Chapter 1

Information Theoretic Approaches for Next Best View Planning in Active Computer Vision

C. Derichs*, B. Deutsch*, S. Wenhardt, H. Niemann and J. Denzler†

Chair for Pattern Recognition, University of Erlangen-Nuremberg,
Martensstr. 3, 91058 Erlangen,
{derichs,deutsch,wenhardt,niemann}@informatik.uni-erlangen.de

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

This paper describes an information theoretic approach for next best view planning in active state estimation, and its application to three computer vision tasks. In active state estimation, the state estimation process contains sensor actions which affect the state observation, and therefore the final state estimate. We use the information theoretic measure of mutual information to quantify the information content in this estimate. The optimal sensor actions are those that are expected to maximally increase the information content of the estimate.

This action selection process is then applied to three seperate computer vision tasks: object recognition, object tracking and object reconstruction. Each task is formulated as an active state estimation problem. In these tasks, a given sensor action describes a camera position, or view. The information theoretic framework allows us to determine the next best view, i.e. the view that best supports the computer vision task.

We show the benefits of next best view planning in several experiments, in which we compare the estimation error produced by planned views with the error produced by regularly sampled or unchanging views.

1.1. Introduction

We present a general framework for determining the next best view in active state estimation problems. In active state estimation, the state observation process is additionally parameterized with a sensor action which is freely selectable and adaptable at run-time. The observation taken with the action which most supports the state estimation is referred to as the next best view. This framework is adapted to three computer vision tasks: object recognition, object tracking, and object reconstruction. We formulate each task as a state estimation problem and use the framework to determine the next best view.

This work is based primarily on the work of Denzler and Brown,[1] who introduce a formalism for sensor parameter optimalization in general state estimation problems and

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

demonstrated its validity with a simple object classification task (which, unlike the approach outlined later in this work, does not use Reinforcement Learning, but evaluates the mutual information directly). Such a state estimation problem produces not only an estimate of the state, but also a measure of the reliability of this estimate, in that the *a posteriori* estimate is of the form of a probability density function (pdf). The information theoretic method of maximal mutual information is used to optimize the information content of this *a posteriori* pdf by finding the optimal action, or view, *in advance*. This corresponds to minimizing the expected entropy of the pdf. We define the next best view as the one that maximizes the mutual information, and adapt this framework to our computer vision tasks.

The notion of a "next best view" depends strongly on the goals of the system. The following is a short overview of related work in active next best view selection for the three computer vision tasks:

In object recognition, a common approach for viewpoint selection is the exploitation of a few distinctive object views computed offline, as in Sipe and Casasent[2] or Dickinson.[3] Others, similar to this work, apply more complex training phases, e. g. performing a cluster analysis like Kovačič.[4] Similar to our approach, which will be shown to utilize Reinforcement Learning training, Arbel and Ferrie[5] also combine an information theoretic measure with a training stage by setting up entropy maps. But in contrast, thereby they do not consider inaccuracies in the sensor movement. Thirdly, works like Zhou and Comaniciu[6] and Laporte[7] follow the idea of omitting any training, but concentrate on passing the most supporting features to information theoretic approaches, which is convenient but naturally less reliable.

In object tracking, views have typically been changed reactively. Tordoff and Murray[8] use zoom control to keep the scale of an object of interest fixed over time. Micheloni and Foresti[9] adapt this approach with a feature clustering technique to detect moving objects, and zoom on an object if required. Both approaches do not adapt the zoom based on any predicted information gain, unlike the methods shown here. Recently, Tordoff and Murray[10] have considered the uncertainty of the estimate for zoom planning; however their approach considers the innovation covariance as an indicator to adapt the process noise, not the expected gain in information. Davison[11] also uses mutual information to guide the search of features on an image for 2-D object tracking. In contrast, the tracking discussed here optimizes the continuous parameterization of several cameras, tracking in 3-D.

