International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 23, No. 4 (2009) 765–799

World Scientific
www.worldscientific.com

A FRAMEWORK FOR ACTIVELY SELECTING VIEWPOINTS IN OBJECT RECOGNITION

FRANK DEINZER[*,†], CHRISTIAN DERICHS[*,‡]
and HEINRICH NIEMANN[§]

Chair for Pattern Recognition
Department of Computer Science
University of Erlangen-Nuremberg
Martensstr. 3, 91058 Erlangen, Germany
[†]*lme@deinzer.eu*
[‡]*derichs@informatik.uni-erlangen.de*
[§]*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

Object recognition problems in computer vision are often based on single image data processing. In various applications this processing can be extended to a complete sequence of images, usually received passively. In contrast, we propose a method for active object recognition, where a camera is selectively moved around a considered object. Doing so, we aim at reliable classification results with a clearly reduced amount of necessary views by optimizing the camera movement for the access of new viewpoints (viewpoint selection). Therefore, the optimization criterion is the gain of class discriminative information when observing the appropriate next image.

We show how to apply an unsupervised reinforcement learning algorithm to that problem. Specifically, we focus on the modeling of continuous states, continuous actions and supporting rewards for an optimized recognition. We also present an algorithm for the sequential fusion of gathered image information and we combine all these components into a single framework.

The experimental evaluations are split into results for synthetic and real objects with one- or two-dimensional camera actions, respectively. This allows the systematic evaluation of the theoretical correctness as well as the practical applicability of the proposed method. Our experiments showed that the proposed combined viewpoint selection and viewpoint fusion approach is able to significantly improve the recognition rates compared to passive object recognition with randomly chosen views.

Keywords: Active vision; viewpoint selection; reinforcement learning; sensor data fusion; optimization.

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

1. Introduction

The work introduced in this paper emerges from the fundamental problem of integrating autonomous systems into the human environment in a supporting manner. In particular, we concentrate on robots that are intended to recover arbitrary objects within an environment. Thus, their main task is to first of all detect and then classify relevant objects. If decisions on object classes are hard to make, more views from usually different positions are required.

Today's prevalent *active vision* approaches in computer vision that use an active observer go back to the work of Ref. 1, 4 and others. An overview of work related to active vision can be found in Ref. 14. In this context, active object recognition differs from passive object recognition in the sense that a sequence of views of the same object is not provided randomly. Instead, viewpoint selection tackles precisely the problem of finding a sequence of optimal views to improve classification and localization results by avoiding ambiguous views or by sequentially ruling out possible object hypotheses. Here, localization refers to the determination of the rotational pose of the object to the camera, instead of its position in the scene. Please note that almost any large or extensive series of non-identical image data of an object would lead the class decision process towards the correct result, independent of whether the viewpoints were chosen actively or not. So, optimality in our problem specification is defined by the demand to simultaneously reduce the amount of necessary object views and raise the recognition certainty. In general, this requirement is justified if taking images itself is very costly, like in various industrial and medical applications. For example, classifying a disease pattern based on X-ray images should only apply as few views as necessary for the welfare of the patient. Also, if performing a planned sensor action uses energy from a very limited source, like in a Mars exploring robot, actions should be well chosen. Please note, that because of this reason we do not explicitly address the computation time question for calculating next best views. These thoughts lead to the question of complexity for such tasks. Important work related to this problem was done by Tsotsos, for example, in Ref. 40 where it was pointed out that attention is a key element in solving the complexity problem. Since Ref. 45 proved that such sensor planning tasks are NP-complete in general, in our work the phrase of optimality itself needs to be considered as associated with the spent learning effort as well. So, for a growing experience our solution converges towards the globally optimal one.

1.1. *Proposed framework and methods therein*

In the given framework, the technical expression *viewpoint selection* might be confusing since we cannot explicitly declare a predefined pose of the object to be accessed next. This is due to the fact that the only information we get is the pure image pixel data when taking a picture. Thus, all we can build up is a single combined discrete-continuous **probability density**, representing our knowledge about the considered object's discrete class and its continuous pose *relative* to the camera.

This distribution, which becomes a **state** in our framework, contains no knowledge about world coordinates. Accordingly, we do not need to differentiate between moving the object and moving the camera, apart from the algebraic sign. Of course, in real world applications, like a robot task, we have to perform initial calibrations as well as a preceding or integrated object search in the scene. But this part is omitted in this presentation since it would distract from the original topic, which is object learning and subsequent recognition. However, a detailed explanation of active vision for object search can be found in Ref. 44. There, decisively different requirements on a camera movement, like the tracking of a spatial relationship between consecutive images, need to be considered. Additionally, those tasks have to deal with aspects like occlusion by other items[42] or in opposite the exploitation of those items for finding the deserved object by appearance correlation information.[43] Consciously not dealing with these problems we assume to have a perfectly tracked object right from the start, so we expect to know the rough object position within the image. So, in particular we do not claim to improve or expand the well-known idea of active vision itself, but the novelty in this work is the specific acquisition and exploitation of information about the given objects. Consequently, what we compute is a relative camera movement — typically on a circle or a hemisphere around the object — that should bring us to the desired viewpoint if our probabilistic assumption points out the correct state. Thus, the ongoing consideration of the object pose (rotation relative to the camera) is more of an essential support for active view planning and object recognition, respectively, rather than an overriding goal of the proposed methodology.

Though the main goal of our view planning framework (with its states of combined discrete and continuous densities) is to select the best camera action at each unit of time, there is another very important **fusion** issue that is also seamlessly handled in our framework. In particular, the afore-mentioned fusion is that of merging the classification and localization results of a sequence of viewpoints. This fusion process is the *state adaptation* part within our active view planning framework — so, a problem arises, if how to fuse the collected views to return an inline classification and localization estimation. In general, a random sequence of views will improve the recognition rate if a decent fusion scheme is applied. In our framework, we apply a fusion scheme which is based on the **Condensation Algorithm**,[17] i.e. we use a particle representation for the state densities. The applicability of such a fusion method to multimodal distributions over the class and pose space of the objects has already been shown in Ref. 8. We extend here that work to show that such a fusion scheme can handle the incoming image information as well as the camera action in an integrative way.

Active object recognition in our framework is based on a machine learning procedure, namely **Reinforcement Learning (RL)**.[38] Multiple sequences of randomly selected camera movements are performed in a training phase. Prior to this training process, we also build a model for each object class we want to recognize (see Sec. 3). Newly gathered image data is used to modify the state probability density after each camera action that is performed in this training phase. Consequently, we learn triples of values, containing (a) a state, (b) camera actions performed in this state and (c) a resulting reward for that action. A set of meaningful and unsupervised acquirable rewards will be introduced in Sec. 5.3.

Using the afore-mentioned, established components, like reinforcement learning, probability densities and Condensation Algorithm, we present a common framework that combines them into a single working system (see Sec. 2) for active object recognition. In developing this framework, we enhanced some of the components to provide abilities that are not inherent to them. These adaptions in conjunction with the proposed framework of density combination states makes our approach new and unique. More specifically,

- Our approach introduces a new method for the fusion of the generated views by applying a recursive density propagation algorithm. There, the fusion method is not limited to a special classifier, but is sufficiently general to work with almost all classifiers. This makes it applicable to a very wide range of tasks and supports the original intention of active view planning: "The importance, however, of this understanding is that one does not spend time on processing and artificially improving imperfect data but rather on accepting imperfect, noisy data as a matter of fact and incorporating it into the overall processing strategy".[3]
- The optimal sequence of views is learned automatically in a training step without any user interaction. Therefore, we present different continuous measures that can directly be calculated out of the state representations during that training. Usually, the common reinforcement learning just makes use of a discrete success/failure information after each action step.
- Due to limitations in memory, the traditional reinforcement learning is limited to a discrete set of reinforcement learning-states — the probability density states in our case — and discrete actions which can be performed by the camera. We will show how to utilize those discrete instances for finding an optimal action in a continuous action space when coming to the evaluation phase. We will, thus, show how to calculate Kullback–Leibler density distances on the underlying particle representation. Then we combine those state distances with the difference in learned and currently possible actions into a common approximation term for the next best action.

The resulting universal applicability will experimentally be shown in Sec. 6.

1.2. *Related work*

Active viewpoint selection has been discussed in the past regarding several different applications. For example, Refs. 28, 25 and 31 adopt the selection of a next best view to the task of 3-D object reconstruction. Those algorithms are designed for reconstructing arbitrary, previously unseen objects, thus omitting any kind of training phase. In return, viewpoint optimality is influenced by constraints like an overlap of object surfaces in consecutive views — a requirement that is dispensable in object recognition. In Ref. 23, active approaches are also utilized in optimal segmentation of image data. This method is also based on a specially defined objective function.

Active planning methods have also been used in object recognition itself. For example, Roy *et al.*[36] planned the next view for a movable camera based on probabilistic reasoning. The active part is the selection of a certain cut of an object that typically does not fit into the image. Unlike our approach, subsequently taken images with view planning need an overlap of object cuts — as is normal in object reconstruction tasks. Dickinson *et al.* determined distinctive object views before starting any object recognition.[13] The drawback of such methods is that viewpoint planning is completely independent of the current class probability distribution at a time step. The algorithm just causes the camera to go for the one, globally best position. Based on this problem, a more sophisticated planning in Ref. 19 provides the whole sequence of necessary views for reliable classification by applying a cluster analysis combined with a tree search. Unfortunately, this algorithm is strictly bounded to discrete sensor positions. It also cannot detect the potential need of policy changing within a view planning run. In another approach, the work of Ref. 20 uses Bayesian networks to decide on the next view to be taken. This method is limited to particular recognition algorithms and to certain types of objects, for which the Bayesian network has been manually constructed. In other words, the approach cannot be applied without user interaction.

Approaches that completely omit a training phase typically rely on information theoretic measures, like in Refs. 37 and 10. Here, the optimal action is directly searched for by maximizing the mutual information between the observation and the estimation on the object's class and pose. These techniques are typically vulnerable to approximation errors when estimating image features at previously unseen positions. In trying to gain more stable decisions on the next best view, works like Ref. 21 additionally actively select features that are most likely to support the information theoretic approach. Overall, the performance of these methods is weaker than that of learning-based ones. Nevertheless, they have an advantage when training is inapplicable or too expensive.

Other approaches are more closely related to the one proposed in this paper as they selectively move a camera for optimized object recognition. Arbel and Ferrie[2] addressed the view planning problem by establishing entropy maps in the training phase instead of using reinforcement learning. However, object ambiguities are introduced by the fact that the object recognition part is based solely on object shapes and unlike our method, it does not use the intensity images. Furthermore, in comparison to our work, the inaccuracy in the mobile robot's movement is not modeled at all in Ref. 2. In contrast, in the object recognition approach of Ref. 29, the training is also performed according to reinforcement learning. However, it omits the detailed declaration of a parametric function approximation for the objective function concerning the view planning optimization. Instead, this is extensively provided in this paper to permit an action search in continuous space.

1.3. *Contents*

In Sec. 2, we will give a basic introduction to the interaction of our system components, i.e. the object recognition, the data fusion, the viewpoint selection and the termination decision component. Section 3 shortly introduces the potential feature extraction methods, namely the well known Principal Component Analysis and a local Wavelet approach as well as a common statistical enhancement. Sections 4 and 5 describe the new contribution of our work to the task of active object classification. They present the knowledge representation and data fusion as well as the adapted reinforcement learning approach for viewpoint selection. The comprehensive experimental results in Sec. 6 show that the proposed approach is able to learn a theoretically optimal strategy for viewpoint selection which records only the minimal number of images. The improvement in classification results compared to randomly taken images is highlighted. The paper concludes with a summary and an outlook to future work in Sec. 7.

