbol{a} \rangle_{t-1}) = p(\boldsymbol{f}_t | \boldsymbol{q}_t)$ is applied. The formulation of $p(\boldsymbol{q}_t | \langle \boldsymbol{f} \rangle_{t-1}, \langle \boldsymbol{a} \rangle_{t-1})$ as an integral in (5) results from the total probability theorem. Obviously the probability (5) only depends on the camera movement $\boldsymbol{a}_{t-1}$. The inaccuracy of $\boldsymbol{a}_{t-1}$ is modeled within the state transition component $p(\boldsymbol{q}_t | \boldsymbol{q}_{t-1}, \boldsymbol{a}_{t-1})$.

So finally, (5) allows for what we requested: The integration of a new image $\boldsymbol{f}_t$ and a new action $\boldsymbol{a}_{t-1}$ into the current knowledge about the object given with the density $p\left(\boldsymbol{q}_{t-1} | \langle \boldsymbol{f} \rangle_{t-1}, \langle \boldsymbol{a} \rangle_{t-2}\right)$. The recursion in (5) can be continued until the distribution $p(\boldsymbol{q}_0)$ of the initial state is reached. This distribution contains the initial knowledge about the object and its pose.

The classic approach for solving a recursive density propagation problem as given in (5) is the Kalman Filter [8]. But in computer vision, the necessary assumption for the Kalman Filter, $p(\boldsymbol{f}_t | \boldsymbol{q}_t)$ being normally distributed, is often not valid. Another approach for the complicated handling of such multi-modal densities are the so called particle filters. The basic idea is to approximate the a posteriori density by a set of weighted samples. In our approach we apply the Condensation Algorithm [7] which uses a sample set Y_t to approximate the multi-modal probability distribution (1) by M samples $y_t^i = \{\boldsymbol{x}_t^i, p_t^i\}$. Each sample y consists of the point $\boldsymbol{x} = (\Omega_\kappa, \phi_1, \ldots, \phi_J)$ within the state space and the weight p for that sample with the condition that $\sum_i p_t^i = 1$.

The Condensation Algorithm starts with an initial sample set Y_0 representing $p(\boldsymbol{q}_0)$. In our application we distribute the samples uniformly over the state space as we will have no knowledge given about the objects before observing the first image.

For the generation of a new sample set Y_t, M new samples y_t^i are

1. drawn from Y_{t-1} with a probability proportional to the sample weightings;
2. propagated with a necessarily predetermined sample transition model according to $p(\boldsymbol{q}_t | \boldsymbol{q}_{t-1}, \boldsymbol{a}_{t-1})$ in (5). In this work we assume the sample transition to be

$$\boldsymbol{x}_t^i = \boldsymbol{x}_{t-1}^i + (0, r_1, \ldots, r_J)^T \quad \text{with} \quad r_j \sim \mathcal{N}(\Delta\phi_j^t, \sigma_j) \quad . \tag{6}$$

Equation (6) models the inaccuracy of the camera movement under the assumption that the errors of the camera movements are independent between the degrees of freedom. The variance parameters σ_j of the Gaussian transition noise have to be defined in advance.

3. evaluated in the image by
$$p(\boldsymbol{f}_t|\boldsymbol{x}_t^i) \quad . \tag{7}$$
This evaluation is performed by the classifier. The only requirement for the classifier is its statistical expandability in order to calculate (7).

With these sample sets, a classification is possible at each time step by marginalization over all possible poses for each class:

$$p(\Omega_\kappa) = \int_{\boldsymbol{\phi}} p\left((\Omega_\kappa, \phi_1, \ldots, \phi_{N_\phi})^{\mathrm{T}}|\boldsymbol{f}_t, \boldsymbol{a}_{t-1}, \ldots\right) d\boldsymbol{\phi} \quad . \tag{8}$$

As we work on sample sets this can be done by a simple summation. Certainly, there are better ways to represent a single probability density, like with a mixture of Gaussians. But when propagating those densities regarding the information fusion, we again have to evaluate it at discrete positions and thus we would not cut down any effort compared to the particle representation.

In the context of the viewpoint selection (see section 3), the densities which are represented by sample sets have to be evaluable at any continuous position. The direct evaluation of them beneath the positions given by the individual samples is not possible. For that purpose, we discussed various ways of continuous representations [1], but in this work we use only the Parzen estimation to evaluate our non-parametric densities.