In view planning for 3-D reconstruction, several authors use range scanners, e. g. Banta,[12] Pito,[13] or Scott.[14] However, these works are not comparable to 3-D reconstruction from intensity images, since the reconstruction process is completely different. Works which optimize 3-D reconstruction results by next best view planning are quite rare in literature.[15–17] Furthermore, those algorithms use geometrical considerations in contrast to the information theoretical approach introduced in this work and adopted to this special task.

This paper is organized as follows: The next section describes the general problem of state estimation, and shows how optimal views can be determined using the information theoretic measures of mutual information and entropy. Section 1.3 then applies and adapts this view planning process to three basic computer vision tasks. Section 1.4 contains ex-

periments for each of the three tasks, showing the validity of the next best view planning process. The last section contains a summary of this work.

1.2. Information theoretical approaches for Next Best View planning

In the following we treat computer vision algorithms as probabilistic state estimation problems. The unknown state is estimated by one or more observations from the environment. The probabilistic formulation of the problem makes it possible to explicitly model uncertainty, arising from using real sensors, and to use *a priori* information, which is sometimes available and can improve the quality or robustness of the results. As mentioned before, prominent examples from computer vision tackled in this article are object recognition (discrete state space with the class label being the state), object reconstruction (continuous state space with the object's points in 3-D being the state) and object tracking (continuous, time varying state space with the position, velocity and acceleration being the state).

Next best view planning is understood as acquiring those observations that are most valuable for the subsequent state estimation process. The term view is used to indicate that the focus in this work is on cameras, whose parameters (internal or external parameters) are optimized during the planning approach. Optimization is done by modeling the benefit of each new view with respect to the state estimation problem.

Strong relations can be drawn to sequential decision making. Aside from the decision (planning), which next view is taken, fusion of the information is also an important aspect. In the next two sections, we present a very general model for sequential decision making and fusion for state estimation. In the later sections, this very general model is applied to problems in computer vision.

1.2.1. *General state modeling and estimation*

The term state, or state vector $\boldsymbol{q}_t$, of a system at time step t comprises all the relevant parameters of that system to be determined from observations $\boldsymbol{o}_0 \dots \boldsymbol{o}_t$ taken by sensors. For static systems, the state does not change over time and we can omit the parameter t. For an estimation of the true state usually the *a posteriori* probability

$$p(\boldsymbol{q}_t|\boldsymbol{o}_0, \dots, \boldsymbol{o}_t) = \frac{p(\boldsymbol{o}_t|\boldsymbol{q}_t)p(\boldsymbol{q}_t|\boldsymbol{o}_0, \dots, \boldsymbol{o}_{t-1})}{p(\boldsymbol{o}_t)} \tag{1.1}$$

of the state given the observations needs to be computed using the well known Bayes formula. In (1.1), the usual Markovian assumption is made, i.e. the current observation $\boldsymbol{o}_t$ only depends on the current state $\boldsymbol{q}_t$. The *a posteriori* density can then be the basis for a maximum a posteriori estimation (MAP) or a minimum mean square error estimator (MMSE). Again, for static systems the parameter t can be omitted. For dynamic systems, the computation of the so called *temporal prior* $p(\boldsymbol{q}_t|\boldsymbol{o}_0, \dots, \boldsymbol{o}_{t-1})$ involves the *a posteriori* probability from the previous time step $t-1$ as well as the state transition probability

$p(\boldsymbol{q}_t|\boldsymbol{q}_{t-1})$ of the system, i.e.

$$p(\boldsymbol{q}_t|\boldsymbol{o}_0,\ldots,\boldsymbol{o}_{t-1}) = \int p(\boldsymbol{q}_{t-1}|\boldsymbol{o}_0,\ldots,\boldsymbol{o}_{t-1})p(\boldsymbol{q}_t|\boldsymbol{q}_{t-1})d\boldsymbol{q}_{t-1} \tag{1.2}$$

It is worth noting that for static systems, although the true state remains constant over time, the estimated *a posteriori* density will change due to the collection of observations $\boldsymbol{o}_k$. This situation will be discussed in section 1.3.1. For dynamic systems, the estimation and tracking of an evolving density is one of the main ideas behind the Kalman filter.