2. System Overview

2.1. *Components of the view planning system*

In our framework, active object recognition is composed of a series of distinct tasks. The first step to solve the given task of active object recognition is to decide which individual tasks within our framework are to be treated. As mentioned in Sec. 1, we tackle the problem by providing a reinforcement learning training first and then search for the best action in a reinforcement learning evaluation run. Figure 1 shows the underlying loop which has to be performed multiple times (so-called episodes) during training and once for each ensuing recognition. It contains four individual components:

- One component must provide the possibility to perform a basic probabilistic class and pose estimation of the presented object. So, it first calculates a feature

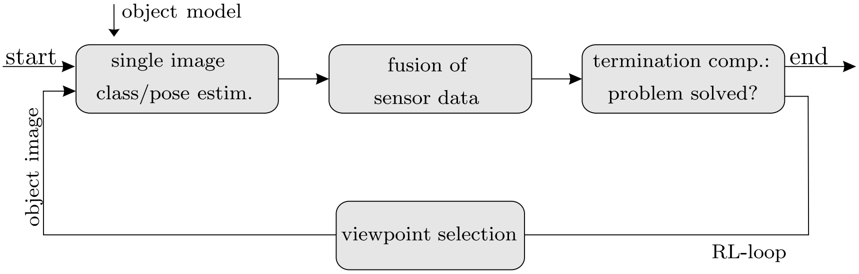

Fig. 1. Active object recognition divided into several interacting components.

vector out of the currently acquired image. This is one of the classical problems in image processing. Here, we decided on two alternative approaches, namely the Principal Component Analysis based on intensity values and a local Wavelet feature extraction method which will both be explained in Sec. 3. Note, that for the PCA, we already need a representative set of object images or feature vectors, respectively. These are taken from the object model that is built prior to the training stage (see Sec. 2.2).

We then subsequently match the image features with those features stored in the model in order to get probabilities of the object class and object pose — independently of earlier image information. Although our approach can be used together with almost any object recognition system, we describe the classifiers used throughout this paper in Sec. 3. This chapter will also present a statistical extension to the mentioned classifiers.

- As it is necessary in many applications to combine the results of several images from different viewpoints taken at different time steps, the integration of new information into the current knowledge about the observed object must be possible. So, the second system component deals with the fusion of sensor data into a common probabilistic description of the current object recognition state. The details of this description and its adaption over time are given in Sec. 4. This chapter will present a new general framework for fusion in object recognition, based on the well-established methods of recursive density propagation and the Condensation Algorithm.
- The termination component determines whether a given goal is reached. In general, this termination component depends heavily on the given problem definition, but in our case, it naturally alludes to the *class certainty*, which will be introduced in Sec. 5.3. Of course, in both reinforcement learning training and evaluation, we additionally have a fixed maximum amount of fusion steps within an episode as a termination criterion.
- If the problem cannot be solved with the existing information, the viewpoint selection component must decide from which viewpoint a new image should be captured. During the evaluation stage, this decision is based on the rewards gathered during the reinforcement learning training phase. Logically, during the training itself, this component degenerates to a random action selection. The essential parts — the adequate mapping of the current situation to the training experience and the reward definition itself — are explained in Sec. 5. The key contribution of this part is the use of continuous states and action spaces. It is important to us that no part of the viewpoint selection module requires the world to be discretized. In our opinion, today's state of the art in that field of research should not depend on such assumptions.

The interaction of the components is shown in Fig. 1. Our system, as described above, is a "flexible construction kit" that allows for an adaption to special requirements. This flexibility is achieved by the possibility to change the realizations of each of the blocks with components that may be better suited for a specific task. For example, the system can be used in any environment, as it is possible to use any statistical classifier for the "class/pose estimation" component. Another adjustable element is the reward in the "viewpoint selection" module. It can be adapted to different goals, like minimum number of views or maximum certainty of the classification result.

2.2. *The underlying object model*

As mentioned, before starting the view planning process itself, we need to have a rough idea of the objects that can possibly appear. Therefore, we initially collect L images of **each** object class, ideally equidistantly distributed over all possible viewpoints. More specifically, we have $L = L_h$ images on a horizontal circular path around the object for a 1-D representation and $L = L_h \cdot L_v$ images in case of a hemisphere covering 2-D representation. Subsequently, the model building itself consists of two steps. First, for each of those images, a **feature vector** is calculated according to one of the methods that will be presented in Sec. 3. Secondly, these feature vectors are stored together according to where they were taken from, i.e. the position on the circular path or the hemisphere, respectively.

3. Single Image Class/Pose Estimation

There exists a large number of object recognition methods that are built on some sort of feature-based representation. Many of these methods differ by the features used in describing the objects.

3.1. *Eigenspace approach*

A very popular feature based method is the so-called *eigenspace approach* based on Ref. 27, which uses a *Karhunen-Loeve Transformation*[22] (also known as *Principal Components Analysis* (PCA)) to obtain a linear system for computing a feature vector by

$$\mathbf{c} = \mathbf{\Phi}(\mathbf{f} - \boldsymbol{\mu}_{\mathbf{f}}), \quad \mathbf{\Phi} \in \mathbb{R}^{N_c \times N_f}. \tag{1}$$

The vector $\mathbf{f}$ contains all intensities of the N_f pixels of the object image. The rows of matrix $\mathbf{\Phi}$ contain the $N_c \leq N_f$ eigenvectors with the largest eigenvalues of the covariance matrix of the training images. In particular, for our classification purpose, we use far less than N_f eigenvectors. Regarding the experiments, the maximal number of applied eigenvectors will be shown to be $N_c = 20$. The average of all training images is denoted as $\boldsymbol{\mu}_{\mathbf{f}}$. Without loss of generality, we assume that the average image is subtracted from the object image in advance.

In Murase's traditional eigenspace approach,[27] a representative set of i images per class κ is used to calculate reliable PCA feature vectors. In our case, we just take the L images $\mathbf{f}^{i,\kappa}$ selected according to Sec. 2.2 to calculate a single eigenspace matrix. Thus, the features $\mathbf{c}^{i,\kappa}$ can be calculated by (1). Since the PCA provides the most discriminative feature vectors for the given set of input images, it is well suited to the classification task. Classification and localization of any other image can finally be done by finding the class and pose in the model representation that has the most similar feature vector $\mathbf{c}^{i,\kappa}$: this is the feature vector which minimizes an Euclidian distance measure

$$d(\mathbf{c}, \mathbf{c}^{i,\kappa}) = \|\mathbf{c} - \mathbf{c}^{i,\kappa}\|_2. \tag{2}$$

3.2. *Local wavelet approach*

In general, the proposed eigenspace approach applies a global measure for comparing images. So, it works quite well with optimally preprocessed images, i.e. object transformation within the image should be minimal and background modification should hardly appear. To overcome this disadvantage — especially regarding our experiments with objects on cluttered background (see Sec. 6.2) — we alternatively consider local wavelet features for recognition. In that case, any image is divided into $W \in \{w_1, \dots, w_{\max}\}$ subimages of 8×8 pixels each, referred to by $\mathbf{f}_{w_n}$. Then, using a Johnston 8-TAP wavelet,[32] a wavelet multiresolution analysis[24] with depth $\delta = \log_2 8 = 3$ is performed on each subimage. Consequently, the low-pass coefficient $l_{n,\delta}$ as well as the three direction dependent high-pass coefficients $h_{n0,\delta}, h_{n1,\delta}$ and $h_{n2,\delta}$ can be calculated for the given depth δ and subimage $\mathbf{f}_{w_n}$. Combining the high-pass parts to a common value, we yield several two-dimensional feature vectors

$$\mathbf{c}_n = \begin{pmatrix} \ln|l_{n,\delta}| \\ \ln(|h_{n0,\delta}| + |h_{n1,\delta}| + |h_{n2,\delta}|) \end{pmatrix} \tag{3}$$

regarding the current subimage index n. So, for image comparison, we separately need to calculate a feature vector distance for each local $\mathbf{f}_{w_n}$ and multiplicatively combine them to a global measure

$$d(\mathbf{c}, \mathbf{c}^{i,\kappa}) = \left(\prod_{n=1}^{W} d(\mathbf{c}_n, \mathbf{c}_n^{i,\kappa}) \right)^{\frac{1}{W}} \tag{4}$$

with $d(\mathbf{c}_n, \mathbf{c}_n^{i,\kappa})$ according to (2).

3.3. *Statistical enhancement*

In our fusion approach that will be presented in Sec. 4, a classifier is needed that gives a statistical measure. For that reason, we will shortly present two ways to extend (2) and (4) to a statistical value.

3.3.1. *Statistical enhancement using exponential distributed distances*

An easy and common way to get a statistical measure from a distance is to assume exponentially distributed distances. This means that the classification leads to a statistical version

$$p(\mathbf{c}|\mathbf{c}^{i,\kappa}) = \frac{1}{\mu}\exp\left(-\frac{d\left(\mathbf{c},\mathbf{c}^{i,\kappa}\right)}{\mu}\right), \tag{5}$$

where $\mu > 0$ is a parameter of the exponential distribution. The classification and localization is finally done by finding the class and pose in the training database that maximizes (5), the Gibbs probability.

3.3.2. *Statistical enhancement using normally distributed features*

Another statistical approach was presented for the first time in Ref. 15 where $\mathbf{c}^{i,\kappa}$ is replaced by a normal distribution

$$p(\mathbf{c}|\boldsymbol{B}^{i,\kappa}) = \mathcal{N}(\mathbf{c}|\mu^{i,\kappa}, \boldsymbol{\Sigma}^{i,\kappa}), \tag{6}$$

where $\mu^{i,\kappa}$ denotes the mean vector and $\boldsymbol{\Sigma}^{i,\kappa}$ the covariance matrix of the resulting feature vectors when adding noise several times to the same instance of an image $\mathbf{f}^{i,\kappa}$. These two components build a *statistical model* $\boldsymbol{B}^{i,\kappa} = (\boldsymbol{\mu}^{i,\kappa}, \boldsymbol{\Sigma}^{i,\kappa})$ which is estimated by adding noise and applying small transformations (e.g. translation) to the training image $\mathbf{f}^{i,\kappa}$ n-times. The result of the noise- and transformation-adding processes are the new training images $\mathbf{f}^{i_1,\kappa}, \mathbf{f}^{i_2,\kappa}, \ldots, \mathbf{f}^{i_n,\kappa}$ and the corresponding feature vectors $\mathbf{c}^{i_1,\kappa}, \mathbf{c}^{i_2,\kappa}, \ldots, \mathbf{c}^{i_n,\kappa}$ which are used to estimate the mean vector $\boldsymbol{\mu}^{i,\kappa}$ and the covariance matrix $\boldsymbol{\Sigma}^{i,\kappa}$. The drawback of this approach is that it is only able to give localization results for object poses that are in the set of the training images, indicating that only *discrete* poses are found. A solution to this problem is also given in Ref. 15 by providing the following continuous parameterization of the normal distribution

$$\boldsymbol{B}(\kappa,\phi) = (\boldsymbol{\mu}(\kappa,\phi), \boldsymbol{\Sigma}(\kappa,\phi)) \tag{7}$$

where κ describes the discrete class number and ϕ denotes the continuous pose parameter. Since it is assumed that a discrete number of normal distributions are available, classification and pose estimation are done by searching for the discrete class and continuous pose that maximize (7). The result usually will specify an object pose that was not in the training database.