3 Viewpoint Selection

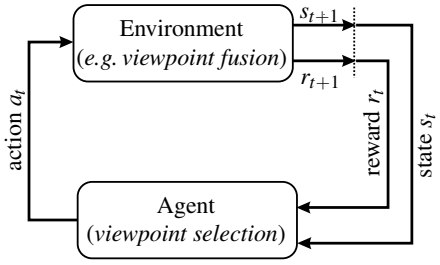


Figure 1: Reinforcement learning loop

A straight forward and intuitive way to formalize the problem of viewpoint selection is shown in Figure 1. A closed loop between sensing s_t and acting $\boldsymbol{a}_t$ can be seen. The chosen *action* $\boldsymbol{a}_t$ corresponds to the executed camera movement as described in section 2, the *presumption state*

$$s_t = p(\boldsymbol{q}_t|\langle \boldsymbol{f} \rangle_t, \langle \boldsymbol{a} \rangle_{t-1}) \tag{9}$$

is the density as given in (1). Additionally, the classification module returns a so called *reward* r_t, which measures the quality of the chosen action and the resulting viewpoint, respectively. It is well known that the definition of the reward is an important aspect as this reward shall model the goal that has to be reached. A proper definition for the reward in the context of our viewpoint selection problem is given later in this paper.

At time t during the decision process, i.e. the selection of a sequence of actions, the goal will be to maximize the accumulated and weighted future rewards, called the *return*

$$R_t = \sum_{n=0}^{\infty} \gamma^n r_{t+n+1} \quad \text{with } \gamma \in [0;1]\,. \tag{10}$$

The weighting factor γ defines how much influence a future reward will have on the overall return R_t at time $t+n+1$, because it appoints the decrease of the rewards' weight γ^n when increasing the step indicator n. A value of $\gamma = 0.0$ would mean that the return depends only on the reward of the next step. In contrast, $\gamma = 1.0$ would mean that all following rewards have the same influence on the return. Of course, the future rewards cannot be observed at time step t. Thus, the following function, called the *action-value function*

$$Q(s, \boldsymbol{a}) = E\{R_t | s_t = s, \boldsymbol{a}_t = \boldsymbol{a}\} \tag{11}$$

is defined. It describes the expected return when starting at time step t in presumption state s with action $\boldsymbol{a}$. In other words, the function $Q(s, \boldsymbol{a})$ models the expected quality of the chosen camera movement $\boldsymbol{a}$ for the future, if the viewpoint fusion has returned s before.

One of the demands defined in section 1 is that the selection of the most promising view should be learned without user interaction. Reinforcement learning provides many different algorithms to estimate the action value function based on a trial and error method [14]. Trial and error means that the system itself is responsible for trying certain actions in a certain presumption state. The result of such a trial is then used to update $Q(\cdot, \cdot)$.

In reinforcement learning a series of *episodes* is performed: Each episode k consists of a sequence of state/action pairs $(s_t, \boldsymbol{a}_t), t \in \{0, 1, \ldots, T\}$, where the performed action $\boldsymbol{a}_t$ in presumption state s_t results in a new state s_{t+1}. A final presumption state s_T is called the terminal state, where a predefined goal—like a reliable classification—is reached. During the episode, new returns $R_t^{(k)}$ are collected for those state/action pairs $(s_t^k, \boldsymbol{a}_t^k)$ which have been visited at time t during the episode k. At the end of the episode the action-value function is updated.

We are still missing the definition of the reward r_t. In the context of this paper the following two different definitions of rewards are compared.

- A way to model the goal is to define a reward that has a value of 0 except when reaching the terminal presumption state:

$$r_t = \begin{cases} C & s_t \text{ is terminal state}, C > 0 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

  This approach has the advantage that the goal is defined very clearly. But one has to decide when the confidence of the classification is high enough to stop the viewpoint selection. If this decision is hard to make, no proper strategy will be learned. The advantage is that (12) maximizes the return of an episode with short episodes (at least for $\gamma \neq 0, \gamma \neq 1$). So this strategy promises to look for episodes with only a minimal number of views independent of the chosen C.