If the assumptions are met, the Kalman filter is sometimes more intuitive to apply due to its algebraic formulation. The basic model of the linear Kalman filter consists of two equations for state transition and observation. The state transition is given by

$$\boldsymbol{q}_{t+1} = \boldsymbol{F}_t\boldsymbol{q}_t + \boldsymbol{w}_t \tag{1.3}$$

with $\boldsymbol{W}_t$ being the covariance matrix of the Gaussian noise process $\boldsymbol{w}_t$. This equation describes the dynamics of the system and is in general equivalent to the density $p(\boldsymbol{q}_{t+1}|\boldsymbol{q}_t)$. The relation between state and observation is modeled by

$$\boldsymbol{o}_t = \boldsymbol{G}_t\boldsymbol{q}_t + \boldsymbol{r}_t \tag{1.4}$$

with $\boldsymbol{R}_t$ being the covariance matrix of the Gaussian noise process $\boldsymbol{r}_t$. Again, we can relate this equation to the density $p(\boldsymbol{o}_t|\boldsymbol{q}_t)$ in the probabilistic formulation of the problem. In the extended Kalman filter the linear relationships in (1.3) and (1.4) are substituted by in general non-linear functions $\boldsymbol{f}(\boldsymbol{q}_t)$ and $\boldsymbol{g}(\boldsymbol{q}_t)$ for state transition and observation, respectively.

The basic assumption behind the Kalman filter are Gaussian noise processes during state transition and observation, which gives us Gaussian densities for $p(\boldsymbol{q}_{t+1}|\boldsymbol{q}_t)$ and $p(\boldsymbol{o}_t|\boldsymbol{q}_t)$. As a consequence, the resulting *a priori* and *a posteriori* densities in (1.1) are also Gaussian, with the notation

$$p(\boldsymbol{q}_t|\boldsymbol{o}_0,\ldots,\boldsymbol{o}_{t-1}) \sim \mathcal{N}(\widehat{\boldsymbol{q}}_t^-, \boldsymbol{P}_t^-) \quad \text{and} \quad p(\boldsymbol{q}_t|\boldsymbol{o}_0,\ldots,\boldsymbol{o}_t) \sim \mathcal{N}(\widehat{\boldsymbol{q}}_t^+, \boldsymbol{P}_t^+) \tag{1.5}$$

and the Kalman update step (not detailed here) containing

$$\boldsymbol{P}_t^+ = (\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{G}_t)\boldsymbol{P}_t^- \tag{1.6}$$

where $\boldsymbol{K}_t$ is the *Kalman gain matrix* and $\boldsymbol{I}$ the identity matrix.

Thus, both the MAP and the MMSE estimate are the mean of (1.1). In case, that the involved densities cannot be modeled as Gaussian densities, the solution to MAP and MMSE can be achieved using particle filters. A short introduction to particle filters in practice is given in section 1.3.1.

1.2.2. *Optimality criteria for active view planning*

Having in mind that the *a posteriori* density from (1.1) is the basis of the state estimate in probabilistic estimation theory, it is quite natural to search for such observations that make (1.1) most suited for the following steps. One simple example is a MAP estimation. Having a density that consists of local maxima makes the estimation process ambiguous. A second example is a very flat density that makes the estimation process uncertain. Thus,

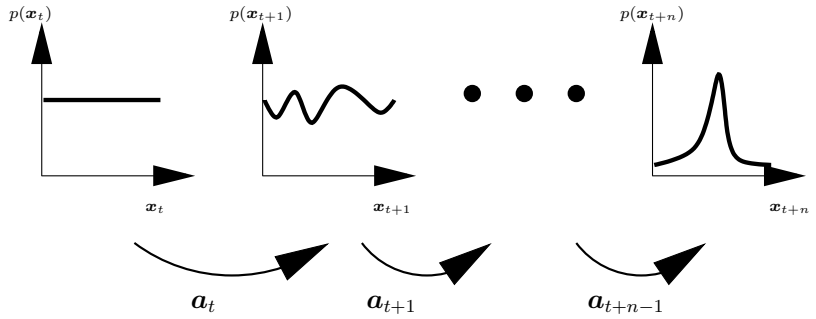


Fig. 1.1. General principle of active view planning in state estimation.