4. Fusion of Sensor Data

The fusion expression is used in mainly two different meanings in computer vision literature. On the one hand, it describes how to deal with different sensor modalities that are sensing mainly the same section of a scene at a time. To get an overall impression of a possibly observed object, various methods of intelligent sensor data combination are then known, reaching from average calculation to selective voting.[35] Instead, fusion in this work refers to the sequential integration of different images acquired by a single modality, i.e. camera intensity images. So, this section discusses how this problem can be formalized in a statistical framework. First, we would like to introduce the basic notation used in this paper. In active object recognition, a series of observed images

$$\langle \mathbf{f} \rangle_t = \mathbf{f}_t, \mathbf{f}_{t-1}, \ldots, \mathbf{f}_0 \tag{8}$$

of an object at different times t is given together with the sequence of camera movements

$$\langle \mathbf{a} \rangle_{t-1} = \mathbf{a}_{t-1}, \ldots, \mathbf{a}_0 \tag{9}$$

between these images. Based on these observations of images and movements, one wants to draw conclusions for a non-observable **state sample $\mathbf{q}_t$** of the object. This state sample $\mathbf{q}_t$ must contain both the *discrete* class Ω_κ and the *continuous* appearance parameters

$$\phi = (\phi_1, \ldots, \phi_\psi)^T \tag{10}$$

of the object, like pose, scaling, internal rotation and so on. This leads to the state sample definition

$$\mathbf{q}_t = (\Omega_\kappa, \phi_1^t, \ldots, \phi_\psi^t)^T. \tag{11}$$

In our framework, a state is a collection of probability values that can be calculated at diverse state samples $\mathbf{q}_t$ (see also (18)).

Please note that the declaration of the pose parameters of the object has to be associated and updated with the actual, current camera position. Thus, ϕ is always assumed to be time variant. The actions $\mathbf{a}_t$ consist of the relative camera action with ψ degrees of freedom, $\mathbf{a}_t = (\Delta\phi_1^t, \ldots, \Delta\phi_\psi^t)$ with $\Delta\phi^t = \phi^{t+1} - \phi^t$. In order to provide a class and pose estimation for every captured image by the underlying model, camera actions are restricted to those which leave the camera on the mentioned circular path or hemisphere, respectively.

In this paper, experimental results will be based upon two appearance parameters at most, namely the vertical and horizontal pose (ϕ_1, ϕ_2) of the object relative to the camera. These pose parameters will be measured in angle values and determine the camera position on a hemisphere relative to the object. Since we use the class and 2-D pose information in the object modeling as well, $\mathbf{q}_t$ can be directly mapped to the underlying model.

4.1. *Recursive density propagation*

A common technique for state estimation which became very popular in the last years is based on the principle of recursive density propagation which we proposed in Ref. 6. A main contribution of our work is the adaption of this well-known approach to the problem of classification and localization of objects with more than one observed image (as introduced in Sec. 2). In contrast to previous work, we do not only have a state and an observation (the observed image), but also camera actions. Additionally, the proposed algorithm is general enough to work with any classifier besides those introduced in Secs. 3.1 and 3.2, as long as it is able to provide a probability in the form of (5) or (6), respectively. If we model this problem as a recursive density propagation, the knowledge on the object state is given in form of the *a posteriori* density

$$p(\mathbf{q}_t | \langle \mathbf{f} \rangle_t , \langle \mathbf{a} \rangle_{t-1}). \tag{12}$$

This requires having all actions $\langle \mathbf{a} \rangle_{t-1}$ and observed images $\langle \mathbf{f} \rangle_t$ available. For practical applications, this is not suitable. 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 (12) specialized for our problem with additional camera actions:

$$p(\mathbf{q}_t | \langle \mathbf{f} \rangle_t , \langle \mathbf{a} \rangle_{t-1}) = p(\mathbf{f}_t | \mathbf{q}_t) \int p(\mathbf{q}_t | \mathbf{q}_{t-1}, \mathbf{a}_{t-1}) \cdot p(\mathbf{q}_{t-1} | \langle \mathbf{f} \rangle_{t-1} , \langle \mathbf{a} \rangle_{t-2}) d\mathbf{q}_{t-1}. \tag{13}$$

The detailed algorithm including our task specific constraints and the detailed derivation of (13) can be found in Ref. 9.

This formulation fulfills our requirements: the integration of a new image $\mathbf{f}_t$ and a new action $\mathbf{a}_{t-1}$ into the current knowledge about the object given by the density $p(\mathbf{q}_{t-1} | \langle \mathbf{f} \rangle_{t-1} , \langle \mathbf{a} \rangle_{t-2})$. A simplified example for one step of such a density propagation considering the cups in Fig. 4 is shown in Fig. 2. This figure shows the components for one fusion step:

(1) *a posteriori* distribution for $\mathbf{q}_{t-1}$
(2) state transition with movement inaccuracy modeling
(3) *a priori* distribution for $\mathbf{q}_t$
(4) integration of knowledge $p(\mathbf{f}_t | \mathbf{q}_t)$ leading to *a posteriori* distribution for $\mathbf{q}_t$.

One can calculate the feature vector of $\mathbf{f}_t$ using (1) or (3) and approximate the feature vector for each $\mathbf{q}_t$ via the underlying model. Then $p(\mathbf{f}_t | \mathbf{q}_t)$ can be computed via (5) or (6).

The recursion of (13) bottoms out at the initial state probability distribution $p(\mathbf{q}_0)$. This distribution contains the initial knowledge about the object and its pose. If no *a priori* knowledge is available, $p(\mathbf{q}_0)$ is assumed to be uniformly distributed over the state space. The only assumption we make within this framework is the permanence of the considered object during the classification process. So, to classify an object with class $q_1 = \Omega_\kappa$ in a sequence of T images, we postulate

$$q_{1,t} = q_{1,t-1} = \cdots = q_{1,t-T+1}. \tag{14}$$

If the problem is reduced to discrete pose parameters in $\mathbf{q}_t$, the integral in (13) could easily be evaluated in an analytical way. But we are interested in the fusion

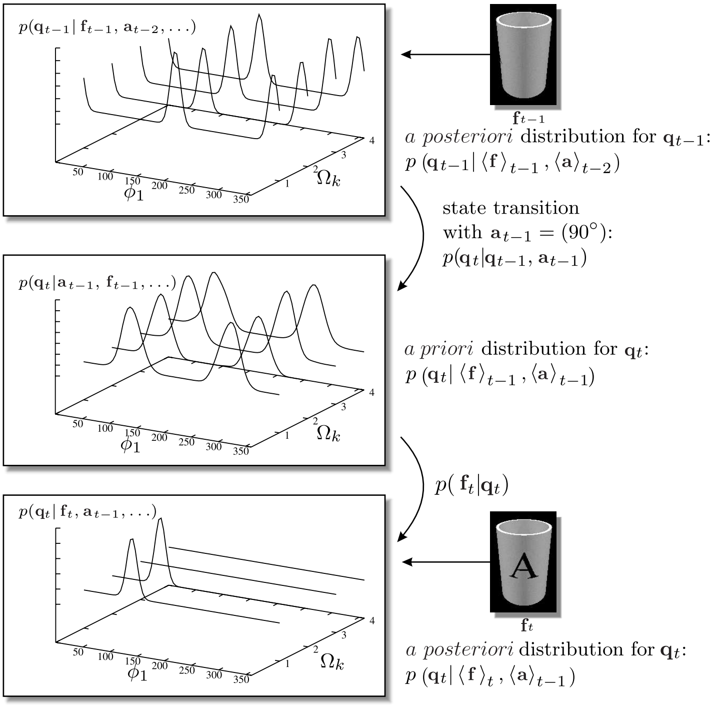
$p(\mathbf{q}_{t-1} | \mathbf{f}_{t-1}, \mathbf{a}_{t-2}, \ldots)$
50 100 150 200 250 300 350
ϕ_1
1 2 3 4
Ω_k
$\mathbf{f}_{t-1}$
a posteriori distribution for $\mathbf{q}_{t-1}$:
$p\left(\mathbf{q}_{t-1} | \langle \mathbf{f} \rangle_{t-1}, \langle \mathbf{a} \rangle_{t-2}\right)$
state transition
with $\mathbf{a}_{t-1} = (90°)$:
$p(\mathbf{q}_t | \mathbf{q}_{t-1}, \mathbf{a}_{t-1})$
$p(\mathbf{q}_t | \mathbf{a}_{t-1}, \mathbf{f}_{t-1}, \ldots)$
a priori distribution for $\mathbf{q}_t$:
$p\left(\mathbf{q}_t | \langle \mathbf{f} \rangle_{t-1}, \langle \mathbf{a} \rangle_{t-1}\right)$
$p(\mathbf{f}_t | \mathbf{q}_t)$
$p(\mathbf{q}_t | \mathbf{f}_t, \mathbf{a}_{t-1}, \ldots)$
A
$\mathbf{f}_t$
a posteriori distribution for $\mathbf{q}_t$:
$p\left(\mathbf{q}_t | \langle \mathbf{f} \rangle_t, \langle \mathbf{a} \rangle_{t-1}\right)$

Fig. 2. Illustration of the single tasks necessary for the density propagation for one new view. Each plot line shows the probability of a continuous pose ϕ_1 given a discrete class Ω_κ.

of multiple views in a general way with the possibility of continuous pose parameters. This means that we have to develop extended methodologies for handling the continuous densities in a nonparametric way.

4.2. *Fusion using particle filters*

The classic approach for solving a recursive density propagation problem as given above is the Kalman Filter.[18] But in computer vision, the necessary assumption for the Kalman Filter, $p(\mathbf{f}_t|\mathbf{q}_t)$ being normally distributed, is often not valid due to object ambiguities, sensor noise, occlusion, etc.

One approach for the complicated handling of such nonanalytical and additionally combined discrete and continuous densities are the so-called particle filters.[17] The basic idea is to approximate *a posteriori* density by a set of weighted samples. In our approach, we use the Condensation Algorithm[17] which uses a sample set $Y_t = \{y_t^1, \ldots, y_t^M\}$ at time step t to approximate the multimodal probability distribution (12) by M samples $y_t^i = \{\mathbf{x}_t^i, p_t^i\}$. Each sample y consists of the point $\mathbf{x} = (\Omega_\kappa, \phi_1, \ldots, \phi_\psi)$ within the state space and the weight p for that sample with the condition that $\sum_i p_t^i = 1$. Please note the identical declarations of $\mathbf{q}$ and $\mathbf{x}$. The only difference is that $\mathbf{x}$ is a state hypothesis that is actually represented by a particle, whereas $\mathbf{q}$ should be seen as an arbitrary position within the state space.

The sample set only represents the density (12). An algorithm that allows for the use of sample sets in recursive density propagation problems is the Condensation Algorithm.[17] It starts with an initial sample set Y_0 representing $p(\mathbf{q}_0)$. In our application, we distribute the samples uniformly over the state space as we will have no prior knowledge about the objects before observing them for 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 weighting;

(2) propagated with a necessarily predetermined sample transition model according to $p(\mathbf{q}_t|\mathbf{q}_{t-1}, \mathbf{a}_{t-1})$ in (13). In this work, we assume the sample transition to be

$$\mathbf{x}_t^i = \mathbf{x}_{t-1}^i + (0, u_1^i, \ldots, u_\psi^i)^T \quad \text{with } u_j^i \sim \mathcal{N}(\Delta\phi_j^t, \sigma_j). \tag{15}$$

Equation (15) models the inaccuracy of the camera movement under the assumption that the former is independent of the movement components. The variance parameters σ_j of the Gaussian transition noise have to be defined in advance (see Sec. 6). Since the error in transition is supposed to be unbiased and symmetric, modeling noise with a Gaussian distribution is reasonable. It is important to note that the uncertainty modeled by the sample transition is external, i.e. the inaccuracy of the camera movement, and not internal, i.e. the state estimation itself, as it is usually the case for particle filter applications.

(3) evaluated in the image by $p(\mathbf{f}_t|\mathbf{x}_t^i)$. This evaluation is performed by the classifier. The only requirement for the classifier that shall be used together with our fusion approach is its probabilistic expandability in order to evaluate this density. In this work, we use a classifier based on the continuous statistical eigenspace approach as presented in Ref. 15. In related previous work,[16,33] we have shown that other classifiers, like the Wavelet classifier, can also be used in the presented fusion approach.