- Another approach follows the idea that viewpoints that increase the information observed so far should have large values for the reward. A well-known measure for expressing the informational content that fits our requirements is the entropy

$$r_t = -H(s_t) = -H\left(p(\boldsymbol{q}_t | \langle \boldsymbol{f} \rangle_t, \langle \boldsymbol{a} \rangle_{t-1})\right) \quad . \tag{13}$$

  In that sense, the reward expresses the gain of knowledge about the object. (13) has the advantage that the goal is to maximally improve the classification in every time step instead of eventually reaching the stop criterion with even suboptimal classification values.

It is worth noting that the reward might also include costs for the camera movements, so that large movements are punished. In this paper we neglect costs for camera movement for the time being. But work in this area has be done, for example, in [5].

Most of the algorithms in reinforcement learning treat the object states and actions as discrete variables. Of course, in viewpoint selection parts of the state space (the pose of the object) and the action space (the camera movements) are continuous. For that reason, the common reinforcement learning techniques cannot be applied directly for our viewpoint selection problem. It is necessary to find a way to allow for the usage of the required continuous object states and actions. [3] proposed an approximation

$$\widehat{Q}(s,\boldsymbol{a}) = \frac{\sum_{(s',\boldsymbol{a}')} K\left(d\left(\theta(s,\boldsymbol{a}),\theta(s',\boldsymbol{a}')\right)\right) \cdot Q(s',\boldsymbol{a}')}{\sum_{(s',\boldsymbol{a}')} K\left(d\left(\theta(s,\boldsymbol{a}),\theta(s',\boldsymbol{a}')\right)\right)}, \tag{14}$$

of the action-value function based on a weighted sum of the action-values $Q(s',\boldsymbol{a}')$ of all previously collected state/action pairs $(s',\boldsymbol{a}')$.

Thereby, the **transformation function** $\theta(s,\boldsymbol{a})$ transforms a presumption state s with a known action $\boldsymbol{a}$ with the intention of bringing a state to a "reference point" (required for the distance function in the next item). In the context of the current definition of the presumption states from (9) it can be seen as a density transformation

$$\theta(s_t,\boldsymbol{a}_t) = \theta\left(p(\boldsymbol{q}_t|\langle\boldsymbol{f}\rangle_t,\langle\boldsymbol{a}\rangle_{t-1}),\boldsymbol{a}_t\right) = \det\left(\boldsymbol{J}_{\boldsymbol{\zeta}_{\boldsymbol{a}_t}^{-1}}(\boldsymbol{q}_t)\right) p\left(\boldsymbol{\zeta}_{\boldsymbol{a}_t}^{-1}(\boldsymbol{q}_t)|\langle\boldsymbol{f}\rangle_t,\langle\boldsymbol{a}\rangle_{t-1}\right) \tag{15}$$

with $\boldsymbol{\zeta}_{\boldsymbol{a}}^{-1}(\boldsymbol{q}) = (q_1, q_2+a_1, \ldots, q_{m+1}+a_m)^T$. [3] showed that $\boldsymbol{J}_{\boldsymbol{\zeta}_{\boldsymbol{a}}^{-1}}(\boldsymbol{q}) = \boldsymbol{I}$ is true for that application. This means that this transformation simply performs a shift of the density.

The **distance function** $d(\cdot,\cdot)$ is used to calculate the distance between two states via the *extended Kullback-Leibler Distance* $d_{\mathrm{EKL}}(s_n,s'_m) = d_{\mathrm{KL}}(s_n,s'_m) + d_{\mathrm{KL}}(s'_m,s_n,)$ with

$$d_{\mathrm{KL}}(s_n,s'_m) = \int p(\boldsymbol{q}|\langle\boldsymbol{f}\rangle_n,\langle\boldsymbol{a}\rangle_{n-1}) \log \frac{p(\boldsymbol{q}|\langle\boldsymbol{f}\rangle_n,\langle\boldsymbol{a}\rangle_{n-1})}{p(\boldsymbol{q}|\langle\boldsymbol{f}'\rangle_m,\langle\boldsymbol{a}'\rangle_{m-1})} d\boldsymbol{q} \quad . \tag{16}$$

Please note that in general there is no analytic solution for d_{EKL}, but as we represent our densities as sample sets anyway (see section 2) there are well-known ways to approximate d_{EKL} by Monte Carlo techniques [1].