the key aspect of an optimal state estimation is the collection of those observations from the data that the resulting density in (1.1) is at the best unimodal and with small variance. This situation is indicated in Fig. 1.1. Starting with a uniform density over the state space (i.e. knowing nothing at all), we choose so called actions $\boldsymbol{a}_t$ at each time t step to influence the subsequent observations, such that the observation will lead to a more suited density. In our case, an action is any change in the internal or external parameters of a camera. However, the whole formulation is not restricted to actions for cameras. Arbitrary actions are possible, for example, one that might influence the environment (for example, the illumination of the scene) or those which select algorithms for further processing of the data.

The problem formulation directly points to the solution. To find the best action at each time step, we have to define a criterion that favors unambiguous densities with small variance. The entropy

$$H(\boldsymbol{q}_t) = \int p(\boldsymbol{q}_t) \log p(\boldsymbol{q}_t) d\boldsymbol{q}_t \tag{1.7}$$

could serve as a criterion, as done by other researchers before. However, this criterion does not allow to consider the sequential aspect of the whole planning process. Thus, we map the sequential planning process to the sender-receiver model in information theory as indicated in Fig. 1.2. The sender corresponds to the unknown state of the system. The channel and channel parameter consists of the sensor and its parameters, respectively. At the receiver side we observe the image. The question that is readily answered in information theory is the amount of information that is contained in the observation about the state, and vice versa. The quantity is the so called mutual information

$$I(\boldsymbol{q}_t; \boldsymbol{o}_t | \boldsymbol{a}_t) = \int_{\boldsymbol{q}_t} \int_{\boldsymbol{o}_t} p(\boldsymbol{q}_t|\boldsymbol{a}_t) p(\boldsymbol{o}_t|\boldsymbol{q}_t, \boldsymbol{a}_t) \log \left(\frac{p(\boldsymbol{o}_t|\boldsymbol{q}_t, \boldsymbol{a}_t)}{p(\boldsymbol{o}_t|\boldsymbol{a}_t)} \right) d\boldsymbol{o}_t d\boldsymbol{q}_t \tag{1.8}$$

and depends in our case on the chosen action, which influences the observation at time step t. Another information theoretic quantity is the conditional entropy[18]

$$H(\boldsymbol{q}_t|\boldsymbol{o}_t, \boldsymbol{a}_t) = -\int p(\boldsymbol{o}_t|\boldsymbol{a}_t) \int p(\boldsymbol{q}_t|\boldsymbol{o}_t, \boldsymbol{a}_t) \log p(\boldsymbol{q}_t|\boldsymbol{o}_t, \boldsymbol{a}_t) d\boldsymbol{q}_t d\boldsymbol{o}_t \tag{1.9}$$

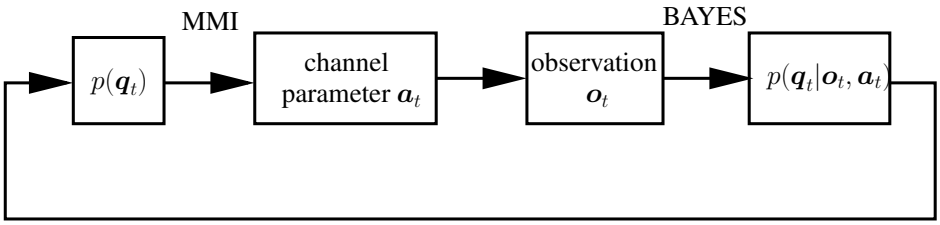


Fig. 1.2. Sequential decision process of maximum mutual information (MMI) for camera parameter selection and Bayesian update of $p(\boldsymbol{q}_t|\boldsymbol{o}_t, \boldsymbol{a}_t)$ based on the observed feature $\boldsymbol{o}_t$. Taken from Denzler and Brown.[1]