Given a collection of Y_t samples, a classification is possible at each time step via marginalization over all possible poses for each class. This can be done in our setup by a simple summation

$$p(\Omega_\kappa) = \int_\phi p((\Omega_\kappa, \phi_1, \ldots, \phi_\psi)^{\mathrm{T}}|\mathbf{f}_t, \mathbf{a}_{t-1}, \ldots)\, d\phi = \sum_{\left(i|(x_t^i)_1 = \Omega_\kappa\right)} p_t^i. \tag{16}$$

At this point, we want to note that it is important to include the class Ω_κ in the object state $\mathbf{q}_t$ and the samples y_t^i. An alternative would be to omit this by setting up several sample sets — one for each object class — and perform the Condensation Algorithm separately on each set. But this would not result in an integrated classification/localization.

4.3. *Evaluation of particle filter densities*

In the context of the viewpoint selection (see Sec. 5), the densities which are represented by sample sets have to be evaluable at any continuous position. The direct evaluation of them beyond the positions given by the individual samples is not possible. So it is necessary to find a continuous representation of them. A common way to evaluate non-parametric densities is the *Parzen estimation*[30] which is calculated for the sample set Y by

$$p(\mathbf{q}_t | \langle \mathbf{f} \rangle_t , \langle \mathbf{a} \rangle_{t-1}) \approx \widetilde{p}_Y \left(\mathbf{q}_t | \langle \mathbf{f} \rangle_t , \langle \mathbf{a} \rangle_{t-1} \right) = \frac{1}{M_Y} \sum_{i=1}^{M_Y} g_0(\mathbf{q}_t - \mathbf{x}_t^i), \tag{17}$$

with $g_0(\mathbf{v}) = \mathcal{N}(\mathbf{v}|\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma})$ being a zero-mean normal distribution which denotes the window function. The choice of the mean vector $\boldsymbol{\mu} = \mathbf{0}$ follows from the fact that the difference $(\mathbf{q}_t - \mathbf{x}_t^i)$ in (17) results in zero-mean data. In contrast, the definition of the covariance matrix requires a careful consideration of methods like the mean minimal distance of samples[5] or the entropy-based approach of Ref. 41. Since the latter includes a complex, time consuming optimization method, we applied the approach of in Ref. 5. For a more detailed explanation on the theoretical background of the approximation of (13) by a sample set, we refer to Ref. 17.

In Ref. 6, we discussed the use of probability density trees to evaluate the density. But results have shown that the quality of the approximation generated by the probability density trees is not sufficient for our purpose. The major advantage of density trees would be that they show a significantly lower memory consumption.

5. Viewpoint Selection

A straight forward and intuitive way to formalizing the problem of viewpoint selection is given in Fig. 3 which shows the basic loop performed in reinforcement learning. A continuous alternation between sensing s_t and action $\mathbf{a}_t$ can be seen. The chosen *action* $\mathbf{a}_t$ corresponds to the executed camera movement as described in Sec. 4, the sensed state (introduced in Sec. 1)

$$s_t = p(\mathbf{q}_t | \langle \mathbf{f} \rangle_t , \langle \mathbf{a} \rangle_{t-1}) \tag{18}$$

is the density as given in (12). Additionally, the classification module returns a so-called *reward* r_t, which measures the quality of the chosen action with respect to

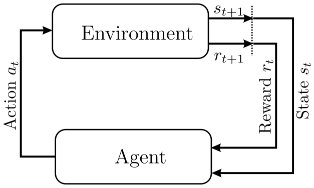

Fig. 3. Interaction between the environment and the reinforcement learning agent.

the resulting viewpoint. The definition of the reward is an important aspect as this reward shall model the goal that has to be reached. So, proper definitions for the reward in the context of our viewpoint selection problem are another crucial contribution of this work. Particular emphasis was given to measures that can be directly extracted from an available state density s_t, thus satisfying the requirements of unsupervised learning. Details on the rewards are given in Sec. 5.3.

5.1. *Reinforcement learning*

The goal at time t during the evaluation phase of a reinforcement learning process is to maximize the accumulated and weighted future rewards, called the *return*

$$R_t = \sum_{n=0}^{\infty} \gamma^n r_{t+n+1} \quad \text{with } \gamma \in [0;1]. \tag{19}$$

The weight γ defines how much influence a future reward will have on the overall return R_t at time $t+n+1$. 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*

$$Q\left(s, \mathbf{a}\right) = E\left\{R_t | s_t = s, \mathbf{a}_t = \mathbf{a}\right\} \tag{20}$$

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

Viewpoint selection can now be defined as a two-step approach: First, estimate the function $Q\left(s, \mathbf{a}\right)$ during training. Second, if at any time the viewpoint fusion returns s as classification result, select that camera movement which maximizes the expected accumulated and weighted rewards. This function is called the *policy*

$$\pi(s) = \underset{\mathbf{a}}{\operatorname{argmax}}\, Q\left(s, \mathbf{a}\right). \tag{21}$$

The key issue of course is the estimation of the function $Q\left(s, \mathbf{a}\right)$, which is the basis for the decision process in (21). One of the demands defined in Sec. 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. Details about the learning methods can be found in Ref. 38 for the general case and in Ref. 8 for our specific viewpoint selection problem.

5.2. *Continuous state and action spaces*

Most of the algorithms in reinforcement learning treat the states and actions as discrete variables. Of course, in our viewpoint, selection framework 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 directly applied to our viewpoint selection framework. It is necessary to find a way to allow for the usage of the required continuous states and actions. So, next to the adapted fusion process and the preparation of adequate reward definitions, this is a central aspect of our work.

We propose the extension of the algorithms presented in Ref. 38 to continuous reinforcement learning by approximating the action-value function as follows:

$$\widehat{Q}\left(s,\mathbf{a}\right) = \frac{\sum_{(s_t^k,\mathbf{a}_t^k)} K(d(\theta(s,\mathbf{a}),\theta(s_t^k,\mathbf{a}_t^k)))\cdot Q\left(s_t^k,\mathbf{a}_t^k\right)}{\sum_{(s_t^k,\mathbf{a}_t^k)} K(d(\theta(s,\mathbf{a}),\theta(s_t^k,\mathbf{a}_t^k)))}, \tag{22}$$

where $\theta(s,\mathbf{a})$ is a transformation function (see next subsection). The other components within (22) are the *distance function* $d(\cdot,\cdot)$ and a kernel function $K(\cdot)$. Equation (22) can be evaluated for any continuous state/action pair $(s,\mathbf{a})$. Basically, this is a weighted sum of the action-values $Q\left(s_t^k,\mathbf{a}_t^k\right)$ of all the state/action pairs $(s_t^k,\mathbf{a}_t^k)$ which were collected during all previous training episodes k.

Finally, the viewpoint selection problem of finding the optimal action $\mathbf{a}^*$, i.e. the computation of the policy π, can now be written, according to (21), as an optimization problem

$$\pi(s) = \mathbf{a}^* = \underset{\mathbf{a}}{\operatorname{argmax}}\,\widehat{Q}\left(s,\mathbf{a}\right). \tag{23}$$

It is solved in this work by applying a global Adaptive Random Search Algorithm followed by a local Simplex.[39]

5.2.1. *Transformation function*

The *transformation function* $\theta(s,\mathbf{a})$ transforms a state s with a known action $\mathbf{a}$ to a new state 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 state from (18) it can be seen as a density transformation

$$\theta(s_t,\mathbf{a}_t) = \theta(p(\mathbf{q}_t|\left\langle\mathbf{f}\right\rangle_t,\left\langle\mathbf{a}\right\rangle_{t-1}),\mathbf{a}_t) = \det(\boldsymbol{J}_{\zeta_{\mathbf{a}_t}}(\mathbf{q}_t))p(\zeta_{\mathbf{a}_t}(\mathbf{q}_t)|\left\langle\mathbf{f}\right\rangle_t,\left\langle\mathbf{a}\right\rangle_{t-1})) \tag{24}$$

with

$$\zeta_{\mathbf{a}}(\mathbf{q}) = (q_1, q_2 - a_1, \ldots, q_{\psi+1} - a_\psi)^T \tag{25}$$

and the Jacobian matrix

$$\boldsymbol{J}_{\zeta_{\mathbf{a}}}(\mathbf{q}) = \begin{pmatrix} \frac{\partial(\zeta_{\mathbf{a}})_1}{\partial q_1} & \cdots & \frac{\partial(\zeta_{\mathbf{a}})_{\psi+1}}{\partial q_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial(\zeta_{\mathbf{a}})_1}{\partial q_{\psi+1}} & \cdots & \frac{\partial(\zeta_{\mathbf{a}})_{\psi+1}}{\partial q_{\psi+1}} \end{pmatrix} = \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix}. \tag{26}$$

The density transformation simply performs a shift of the density.

5.2.2. *Distance function*

The distance function is used to calculate the distance between two states. Generally speaking, similar states must result in low distances. The lower the distance, the more transferable is the information from a learned action-value to the current situation. As the transformation function (24) results in a density, the *Kullback–Leibler Distance*

$$\begin{aligned} d_{\mathrm{KL}}(s_t, s'_{t'}) &= d_{\mathrm{KL}}\big(p(\mathbf{q}_t|\left\langle \mathbf{f}\right\rangle_t, \left\langle \mathbf{a}\right\rangle_{t-1}), p(\mathbf{q}_{t'}|\left\langle \mathbf{f}'\right\rangle_{t'}, \left\langle \mathbf{a}'\right\rangle_{t'-1})\big) \\ &= \mathrm{E}\left\{\log \frac{p(\mathbf{q}_t|\left\langle \mathbf{f}\right\rangle_t, \left\langle \mathbf{a}\right\rangle_{t-1})}{p(\mathbf{q}_{t'}|\left\langle \mathbf{f}'\right\rangle_{t'}, \left\langle \mathbf{a}'\right\rangle_{t'-1})}\right\} \qquad (27) \\ &= \int p(\mathbf{q}_t|\left\langle \mathbf{f}\right\rangle_t, \left\langle \mathbf{a}\right\rangle_{t-1}) \log \frac{p(\mathbf{q}_t|\left\langle \mathbf{f}\right\rangle_t, \left\langle \mathbf{a}\right\rangle_{t-1})}{p(\mathbf{q}_{t'}|\left\langle \mathbf{f}'\right\rangle_{t'}, \left\langle \mathbf{a}'\right\rangle_{t'-1})} d\mathbf{q}\,, \qquad (28) \end{aligned}$$

is an appropriate distance metric. In our method, we use its symmetric distance measure extension, the so-called *extended Kullback–Leibler Distance*

$$d_{\mathrm{EKL}}(s_t, s'_{t'}) = d_{\mathrm{KL}}(s_t, s'_{t'}) + d_{\mathrm{KL}}(s'_{t'}, s_t). \qquad (29)$$

Please note that, in general, there is no analytic solution for (29). However, since we represent our densities as sample sets anyway (see Sec. 4) there are well-known ways to approximate (29) by Monte Carlo techniques using the Parzen estimation (17). Calculating the Kullback–Leiber Distance using (28) is numerically too costly, so it is preferable to approximate the expectation in (27). As Ref. 41 showed, this can be done by drawing $M_{\widehat{Y}}$ samples from (17), setting up a new sample set $\widehat{y}^1, \widehat{y}^2, \ldots, \widehat{y}^{M_{\widehat{Y}}}$ with particles $\widehat{y}^i = \{\widehat{\mathbf{x}}^i, \widehat{p}^i\}$ analog to the definition of particle sets in Sec. 4.2. For $M_{\widehat{Y}} \rightarrow \infty$ the expectation (27) in combination with the Parzen estimation (17) converges to the real Kullback–Leibler-Distance:

$$\mathrm{E}\left\{\log \frac{p(\mathbf{q}_t|\left\langle \mathbf{f}\right\rangle_t, \left\langle \mathbf{a}\right\rangle_{t-1})}{p(\mathbf{q}_{t'}|\left\langle \mathbf{f}'\right\rangle_{t'}, \left\langle \mathbf{a}'\right\rangle_{t'-1})}\right\} = \frac{1}{M_{\widehat{Y}}} \sum_{i=1}^{M_{\widehat{Y}}} \log \frac{\widetilde{p}_{\widehat{Y}}\left(\widehat{\mathbf{x}}^i|\mathbf{f}_t, \mathbf{a}_{t-1}, \mathbf{f}_{t-1}, \ldots\right)}{\widetilde{p}_{\widehat{Y}}\left(\widehat{\mathbf{x}}^i|\mathbf{f}'_{t'}, \mathbf{a}'_{t'-1}, \mathbf{f}'_{t'-1}, \ldots\right)}. \qquad (30)$$

Again, we need to calculate (30) twice with a switch of densities to get the symmetric measure (29).