The **kernel function** $K(\cdot)$ finally weights the calculated distances. A suitable kernel function is, for example, the Gaussian $K(x) = \exp(-x^2/D^2)$ where D denotes the width of the kernel. Low values for D will result in very detailed approximations well-suited if a lot of action-values $Q(s',\boldsymbol{a}')$ are available. If the system has so far observed only very few action-values, high values for D are the better choice as they give smoother approximations.

Using (14), the viewpoint selection problem of finding the particularly optimal action $\boldsymbol{a}^*$, i.e. the computation of the policy π, can now be written as a continuous optimization problem

$$\pi(s) = \boldsymbol{a}^* = \mathrm{argmax}_{\boldsymbol{a}}\, \widehat{Q}(s,\boldsymbol{a}) \quad . \tag{17}$$

It is solved in this work by applying a global Adaptive Random Search Algorithm [15] followed by a local Simplex. The application of reinforcement learning to the view planning problem is no additional effort, but essential. In comparison, a simple linear scan of best views would completely omit the current knowledge, e.g. the information of all images fused so far.

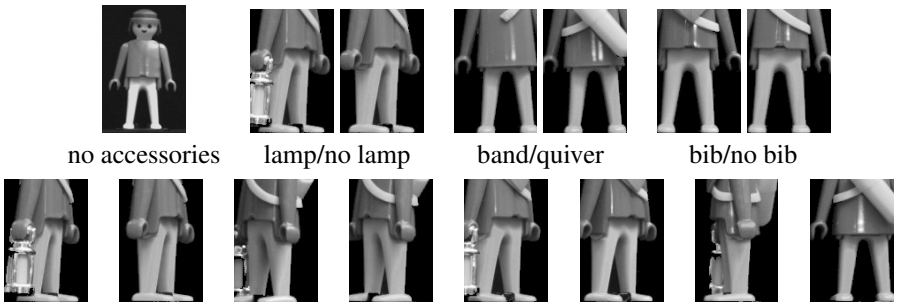


Figure 2: Upper row: Significant accessories of the manikins. Lower row: One exemplary image for each object class as summarized in Table 1.

Object	o1	o2	o3	o4	o5	o6	o7	o8
bib/no bib	no bib	no bib	no bib	no bib	bib	bib	bib	bib
band/quiver	band	band	quiver	quiver	band	band	quiver	quiver
lamp/no lamp	lamp	no lamp	lamp	no lamp	lamp	no lamp	lamp	no lamp

Table 1: List of all object classes. The accessories are shown in Figure 2.

4 Experimental Evaluation

The primary goal in the experiments was to show how the recognition rates improve when using our approach for viewpoint selection compared to passive object recognition approaches and the viewpoint fusion with randomly chosen images. In our previous work [1] we concentrated on a viewpoint selection with *synthetically generated object classes*, providing clearly defined ambiguities and completely traceable passive object recognition results. In this work we focus on the practicability of our viewpoint selection problem regarding the definition of the reward and the relevant parameters for much more complicated *real objects*.

Our data set consists of eight toy manikins as shown in Figure 2 and summarized in Table 1. The manikins have been selected in a way that they are strongly ambiguous: They can hold a lamp or not, can have a bib or not and carry a quiver or a simple band instead. Combining these three types of features gives a total of eight different manikins. Additionally, the camera is zoomed such close that parts of the manikins drop out of the visible area, e.g. the lamp in the hand is not visible in the frontal or backside view. The reader should note that there does not exist one unique viewpoint, which allows to distinguish all eight objects.

In this work we use a classifier based on the eigenspace approach introduced by [10]. But in our fusion approach we need a statistical measure in (7) to evaluate the particles. The work of [10] gives only a *distance* to some prototype features. An easy way to obtain the required statistical measure is to put these distances into a Gibbs distribution. The evaluation of (7) is realized that way in this work. For other application scenarios and other classifiers the presented fusion approach already showed to work very well [6, 2].

In our setup the camera is restricted to a movement around the object on a circle. This leads to a one-dimensional definition of the samples (according to section 2) $\boldsymbol{x} = (\Omega_\kappa, \phi_1)$ with actions $\boldsymbol{a}_t = (\Delta\phi_1^t), \Delta\phi_1^t \in [0^\circ, 360^\circ]$. In total, our sample set has a fixed size of $M = 2880$ samples.