There is a nice relationship between the mutual information in (1.8) and the conditional entropy:

$$I(\boldsymbol{q}_t; \boldsymbol{o}_t|\boldsymbol{a}_t) = H(\boldsymbol{q}_t|\boldsymbol{a}_t) - H(\boldsymbol{q}_t|\boldsymbol{o}_t, \boldsymbol{a}_t) \tag{1.10}$$

The reader should note that the entropy $H(\boldsymbol{q}_t|\boldsymbol{a}_t)$ of the *a priori* probability (which lacks the current observation $\boldsymbol{o}_t$) usually does not depend on the chosen action $\boldsymbol{a}_t$. Thus, it is also possible to minimize the conditional entropy $H(\boldsymbol{q}_t|\boldsymbol{o}_t, \boldsymbol{a}_t)$, i.e. the entropy of the *a posteriori* probability (which includes the current observation $\boldsymbol{o}_t$), averaged over all possible observations.

The optimal action $\boldsymbol{a}_t^*$ with respect to the following state estimation process is now given either by the maximum of mutual information (MMI)

$$\boldsymbol{a}_t^* = \underset{\boldsymbol{a}_t}{\operatorname{argmax}}\, I(\boldsymbol{q}_t; \boldsymbol{o}_t|\boldsymbol{a}_t). \tag{1.11}$$

or equivalently by the minimum of the conditional entropy.

To complete the discussion of state estimation from Section 1.2.1, we have to add the dependency on the chosen action $\boldsymbol{a}_t$ to all densities. More precisely, the *likelihood* $p(\boldsymbol{o}_t|\boldsymbol{q}_t)$ from (1.1) will become $p(\boldsymbol{o}_t|\boldsymbol{q}_t, \boldsymbol{a}_t)$ considering that the current observation now depends not just on the state but also on the chosen action.

The optimal action $\boldsymbol{a}_t^*$ is now used to adjust the camera parameters to acquire the next best view. The observation taken by that view is fed into the estimation process (1.1) (BAYES). By definition the *a posteriori* density from (1.1) will have minimum expected entropy, i.e. *on average* minimum ambiguity and variance. Finally, this *a posteriori* density can be input directly to the whole planning process at the next time step (for time invariant systems) or by means of the temporal prior (1.2) (for dynamic systems). More details can be found in Denzler and Brown.[1] The next sections will demonstrate how this information theoretic concept can be applied to three different state estimation problems in computer vision.

1.3. Planning tasks

Given the theoretical background of chapter 1.2, we need to specify parameters like *state*, *observation* or *action* more precisely when applied to a specific application. Next to the consideration of possible problem based constraints, this is the decisive scientific transfer

work one has to perform. Therefore, this chapter introduces the adaption of the theory explained above to three applications in the research fields of computer vision. It will be shown that in all these topics, active view planning is of distinct advantage.

1.3.1. *Active Object Recognition*

Taking a look at the majority of research work in the field of object recognition, problems deal with single image processing and the assumption that the object can actually be distinguished from all others by that single view. In opposite, active object recognition searches for a selectively chosen series of different images of one object in order to combine their information for gaining the optimal improvement of certainty. So active object recognition permits the handling of more difficult classification problems, e.g. when objects show ambiguities (see Fig. 1.5) or single image quality prohibits a reliable discrimination at all.

1.3.1.1. *State representation and information fusion*

With reference to section 1.2.1, we first need to define a meaningful state $\boldsymbol{q}_t$ in the process of object recognition. In the first instance, the state has to contain the attributes we are finally interested in. Basically, always assuming a probabilistic representation of class certainties, in the most simple case this would just be a group of κ probabilities regarding all classes $\Omega_{l=1,\cdots,\kappa}$ under consideration. For each test image and object class, such a value definitely has to be calculated by the summation over a theoretically continuous set of poses $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_J)^T$ which represent the camera position relative to the object in the various dimensions:

$$p(\Omega_l) = \int_{\boldsymbol{\phi}} p(\Omega_l|\boldsymbol{\phi}) d\boldsymbol{\phi} \,. \tag{1.12}$$

So, for gaining a more distinguishing and less ambiguous state representation, it is only meaningful and free of additional effort to augment it with the pose $\boldsymbol{\phi}$, yielding $\boldsymbol{q}_t = (\Omega_\kappa, \phi_1, \ldots, \phi_J)^T$. Given this state representation we can establish (1.1) with the recognition specific parameters. Consequently, (1.1) can be considered to be a combined discrete-continuous density, i.e. discrete regarding the class assumption and continuous in $\boldsymbol{\phi}$.