5.2.3. *Kernel function*

The kernel function $K(\cdot)$ weighs the calculated distances. A suitable kernel function is the Gaussian

$$K(x) = \exp\left(-\frac{x^2}{D^2}\right) \qquad (31)$$

because it transfers the distance values into similarity values and at the same time, creates less influence on those densities with a low similarity, regarding (22). In (31), 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 $Q\left(s', \mathbf{a}'\right)$ 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. If one wants to be conservative with this parameter, a potential larger value for D is favorable.

5.3. *Reward definition*

As mentioned above, the proper definition of the reward r_t is a key point in our viewpoint selection and reinforcement learning. In our context, four different definitions of rewards are reasonable.

5.3.1. *Inter-class distance*

The simplest reward definition which we have looked into over time[7,34] is the inter-class distance between the two best class hypotheses. In other words, we define a viewpoint to be useful if the difference of the quality measure between the best and second best object hypotheses is large. This definition is a reasonable reward which has shown very good results for simple planning problems.[7,34] This inter-class distance does not require a statistical upgrading of the feature distance to the classifier (see Sec. 3.3). In fact, it is possible to use this measure in combination with any arbitrary classifier as long as the latter delivers some kind of sensible distances representing object similarities.[7]

5.3.2. *Fixed end-value*

Another way to rate camera movements is to define a reward that has a value of 0 except when reaching the terminal state:

$$r_t = \begin{cases} C > 0 & : s_t \text{ is terminal state} \\ 0 & : \text{otherwise} \end{cases} \tag{32}$$

The advantage of (32) is that it 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. The user has to decide beforehand when the confidence of the classification is high enough. As a result, an episode with a merely sufficient information gain will get the same reward as an episode collecting even more class discriminative information. So this strategy does not necessarily maximally increase the class certainty at each time step.

5.3.3. *Entropy*

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

$$r_t = -H(s_t) = -H(p(\mathbf{q}_t| \langle \mathbf{f} \rangle_t , \langle \mathbf{a} \rangle_{t-1})). \tag{33}$$

In that sense, the reward expresses the gain of knowledge about the object. Please note that since we work in an unsupervised manner regarding class and pose estimation, a relatively definite but incorrect state representation would also gain a high entropy based reward. This fact will be of interest in Sec. 6.2. Equation (33) has the advantage that the goal is to improve the classification besides only trying to reach a stop criterion. There are two approaches for the calculation of the entropy in (33):

$$H(p(\mathbf{q}_t|\,\langle\mathbf{f}\rangle_t\,,\langle\mathbf{a}\rangle_{t-1})) = \mathrm{E}\left\{-\log(p(\mathbf{q}_t|\,\langle\mathbf{f}\rangle_t\,,\langle\mathbf{a}\rangle_{t-1}))\right\} \tag{34}$$

$$= -\int_{-\infty}^{+\infty} p(\mathbf{q}_t|\,\langle\mathbf{f}\rangle_t\,,\langle\mathbf{a}\rangle_{t-1})\log p(\mathbf{q}_t|\,\langle\mathbf{f}\rangle_t\,,\langle\mathbf{a}\rangle_{t-1})d\mathbf{q}_t \tag{35}$$

As the real density (12) is not available, one has to use the Monte Carlo summation with Parzen estimation (17) once more. Regarding the definition of the sample set, one obtains:

$$\mathrm{E}\left\{-\log\left(p(\mathbf{q}_t|\,\langle\mathbf{f}\rangle_t\,,\langle\mathbf{a}\rangle_{t-1})\right)\right\} = -\frac{1}{M_{\widehat{Y}}}\sum_{i=1}^{M_{\widehat{Y}}}\log\widehat{p}^i. \tag{36}$$

For $M_{\widehat{Y}} \to \infty$ the expectation (34) converges to the real entropy.

5.3.4. *Class certainty*

A reward measure that is very much aligned to our proposed class probability function (16) is the class certainty. Thereby a resulting density distribution is assessed to be of high significance if the summarized probability over all object poses of the most probable class is high as well:

$$r_t = \max_i \int_\phi p(\mathbf{q}_t^i|\,\langle\mathbf{f}\rangle_t\,,\langle\mathbf{a}\rangle_{t-1})\;d\mathbf{q}^i \quad \text{with} \quad \mathbf{q}^i = (\Omega_{\kappa=i},\phi_1,\ldots,\phi_\psi)^T\,. \tag{37}$$

Consequently, given k classes, rewards in our approach obey the relation $k^{-1} \leq r_t \leq 1$ since this is the range of possible probabilities for the most probable class at each time step. So this reward measure relies on the absolute summarized probability of the most probable class rather than on relative probabilities like in the case of the inter-class distance measure. Please note, that in a final, non-probabilistic classification decision, we decide for the absolutely best class instead of evaluating any class distances.

In contrast to the entropy measure, the compromise to be accepted with the class certainty criteria is its disregard of any accuracy concerning the assumption of the object pose.

5.3.5. *Cost of actions*

It is worth noting that the reward might also include costs for the camera movement, so that large movements of the camera are punished. In this paper, we neglect costs for camera movement for the time being. But work in this area has been done, for example, in Ref. 11.

6. Experimental Results

The goal of our proposed framework of active object recognition is reliable object classification with a reduced amount of necessary view. This means that either we should be able to reduce the number of sensor movements needed to achieve a reliable classification rate to the optimal minimum, or we should obtain improved classification results after the same number of views are compared to an unplanned proceeding. For evaluating the presented approach, we arranged the following three different kinds of experimental setups for showing the theoretical correctness as well as the practical applicability of our approach.

6.1. *Simply structured objects with 1-D planning*

6.1.1. *Theoretical considerations*

For our 1-D setup, the task is to differentiate between the four classes of synthetically generated cups shown in Fig. 4. Those are marked by an A or a B on the one side and a 1 or a 2 on the opposite side. Please note, that these objects were designed to exclude a reliable classification by just one view as well as to provide several positions of ambiguity. In general, all letters and numbers are at least partially visible within a horizontal range of 150° around their centers at 90° and 270°, respectively. For this basic setup, we just consider a 1-D camera movement on a circular path around the object. Furthermore, assuming arbitrary starting positions, an averaged theoretical minimum of $2.1\overline{6}$ necessary views for a reliable classification can be calculated:

- Assuming the mentioned 150° of visibility of the imprints, the chance for getting some class information with the first random view is $83.\overline{3}\%$.

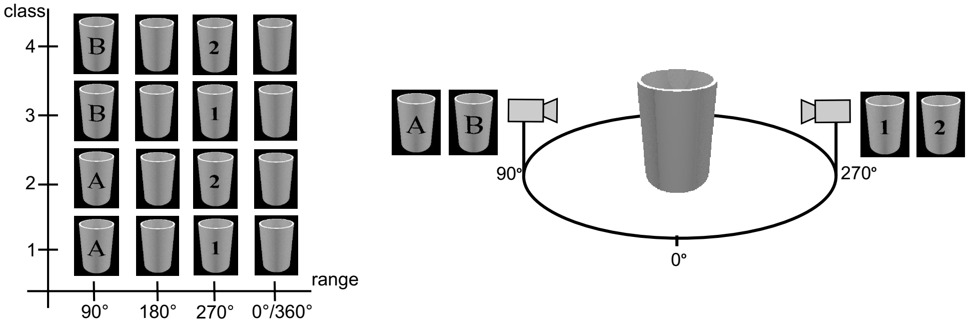

Fig. 4. Views of the set of synthetic object classes.

- Optimal planning needs two information containing views at most for reliable classification, thus three views are the theoretical minimum if the first one is redundant, two otherwise.

The minimum of $2.1\bar{6}$ views implies the assumption that the visibility of an imprint is equal to its recognizability. Given small areas where this assumption fails, the theoretical minimum has to be slightly raised.

6.1.2. *Practical evaluation*

For comparing the performance of our approach to this minimum, we first built an underlying model. It contains the statistically enhanced $N_c = 10$ eigenfeatures for each of 360 equidistantly distributed images of every object class. The corresponding feature extraction and statistical adjustment are computed as shown in Secs. 3.1 and (6), respectively.

Based on this model, $N_R = 10$ reinforcement learning training episodes composed of $T_{\max} = 8$ time steps each were performed for every class in the database. Within each of those steps, a randomized virtual camera movement is carried out, resulting in a next view and a subsequent feature extraction. For compensating a possible movement inaccuracy, σ_1 in (15) was chosen to be 1°, since this value experimentally appeared to be adequate for a complete parameter range of 360°. After fusing the gathered information as described in Sec. 4, we use the entropy based reward (33) for rating the prior movement and iteratively building up the knowledge base during training. For representing the underlying densities, we provided $M = 1440$ particles altogether. This way, we initially cover each pose hypothesis of all four classes in equidistant steps of 1°. This amount of particles was empirically found to be large enough for a reliable state representation and small enough for a fast calculation of the Condensation Algorithm (see Sec. 4.2).

Additionally, we performed three independent training phases for evaluating the influence of three variations of the weighting γ in (19), representing the extremes of a single step approach ($\gamma = 0$) and the one independent of the sequential appearance of rewards within an episode ($\gamma = 1$). $\gamma = 0.5$ is supposed to represent all settings between those two extremes.

Our evaluation phase consisted of 250 episodes for each class. The goal was to proceed greedily. So given an arbitrary starting position, the policy (23) had to be computed in every single step in order to reduce the amount of necessary views for reliable classification. Experiments were repeated for some variations of $D \epsilon \{2, 5, 10, 20, 50\}$ in (31) since this is a crucial parameter when deciding on the similarity of states. Finally, Table 1 shows the results with the last column representing the averaged number of views μ_{VP} that had to be taken before reaching the preset stopping criterion of 90% class certainty, according to (37). Please note that the extremely high classification results gained after t fused steps are not unexpected since we have a fairly simple classification problem here, if not confronted

Table 1. Classification results [%] for the 1-D synthetic dataset (see Fig. 4) after t planned views based on $N_R = 10$ training episodes per class. The first eight columns show the results depending on a pair of parameters (γ, D) whereas the last column displays the averaged number of necessary views μ_{VP} for reaching the stopping criterion.

	$N_R = 10$								
	$t = 1$	2	3	4	5	6	7	8	μ_{VP}
Unplanned	48.5	59.7	71.6	81.4	91.2	94.0	96.1	96.5	3.53
$\gamma = 0$									
2	46.6	92.4	99.5	99.9	99.9	99.7	99.9	99.9	2.22
5	49.8	92.5	99.3	99.6	99.7	99.7	99.7	99.6	2.25
$D = 10$	48.1	94.4	99.7	99.8	100	99.8	99.9	99.9	2.20
20	48.0	94.1	99.7	99.7	99.7	99.7	99.7	99.7	2.20
50	47.3	93.1	99.9	100	100	100	100	100	2.22
$\gamma = 0.5$									
2	45.3	92.1	99.7	99.9	99.9	99.9	99.9	99.9	2.23
5	47.8	93.7	99.8	99.8	99.8	99.8	99.8	99.8	2.23
$D = 10$	46.6	93.3	99.4	99.7	99.8	99.9	99.9	99.9	2.23
20	47.9	92.3	99.6	99.6	99.7	99.9	99.9	99.9	2.25
50	45.7	94.2	99.5	99.6	99.8	99.9	99.9	99.9	2.22
$\gamma = 1$									
2	47.8	92.9	99.4	99.6	99.8	99.9	99.9	99.9	2.23
5	47.8	91.6	99.6	99.6	99.7	99.8	99.9	99.9	2.27
$D = 10$	45.2	92.9	99.7	99.9	99.9	99.9	99.9	99.9	2.26
20	44.6	93.4	99.5	99.8	99.9	99.9	99.8	99.8	2.26
50	46.7	93.0	99.5	99.7	99.6	99.7	99.7	99.7	2.25

with an ambiguity. The values for $t = 1$ are the single image classification results without any information fusion. They differ among the rows of Table 1 since we permitted random starting views in each episode of the reinforcement learning evaluation phase.