Planning type	$n=0$	$n=1$	$n=2$	$n=3$	$n=4$	Length
No viewpoint selection, random views	62.5%	76.5%	83.7%	84.7%	85.0%	n.a.
Viewpoint selection, reward: fixed value (12)	61.8%	96.1%	99.7%	99.9%	99.9%	2.81
Viewpoint selection, reward: entropy (13)	62.4%	94.7%	98.1%	99.0%	99.3%	3.73

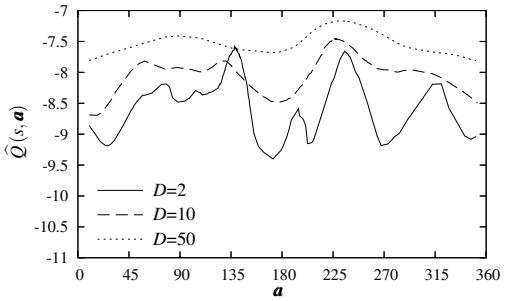
Table 2: Results of the recognition rates of randomly chosen views compared to views chosen by the viewpoint selection with two different types of reward. The percentages show the results after $n=0$, $n=1$, ..., $n=4$ fused views using kernel parameter $D=5.0$. The "Length" column shows the average length of the planned episodes.

In our experiments we evaluated scenarios that differ in the type of applied reward (fixed value according to (12) or entropy-based as given in (13)). In a training phase a total of 400 episodes with a length of 8 steps each and randomly chosen camera movements were performed for every combination of object class and type of reward.

The evaluation was performed on the results of a total of 2000 episodes with randomly chosen classes and starting views. A viewpoint selection episode finished as soon as the stop criterion was reached: The absolute difference between the probabilities of the best and the second best class hypothesis (calculated using (8)) must be at least 60%. Regarding the applied reinforcement learning approach this is just a meaningful value when using the entropy based reward, but in contrast it additionally becomes an important parameter when being driven by the fixed value. In that case, longer episodes which would be caused by a stricter stop criterion result in episodes' early steps ($n << T$) with clearly less significance. For this reason the afore said stopping value had to be found by empirical optimization, thus reflecting the drawback of the usage of the fixed value reward. In all experiments we set the parameter $\gamma = 0.5$ in (10). We discussed the influence of this parameter to the learned strategies extensively in [1]. A repetition of these conclusions will be omitted here.

The results of the experiments with a value of $D=5.0$ (see below) for the parameter in the Gaussian kernel function are shown in Table 2. As one would expect, the recognition rates are improved in all three presented scenarios when the number of views increases. Since the fusion integrates new information independently of the chosen view planning strategy, classification results always tend towards a satiation value for large n. The results for the new views calculated by the viewpoint selection outperform the ones with randomly selected views and show that our approach allows for a very reliable classification. The recognition rates for $n=0$ can be seen as the results for just a single view. They differ between the proposed methods because starting views where chosen randomly for each new episode during evaluation.

The main difference between the two viewpoint selection variations, fixed value and entropy, is that the mean length of the planned view sequences is lower for the fixed value reward. The reason is that this reward directly targets the viewpoint selection to minimize the length of the sequences since an early achievement of the stop criterion maximizes the return in (10). In contrast, the entropy based reward does not depend on the stop criterion at all. Using it as the reward targets the system to minimize the entropy of the density coming from the viewpoint fusion by increasing the information content of it. As it can be seen this does not necessarily minimize the sequence length, but gains comparably



Planning type	Reward	
	fixed value	entropy
$D=2$	99.8%	93.0%
$D=5$	99.9%	99.0%
$D=10$	99.8%	93.9%
$D=20$	93.4%	88.8%
$D=50$	90.8%	87.0%

Figure 3: Influence of the parameter D to the approximation of $\widehat{Q}(s,\boldsymbol{a})$ for an initial view to the back of a toy manikin (left) and to the recognition rate after $n=4$ views (right).

good classification results.