Considering the entries for reasonable values of $\gamma = 0$ and $\gamma = 0.5$, we concluded that we never need more than 1.04 times the number of views of the theoretical minimum. Regarding a single episode, the worst appearing episode with this cooperative objects showed a number of $\mu_{VP\,\max} = 4$ necessary views. In comparison, in the unplanned proceeding, we got 22 out of the 1000 episodes that were not able to provide a reliable classification even after $\mu_{VP\,\max} = 8$ views. An asset of our framework is that for achieving those rates, we only had to perform $N_R = 10$ episodes for each object class during training. Figure 5 shows in greater detail the development of μ_{VP} when even boosting the amount of training episodes. It contains three graphs representing the various values of γ. Results of μ_{VP} are averaged over all discrete $D \,\epsilon\, \{2, 5, 10, 20, 50\}$ and plotted against the training complexity N_R.

As expected, more extensive training provides higher potential of reducing necessary views. But given the theoretical minimum as a value of convergence, at some point, the additional training effort is not profitable any more.

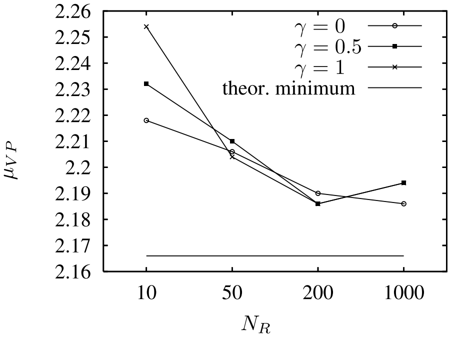

Fig. 5. Development of the number of necessary views μ_{VP} for a reliable classification depending on the amount of underlying training episodes N_R. The bottom line represents the theoretical minimum.

To provide a further insight on the influence of the kernel width D to the gained results, Table 2 shows the entries similar to Table 1 for extreme values of D and $\gamma = 0$. As can be seen, we obtain partially better results than for the unplanned proceeding even if D has extremely small or big values, although performance is not optimal in those cases. But since the decline of classification performance regarding D is quite flat, our approach is supposed to work optimally for a wide range of kernel widths with this dataset.

6.2. *Task specific objects with 2-D planning*

Since the above results indicate that the suggested approach works as expected, we now consider real, but still task oriented objects and extend the possible viewpoints to be located on a hemisphere around them (see Fig. 7). Object classes are represented by the four variants of toy manikins shown in Fig. 6, either carrying a quiver, a lamp, both, or none of these equipments. We selected those items because they are hard to classify from quite a wide range of views.

For offline computation we chose steps of 1.125 degrees in vertical as well as in horizontal direction to gain a fundamental image set of 81 × 320 = 25920 entries per class. Taking every other image and calculating its features [see Eq. (1)], we

Table 2. Classification results [%] for the 1-D synthetic dataset (see Fig. 4) adequate to Table 1, but with extreme values for D.

		$N_R = 10$								
		$t = 1$	2	3	4	5	6	7	8	μ_{VP}
$\gamma = 0$										
$D =$	0.02	47.5	56.2	61.4	66.1	68.1	72.1	74.7	75.7	6.82
	0.05	41.8	76.3	83.2	87.6	89.4	92.1	93.1	93.8	3.83
	500	46.1	82.2	95.0	95.6	96.4	96.6	96.6	96.8	3.70
	1000	45.0	79.9	93.2	93.2	93.6	93.2	93.3	93.4	3.76

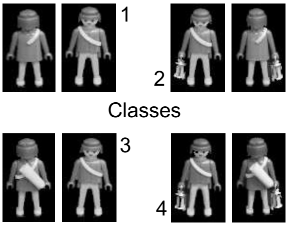

Fig. 6. Views of the non-synthetic object classes.

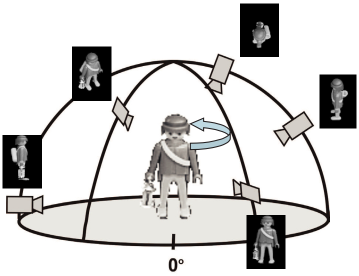

Fig. 7. Description of possible camera positions on a hemisphere around the object.

can construct the underlying model according to Sec. 3.1. This time, we made use of Gibbs formula as a statistical adjustment (5) for calculating the class and pose probabilities.

Given the classifier model, we now take the other half of all taken images for the purpose of the reinforcement learning based training phase as well as for the ensuing evaluation phase. This way, we avoid getting wrongly conditioned results by working on images already appearing in the model representation. Unfortunately, this time we can hardly provide a theoretical minimum number of views essential for classification at this problem. So, here, success is best judged by a comparison of classification results between planned and random camera movement. Since our method is placing less emphasis on high classification results in general, but rather focuses on proper view planning, we added a Gaussian noise to all images recorded during training and evaluation. This is required since the underlying classifier itself might be far too good to leave some space for remarkable enhancements in classification results by the view planning. Please note that our approach is nevertheless not ill-posed because of this fact since it works with any classifier able to evaluate a density, as mentioned in Sec. 4.2.

During the reinforcement learning training phase, for each class in the database, we now provide 25 episodes of randomly chosen sensor actions and resulting images to the algorithm, due to the higher dimensionality in space. For particle transition (see (15)), we again set $\sigma_1 = \sigma_2 = 1°$. Each episode contains eight steps of image retrieval and consecutive information fusion at most, with the termination criterion of at least 90% classification certainty. Further planning is supposed to be superfluous and the episode is terminated accordingly. Please note that with the added noise, we rarely reached a certainty of 90% based on the acquired eight images. Consequently, a comparison of μ_{VP} for planned and unplanned episodes is not meaningful with this dataset and will be omitted.

Thus, up to seven random actions per episode are rated according to the probability density distribution calculated in each step. Handling the 2-D space, the density representation depends on $M = 6720$ particles altogether, that is 1680 particles per class.

In order to show the robustness of the introduced algorithm again, several tests where performed varying some of the essential parameters. We first chose two different values for the number of eigenvectors $N_c \in \{5, 20\}$, making up the feature space. Five eigenvectors provided quite reliable classification results, whereas using more than twenty does not lead to noticeable enhancements anymore. Additionally, we again tested the three variations of the weighting $\gamma \in \{0, 0.5, 1\}$ and the various common kernel parameters $D \in \{2, 5, 10, 20, 50\}$. Table 3 lists the corresponding classification results in each step, compared to those generated by unplanned sensor action. For the evaluation of the unplanned sequences, we considered 250 episodes per class in order to provide solid reference values. Results of the planned sequences were then computed relying on another 250 episodes for each parameter combination and object class.

Offensive environment: Please note that for determining the reinforcement learning reward in these experiments, we now used the class certainty (37). This appears to be suboptimal in a first instance since it noticeably differs from the entropy based reward (33) which is a well-known measurement for information content in a state. But our decision was forced by the fact that the chosen dataset is vulnerable to a heavy misclassification when approaching a particular, quite small range of viewpoints. Of course, misclassification is something we have to deal with, but in that case from those positions a wrong class is slightly more probable than in any other sensor position. So the entropy measure would prefer those harmful positions in usually all following timesteps. As we work in an unsupervised manner concerning the object class assumption, classification rates would extremely drop within an episode. We showed the entropy reward to work well with other datasets,[12] but in general, we cannot assume to have an inoffensive environment. So, for being able to raise the classification rate iteratively, a change to (37) was fundamental as well as meaningful, since it must conform to the classification measurement established in (16). For comparison purposes, the classification results using 33 instead of 37 are shown in Table 4.

Taking a look at the results, our choice of reward as well as the complete view planning approach is justified since we almost universally get higher classification rates compared to performing arbitrary sensor movements. Especially early steps within an episode ($t = 2, 3$) partially gain a benefit of more than 10% in class

Table 3. Classification results [%] for the 2-D dataset after t planned views compared to an unplanned proceeding. Evaluation was done using $N_c = 5, 20$ eigenvectors. Bold numerical values highlight the ten highest gains in classification rate within the particular table.

$N_c = 5$	$t = 1$	2	3	4	5	6	7	8
Unplanned	44.3	64.8	78.6	85.3	89.2	90.4	91.8	93.1
$\gamma = 0$								
2	43.9	**76.9**	87.6	92.2	94.7	95.7	96.2	96.6
5	46.4	**78.3**	**90.4**	93.2	95.3	96.7	97.3	97.7
$D = 10$	42.9	**77.3**	**88.8**	91.8	92.8	93.9	94.7	95.2
20	46.1	**76.2**	**89.1**	92.8	93.8	95.6	95.9	95.4
50	43.4	72.0	84.6	89.7	91.7	93.4	94.8	95.1
$\gamma = 0.5$								
2	42.8	73.6	87.8	94.0	95.1	95.9	96.5	97.0
5	40.6	**74.7**	**88.1**	89.9	92.1	93.3	93.7	94.6
$D = 10$	43.3	**76.9**	85.3	87.4	86.9	87.0	88.2	88.3
20	46.8	71.2	85.0	87.5	89.3	91.1	92.1	92.6
50	42.3	67.2	78.7	82.7	84.3	87.3	88.6	90.5
$\gamma = 1$								
2	41.9	68.7	81.3	87.9	91.4	94.4	95.7	96.5
5	44.2	67.9	80.0	85.1	87.9	89.8	91.6	92.3
$D = 10$	45.1	70.2	80.4	83.8	86.6	88.5	90.4	91.9
20	46.5	66.4	78.9	84.8	88.4	91.7	92.1	92.6
50	43.1	64.3	71.5	78.1	80.8	83.5	86.3	88.0
$N_c = 20$	$t = 1$	2	3	4	5	6	7	8
Unplanned	44.7	66.2	76.3	82.7	87.8	89.3	90.5	91.7
$\gamma = 0$								
2	45.3	74.1	**87.3**	92.1	93.8	95.2	95.9	95.6
5	43.1	**78.5**	**89.3**	**94.0**	95.1	95.9	95.6	95.8
$D = 10$	43.5	76.1	**87.0**	89.8	90.2	90.7	91.4	91.5
20	44.8	**77.3**	**90.3**	92.6	94.1	94.1	94.7	94.7
50	42.0	69.9	85.4	89.5	91.5	93.2	94.2	95.0
$\gamma = 0.5$								
2	47.9	73.7	**87.7**	92.1	94.7	94.7	95.9	97.0
5	42.5	76.7	86.9	90.9	92.5	93.6	93.4	94.0
$D = 10$	44.7	**77.4**	**88.5**	90.9	92.4	94.0	94.1	94.2
20	41.6	73.6	86.9	90.5	91.9	93.4	94.0	94.0
50	41.9	69.7	82.5	86.6	89.2	88.9	90.6	91.5
$\gamma = 1$								
2	43.3	67.9	80.1	86.3	90.6	93.1	95.3	96.0
5	43.8	68.2	79.0	85.0	88.2	90.7	92.1	93.8
$D = 10$	44.5	68.3	79.8	86.0	87.7	88.8	90.7	92.2
20	42.9	64.9	74.4	81.3	85.6	87.7	89.7	91.7
50	41.8	66.5	77.2	82.8	86.9	88.9	89.8	89.8

Table 4. Classification results [%] for the non-synthetic dataset after t planned views. The reinforcement learning reward is based on the entropy of the state probability density (33).

$N_c = 5$	$t = 1$	2	3	4	5	6	7	8
Unplanned	44.3	64.8	78.6	85.3	89.2	90.4	91.8	93.1
$\gamma = 0$								
$D =$ 10	43.4	78.0	85.3	86.5	80.7	74.7	71.2	69.8
20	42.5	76.8	86.1	85.4	79.4	74.9	71.5	70.6

certainty. Regarding the results printed in bold—symbolizing the highest gains in classification—it is worth noting that none of them are in the fields of $\gamma = 1$. Thus, the time sequence of rewards within an episode emerges to be an extremely important factor when building the return (19). As one would expect, there is no value in rating a current action with the averaged sum of the immediate and any following reward. This should only be considered if episodes are quite long and training actions are performed randomly. Furthermore, it is observable that high values for the kernel parameter D tend to result in lower classification results. Especially for $D = 50$, we do not get any noticeable enhancement compared to a random proceeding. The reason is that such wide kernels do not support the formation of a very structured action-value function. Possibly learned optimal actions are devaluated by their rewards' superposition of many other rewards of quite dissimilar state-action combinations. On the other hand, extremely low kernel parameters, like $D = 2$, hardly suffer any loss in classification rate. The optimization algorithm is able to avoid getting stuck in a local minimum of the more detailed approximation of (22). Additionally, we can postulate that using a higher dimensional feature space transformation with $N_c = 20$ instead of five eigenvectors does not systematically lead to an improvement in classification results.

Unfortunately, sometimes the benefit of planning disappears when looking at the later steps of an episode. A clear example of this phenomenon is represented by the line in Table 3, that shows the results for $N_c = 5$, $\gamma = 0.5$ and $D = 10$. Here, for $t > 4$ the classification results for the random proceeding outperform the planned ones.

6.3. *Real world objects with heterogeneous background*

So far, we have concentrated on objects that show real ambiguities, where even high quality images of certain object views cannot provide enough information for reliable object class decision. We claimed that in those environments, the idea of active recognition is meaningful. On the other hand, the decision on ambiguity is less a binary one than a continuous one. Consequently, we can always provide some degree of ambiguity, e.g. by adding noise to the image or by assuming a changing background — even if all objects would be distinguishable by a single view on perfect images.

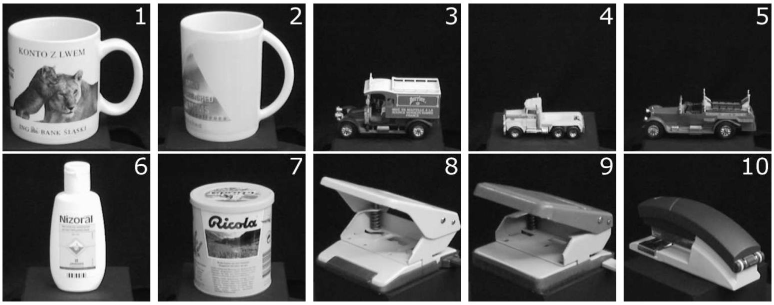

Fig. 8. Exemplary views of the ten objects in the real world database.

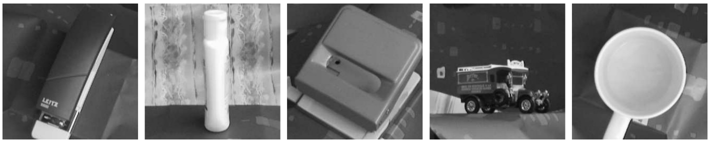

Fig. 9. Exemplary views of objects conjoint with a less complex heterogeneous background.

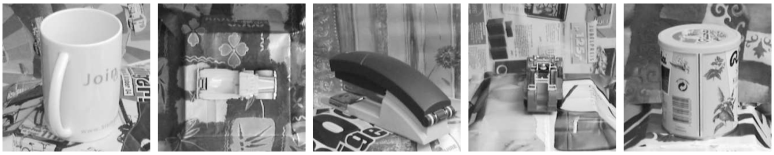

Fig. 10. Exemplary views of objects conjoint with a more complex heterogeneous background.

Thus, for showing the global functionality of our planning algorithm, we tested it on a bigger number of real objects, namely ten, which are likely to appear within the same real world environment. Figure 8 shows these objects in a uniform background, while Figs. 9 and 10 show them on some of the more complex cluttered backgrounds that we also used in our tests.

The images with the homogeneous background were used in building our object model. During that process, we once more take equidistantly distributed images from a hemisphere, this time 21 views in vertical and 80 views in horizontal directions for each object. The reduction of the sampling rate compared to the toy manikins provides a similarly extensive model (16800 entries) since we now have more object classes.

In contrast to the previous experiments, feature acquisition cannot be accomplished by applying the eigenspace approach any more, since it is unable to cope with changing backgrounds. Rather, we make use of the proposed local Wavelet features (see Sec. 3.2) with the statistical Gibbs enhancement applied to images of 256 × 256 pixels. In this setup, we had to augment the classification difficulty by permitting slight object translation within the image. This translation is the displacement of the object in the test image to those in the images making up the underlying model. In doing so, we allow background pixels to be wrongly classified as a foreground member, thus influencing the class decision. In order to handle this additional complexity, we displace every image under consideration to 25 discrete positions. Namely we move it with any combination of $\{-16; -8; 0; 8; 16\}$ pixels vertically and horizontally for finding the maximal fit with a model base image. Then, for each class probability, we just apply the highest occurring probability over all transitions. Of course, real translations are not limited to those discrete values, thus we are most likely to not obtain a perfect fit.

Learning from the previous experiments that early steps provide the highest gain in classification, we reduced the number of steps to $T_{\max} = 6$ for each of $N_R = 50$ reinforcement learning training episodes. As in the synthetic dataset, we again provided the entropy as a reward measure for executed sensor actions. Since we want the view planning to operate on arbitrary backgrounds, training is performed by just utilizing the shaded images (like those in Fig. 8). Also being aware of the robustness of the algorithm concerning D and γ, in this case, we just evaluated $D = 10$, $\gamma = 0$ and once more we set $\sigma_1 = \sigma_2 = 1°$. State representation was based on $M = 1200$ particles and experimental results are based on another 250 episodes performed on every object during evaluation. Figure 11 shows the

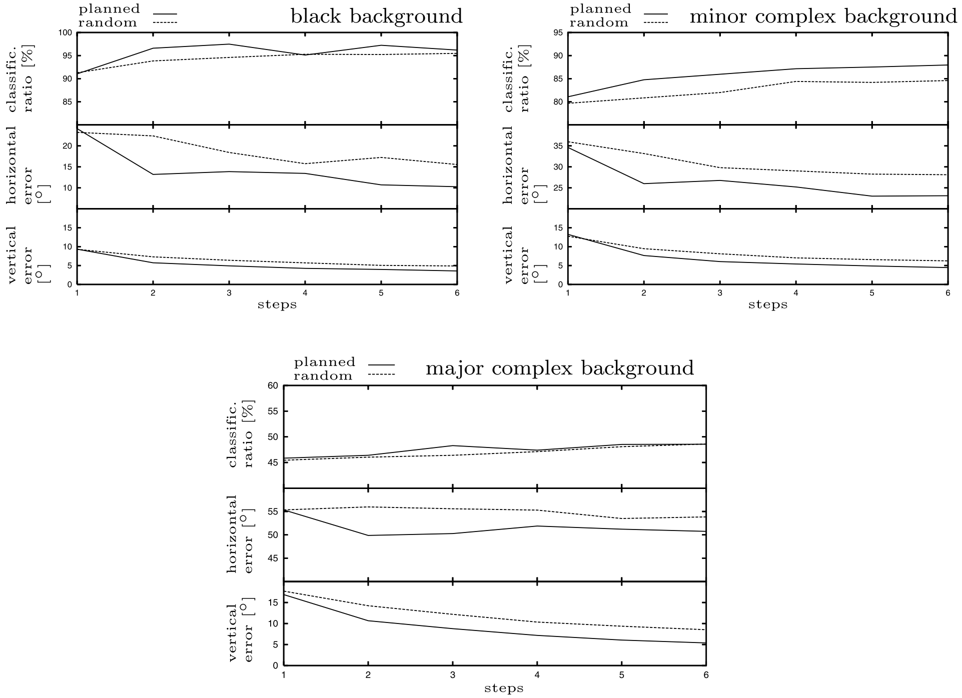

Fig. 11. Results for classification ratio, horizontal error and vertical error differentiated by the kind of image background. All values are averaged over all ten object classes.

classification ratio averaged over all ten classes, itemizing the kind of test image background, the step information and the type of action selection, which is either planned or random. The lower graphs in each plot show the development of the horizontal and vertical pose estimation errors.

For all three background cases, the planned motion provides better values for any arbitrary episode step. Regarding the classification ratio, the application of our view planning algorithm shows the highest benefit when working on images with mildly complex backgrounds. This case best fits the workings of the underlying classifier, where single image classification is between being trivial (like with homogeneous backgrounds) and being unreliable (as in highly complex backgrounds).

Note that the critical impact of our presented approach is closely tied to the information gathered during the reinforcement learning training phase. If in this stage, only highly unprofitable actions were performed, then almost no gain in classification would be achieved by active view planning. Furthermore, during evaluation, the probability of acquiring a state density similar to those occurred in the training phase declines with the step number t. Thus, planning of episodes' later steps is much more error-prone. Nonetheless, if approximation (22) is inoffensive, i.e. if D is selected adequately, planning can still be advantageous.

7. Summary and Future Work

In this paper, we proposed a general framework for viewpoint selection and viewpoint fusion and we demonstrated its application on both synthetic and real world classification problems. We motivated that the optimization criterion therein is the amount of necessary views needed for a reliable class decision, rather than considerations on computation time. 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. Our system can be automatically trained without user interaction. During the actual object recognition task, it continuously provides probabilistic information about the current object class and pose. The experiments show classification rates that outperform those achieved by using random or regularly sampled views. Furthermore, we discussed in detail the impact of different parameter values to the classification success of our proposed framework.

As mentioned, in our previous work[11] we already studied the integration of action costs into the view planning process. But this was solely done with the assumption of one-dimensional, unrestricted camera movements. Future work will consider the cost sensitive action selection when having a higher dimensional action space. In particular, the retaining preparation of consistent rewards — as explained in Sec. 5.3 — forms a challenging task in this extension. Another task will be to simultaneously build an object model and use it for classification, what is called *Online-Learning.* The challenge here is to find an optimized solution for the theoretically permanently competing demand on the camera action regarding these two goals since the best next view for building a significant model is not necessarily the best one for classification. Especially incorporating the afore-mentioned ideas, it is also an interesting and challenging task to extend the classification problem to object categories, i.e. to have multiple similar objects within a single class. Mattern *et al.*[26] can already provide an approach for this idea.