In our viewpoint selection approach we have one important free parameter, the D in the Gaussian kernel function. This parameter controls how detailed (for small values of D) or coarse the approximation of the action-value function will be. The effect of this parameter is shown in the graph of Figure 3 where the approximation of the action-value function is drawn for different values of D. The impact of this parameter to the recognition rate is shown in the right table of Figure 3. As one can see, the recognition rate highly depends on a proper value for D. The reason is that if D is too large, the approximation (14) is getting too coarse, losing important details. This can easily be understood if one knows that, for example, a good view to determine whether the manikin holds the lamp or not is a position where it stands diagonally to the camera. This necessary level of detail is lost for higher values of D.

5 Summary and Future Work

In this paper we have applied our general framework for viewpoint selection and viewpoint fusion to a real world classification problem. The main aspects of our viewpoint selection and fusion approach are that it works in continuous state and action spaces and is independent of the chosen statistical classifier. Furthermore the system can be trained automatically without user interaction. The experimental results on the toy manikins give classification results that outperform those one would achieve by taking views randomly or by passive object recognition approaches.

Future work will point at the practical implementation of higher dimensional state spaces which are theoretically already manageable. But due to the massive usage of memory for storing the action-value functions there might be need of other, highly information efficient, reinforcement learning methods for keeping this database as small as possible. Besides this we have a strong demand on improving the underlying model that provides us with the possibility of calculating probabilities like (1). So upcoming research will try to find a way for extending this model at runtime in an optimal manner concerning the classification results. This way we think to be guided step by step to a system that can perform online-learning, in particular not requiring the foregoing training phase any more and resolving the competing demands on the camera movement for building the model and learning best views all by itself.

References

[1] F. Deinzer, J. Denzler, Ch. Derichs, and H. Niemann. Aspects of optimal viewpoint selection and viewpoint fusion. In *Computer Vision – ACCV 2006*, volume 3852, pages 902–912, Hyderabad, India, 2006.

[2] F. Deinzer, J. Denzler, and H. Niemann. Classifier Independent Viewpoint Selection for 3-D Object Recognition. In *Mustererkennung 2000, 22. DAGM-Symposium, Kiel*, pages 237–244, Kiel, Germany, 2000. Springer.

[3] F. Deinzer, J. Denzler, and H. Niemann. Viewpoint Selection – Planning Optimal Sequences of Views for Object Recognition. In *Computer Analysis of Images and Patterns*, pages 65–73, Groningen, Netherlands, 2003. Springer.

[4] J. Denzler and C.M. Brown. Information theoretic sensor data selection for active object recognition and state estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):145–157, 2002.

[5] Christian Derichs, Frank Deinzer, and Heinrich Niemann. Cost integration in multi-step viewpoint selection for object recognition. In *Proceedings of the International Conference on Machine Learning and Data Mining*, pages 415–425, Leipzig, Germany, 2005. Springer.

[6] M. Grzegorzek, F. Deinzer, M. Reinhold, J. Denzler, and H. Niemann. How Fusion of Multiple Views Can Improve Object Recognition in Real-World Environments. In *Vision, Modeling, and Visualization 2003*, pages 553–560, Munich, Germany, 2003.

[7] M. Isard and A. Blake. CONDENSATION — Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[8] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, pages 35–44, 1960.

[9] C.B. Madsen and H.I. Christensen. A Viewpoint Planning Strategy for Determining True Angles on Polyhedral Objects by Camera Alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):158–163, 1997.

[10] H. Murase and S. Nayar. Visual Learning and Recognition of 3–D Objects from Appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[11] P. Lehel and E.E. Hemayed and A.A. Farag. Sensor Planning for a Trinocular Active Vision System. In *Proceedings of Computer Vision and Pattern Recognition*, pages II: 306–312, Fort Collins, CO, 1999. IEEE Computer Society Press.

[12] S. D. Roy, S. Chaudhury, and S. Banerjee. Recognizing Large 3-D Objects through Next View Planning using an Uncalibrated Camera. In *International Conference on Computer Vision*, pages II: 276 – 281, Vancouver, Canada, 2001. IEEE Computer Society Press.

[13] B. Schiele and J.L. Crowley. Transinformation for Active Object Recognition. In *International Conference on Computer Vision*, pages 249–254, Bombay, India, 1998. IEEE Computer Society Press.

[14] R.S. Sutton and A.G. Barto. *Reinforcement Learning*. A Bradford Book, Cambridge, London, 1998.

[15] A. Törn and A. Žilinskas. *Global Optimization*, volume 350 of *Lecture Notes in Computer Science*. Springer, Heidelberg, 1987.