References

1. J. Aloimonos, I. Weiss and A. Bandyopadhyay, Active vision, *Int. J. Comput. Vis.* **2**(3) (1988) 333–356.
2. T. Arbel and F. Ferrie, Entropy-based gaze planning, *Imag. Vis. Comput.* **19**(11) (2001) 779–786.
3. R. Bajcsy, Active perception, *Proc. IEEE* **76**(8) (1988) 996–1005.
4. Dana H. Ballard, Animate vision, *Artif. Intell.* **48**(1) (1991) 57–86.
5. D. Cremers, *Statistical Shape Knowledge in Variational Image Segmentation*, PhD thesis, Department of Mathematics and Computer Science, University of Mannheim, Germany (2002).
6. F. Deinzer, J. Denzler, Ch. Derichs and H. Niemann, Aspects of optimal viewpoint selection and viewpoint fusion, *Computer Vision — ACCV 2006*, eds. P. J. Narayanan, S. K. Nayar and H. Shum, Lecture Notes in Computer Science, Vol. 3852 (Springer, Hyderabad, India, 2006), pp. 902–912.
7. F. Deinzer, J. Denzler and H. Niemann, Classifier independent viewpoint selection for 3-D object recognition, *Mustererkennung 2000, 22. DAGM-Symp.*, eds. G. Sommer, N. Krüger and Ch. Perwass (Springer, Kiel, Germany, 2000), pp. 237–244.
8. F. Deinzer, J. Denzler and H. Niemann, Viewpoint selection — planning optimal sequences of views for object recognition, *Computer Analysis of Images and Patterns*, (Springer, Groningen, Netherlands, 2003), pp. 65–73.
9. Frank Deinzer, *Optimale Ansichtenauswahl in der aktiven Objekterkennung* (Logos Verlag Berlin, 2005).
10. J. Denzler and C. M. Brown, Information theoretic sensor data selection for active object recognition and state estimation, *IEEE Trans. Patt. Anal. Mach. Intell.* **24**(2) (2002) 145–157.
11. Ch. Derichs, F. Deinzer and H. Niemann, Cost integration in multi-step viewpoint selection for object recognition, *Proc. Int. Conf. Machine Learning and Data Mining MLDM 2005*, (Leipzig, Germany, 2005), pp. 415–425.
12. B. Deutsch, F. Deinzer, M. Zobel and J. Denzler, Active sensing strategies for robotic platforms, with an application in vision-based gripping, INSTICC, *Proc. 1st Int. Conf. Informatics in Control, Automation and Robotics*, eds. H. Arajo, A. Vieira, J. Braz, B. Encarnao and M. Carvalho, Vol. 2 (INSTICC Press, Setbal, Portugal, 2004), pp. 169–176.
13. S. J. Dickinson, I. Christensen, K. Tsotsos and G. Olofsson, Active object recognition integrating attention and viewpoint control, *Comput. Vis. Imag. Underst.* **67**(3) (1997) 239–260.
14. B. Girod, G. Greiner and H. Niemann (eds.), Active Vision, *Principles of 3D Image Analysis and Synthesis* (Kluwer Academic Publishers, 2000).
15. Ch. Gräßl, F. Deinzer, F. Mattern and H. Niemann, Improving statistical object recognition approaches by a parameterization of normal distributions, *Patt. Recogn. Imag. Anal. Advances in Mathematical Theory and Applications* **14**(2) (2004) 222–230.
16. M. Grzegorzek, F. Deinzer, M. Reinhold, J. Denzler and H. Niemann, How fusion of multiple views can improve object recognition in real-world environments, *Vision,*

Modeling and Visualization 2003, eds. T. Ertl, B. Girod, G. Greiner, H. Niemann, H.-P. Seidl, E. Steinbach and R. Westermann, (AKA/IOS Press, Munich, Germany, 2003), pp. 553–560.

17. M. Isard and A. Blake, CONDENSATION — conditional density propagation for visual tracking, *Int. J. Comput. Vis.* **29**(1) (1998) 5–28.
18. R. E. Kalman, A new approach to linear filtering and prediction problems, *J. Basic Engin. D* **82** (1960) 35–44.
19. S. Kovačič, A. Leonardis and F. Pernuš, Planning sequences of views for 3-D object recognition and pose determination, *Patt. Recogn.* **31**(10) (1998) 1407–1417.
20. B. Krebs, M. Burkhardt and B. Korn, Handling uncertainty in 3D object recognition using Bayesian networks, *Computer Vision — ECCV'98*, Freiburg, Germany (1998), pp. 782–795.
21. C. Laporte, R. Brooks and T. Arbel, A fast discriminant approach to active object recognition and pose estimation, *Proc. 17th Int. Conf. Pattern Recognition*, Cambridge (2004), pp. 91–94.
22. M. Loeve, *Probability Theory*, 4th edn. (Springer, New York, 1978).
23. C. B. Madsen and H. I. Christensen, A viewpoint planning strategy for determining true angles on polyhedral objects by camera alignment, *IEEE Trans. Patt. Anal. Mach. Intell.* **19**(2) (1997) 158–163.
24. S. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, *IEEE Trans. Patt. Anal. Mach. Intell.* **11**(7) (1989) 674–693.
25. E. Marchand and F. Chaumette, Active vision for complete scene reconstruction and exploration, *IEEE Trans. Patt. Anal. Mach. Intell.* **21**(1) (1999) 65–72.
26. F. Mattern, T. Rohlfing and J. Denzler, Adaptive performance-based classifier combination for generic object recognition, *Vision, Modeling, and Visualization*, eds. G. Greiner, J. Hornegger, H. Niemann and M. Stamminger (Erlangen, Germany, 2005), pp. 139–146.
27. H. Murase and S. Nayar, Visual learning and recognition of 3-D objects from appearance, *Int. J. Comput. Vis.* **14** (1995) 5–24.
28. P. Lehel and E. E. Hemayed and A. A. Farag, Sensor planning for a trinocular active vision system, *Proc. Computer Vision and Pattern Recognition* (IEEE Computer Society Press, Fort Collins, CO, 1999), pp. II: 306–312.
29. L. Paletta and A. Pinz, Active object recognition by view integration and reinforcement learning, *Robot. Auton. Syst.* **31** (2000) 71–86.
30. E. Parzen, On the estimation of a probability density function and mode, *Ann. Math. Statist.* **33** (1962) 1065–1076.
31. R. Pito, A solution to the next best view problem for automated surface acquisition, *IEEE Trans. Patt. Anal. Mach. Intell.* **21**(10) (1999) 1016–1030.
32. J. Pösl and H. Niemann, Erscheinungsbasierte statistische objekterkennung, *Inf.–Forsch. Entwicklung* **17**(1) (2002) 21–40.
33. M. Reinhold, M. Grzegorzek, J. Denzler and H. Niemann, Appearance-based recognition of 3-D objects by cluttered background and occlusions, *Patt. Recogn.* **38**(5) (2005) 739–753.
34. M. Reinhold, D. Paulus and H. Niemann, Improved appearance-based 3-D object recognition using wavelet features, *Vision Modeling and Visualization 2001*, eds. T. Ertl, B. Girod, G. Greiner, H. Niemann and H.-P. Seidel (AKA/IOS Press, Stuttgart, Germany, 2001), pp. 473–480.
35. L. Rokach, O. Maimon and R. Arbel, Selective voting — getting more for less in sensor fusion, *Int. J. Patt. Recogn. Artif. Intell.* **20**(3) (2006) 329–350.

36. S. D. Roy, S. Chaudhury and S. Banerjee, Recognizing large 3-D objects through next view planning using an uncalibrated camera, *Int. Conf. Computer Vision* (IEEE Computer Society Press, Vancouver, Canada, 2001), pp. II: 276–281.
37. B. Schiele and J. L. Crowley, Transinformation for active object recognition, *Int. Conf. Computer Vision* (IEEE Computer Society Press, Bombay, India, 1998,), pp. 249–254.
38. R. S. Sutton and A. G. Barto, *Reinforcement Learning* (A Bradford Book, Cambridge, London, 1998).
39. A. Törn and A. Žilinskas, *Global Optimization*, Lecture Notes in Computer Science, Vol. 350 (Springer, Heidelberg, 1987).
40. J. K. Tsotsos, On the relative complexity of active vs. passive visual search, *Int. J. Comput. Vis.* **7**(2) (1992) 127–141.
41. P. Viola, *Alignment by Maximization of Mutual Information*, PhD thesis, MIT, Department of Electrical Engineering and Computer Science, Cambridge, Massachusetts (1995).
42. D. Wilkes and J. K. Tsotsos, Integration of camera motion behaviours for active object recognition, In *IEEE Workshop on Visual Behaviours*, Seattle, 1994, pp. 10–19.
43. L. Wixon and D. Ballard, Using intermediate objects to improve the efficiency of visual search, *Int. J. Comput. Vis.* **12**(2–3) (1994) 209–230.
44. Y. Ye and J. K. Tsotsos, Sensor planning for 3D object search, *Comput. Vis. Imag. Underst.* **73**(2) (1999) 145–168.
45. Y. Ye and J. K. Tsotsos, A complexity level analysis of the sensor planning task for object search, *Comput. Int.* **17**(4) (2001) 605–620.

Frank Deinzer obtained his diploma (Dipl.-Inf.) degree in 1998 and his PhD (Dr.-Ing.) in computer science in 2005 from the University of Erlangen. Currently, he holds a position as professor for computer science at the University of Applied Sciences in Wuerzburg, Germany. Before he had a position as project lead for medical image fusion at Siemens AG, Medical Solutions, Forchheim, Germany.

His research interests are in statistical fusion of sensor data in the field of medical image processing, computer vision and multimedia techniques, and has more than 30 conference/journal articles and books, and inventor resp. patentee of more than 50 patents. His work on sensor fusion for active object recognition was awarded the DAGM best paper award in 2001. In 2008 he received the innovation award of the German society for computer science. He is a member of Gesellschaft für Informatik (GI, German society for computer science).

Christian Derichs received his diploma degree in electrical engineering at the RWTH Aachen in 2003. Since 2008 he has been working for Elektrobit Automotive, Erlangen in the field of software integration for speech recognition in car entertainment/information devices. From 2004 to 2007 he worked as a research assistant at the Chair of Pattern Recognition at the Department of Computer Science at the university of Erlangen. Positioned within SFB 603 of the Deutsche Forschungsgemeinschaft (DFG) he worked in the field of the model based analysis of complex scenarios and sensor data.

In particular, his research interests include optimal sensor data fusion and view planning for active object recognition. On this topic he has contributed to a book for the Platinum Jubilee edition of the Indian Statistical Institute in 2008.

Heinrich Niemann obtained the degree of Dipl.-Ing. in electrical engineering and Dr.-Ing. from Technical University Hannover, Germany. He worked with Fraunhofer Institut für Informationsverarbeitung in Technik und Biologie, Karlsruhe, and with Fachhochschule Giessen in the Department of Electrical Engineering. Since 1975 he has been Professor of Computer Science at the University of Erlangen-Nürnberg, where he was dean of the engineering faculty of the university from 1979–1981. From 1988–2003 he was also head of the research group "Knowledge Processing" at the Bavarian Research Institute for Knowledge Based Systems (FORWISS). From 1998–2005 he was speaker of a "special research area" (SFB) entitled "Model-Based Analysis and Visualization of Complex Scenes and Sensor Data" which was funded by the German Research Foundation (DFG).

His fields of research are speech and image understanding and the application of artificial intelligence techniques in these fields. He is on the editorial board of *Pattern Recognition and Image Analysis*, *Signal, Image and Video Processing*, and *Journal of Computing and Information Technology*. He is the author or coauthor of 7 books and about 550 journal and conference contributions as well as editor or coeditor of about 30 proceedings volumes and special issues. He is a member of DAGM, GI, and IEEE and he is a Fellow of IAPR, honorary member of DAGM, and honorary Professor of Vladimir State University.

Joachim Denzler obtained the degree 'Diplom-Informatiker', 'Dr.-Ing.' and 'Habilitation' from the University of Erlangen in the years 1992, 1997 and 2003, respectively. Currently, he holds a position of a full professor for computer science and is head of the Chair for Computer Vision, Faculty of Mathematics and Informatics, Friedrich-Schiller-University of Jena.

His research interests comprise active computer vision, object recognition and tracking, 3-d reconstruction and plenoptic modeling as well as computer vision for autonomous systems. He is author and co-author of over 130 journal papers and technical articles. He is member of the IEEE, IEEE computer society, DAGM and GI. For his work on object tracking, plenoptic modeling, and active object recognition and state estimation he was awarded with DAGM best paper awards in 1996, 1999, and 2001, respectively.