When performing a camera action

$$\boldsymbol{a}_t = (\Delta\phi_1^t, \ldots, \Delta\phi_J^t) \quad \text{with} \quad \Delta\phi^t = \phi^{t+1} - \phi^t \tag{1.13}$$

relative to the object, we gather a new image whose pixel intensity value vector $\boldsymbol{v}$ provides the information to be fused to the current density representation, thus observation $\boldsymbol{o}_t = \boldsymbol{v}_t$. This fusion is a task which is generally tackled using the Kalman filter in various state estimation problems, like in section 1.3.2. But mainly due to ambiguities in the recognition process, we cannot generally assume the required normal distribution form for $p(\boldsymbol{o}_t|\boldsymbol{q}_t)$, thus making the Kalman filter unemployable here. Instead, we apply the so called *particle filters*. The basic idea is to approximate the multi-modal probability distribution by a set of M weighted samples $y^i = \{\boldsymbol{x}^i, p^i\}$. Each sample y consists of the point

$\boldsymbol{x} = (\Omega_l, \phi_1, \ldots, \phi_J)$ within the state space and the weight p for that sample, with the condition that $\sum_i p^i = 1$.

Now, each time step t a new image of the object is received—no matter if randomly or purposefully—we initiate the fusion process. In the case of the particle representation, this can be simple done by applying the Condensation Algorithm,[19] which adapts the given samples of the a priori density to an adequate representation of the a posteriori density, using the observation $\boldsymbol{o}_t$. Additionally, the camera action $\boldsymbol{a}_{t-1}$ between the image acquisition positions is considered in the sample transition:

$$\underset{\text{a posteriori } (t-1)}{p(\boldsymbol{q}_{t-1}|\,\langle \boldsymbol{o}\rangle_{t-1}, \langle \boldsymbol{a}\rangle_{t-2})} \longrightarrow \underset{\text{a priori } (t)}{p(\boldsymbol{q}_t|\,\langle \boldsymbol{o}\rangle_{t-1}, \langle \boldsymbol{a}\rangle_{t-1})} \longrightarrow \underset{\text{a posteriori } (t)}{p(\boldsymbol{q}_t|\,\langle \boldsymbol{o}\rangle_{t}, \langle \boldsymbol{a}\rangle_{t-1})} \tag{1.14}$$

where $\langle \boldsymbol{o}\rangle_k$ is the sequence of observations $\boldsymbol{o}_0 \ldots \boldsymbol{o}_k$ and $\langle \boldsymbol{a}\rangle_k$ the sequence of actions $\boldsymbol{a}_0 \ldots \boldsymbol{a}_k$.

1.3.1.2. *Optimal action selection*

So far, all discussions of section 1.3.1.1 have been so general that it does not matter whether we acquire views randomly or purposefully, since the state representation and propagation is identical. To meet the focus of this paper we now describe the optimality criteria for view planning in object recognition. Unlike the data driven solutions for object tracking and reconstruction, which will be presented in section 1.3.2 and section 1.3.3 respectively, object recognition must be approached quite differently. For recognition, we must always be aware of at least a probabilistic assumption of the properties of *all* our objects under consideration, since we need to know whether a view is discriminative or not. Obviously this information cannot be provided by an image sequence of just one object, which is all we would have in a data driven setup.

Thus, a model-based method was created, representing the features of equidistantly taken images from a circle around all possible objects. Appropriate feature vectors $\boldsymbol{c}$ are calculated by applying the well known PCA to the pixel values of these input images, represented in vector form $\boldsymbol{v}$. At this point, please note that our objective is not the improvement of an individual classifier but the determination of an optimal image acquisition strategy for an *arbitrary classifier*. Accordingly, the measure of quality is not so much the absolute classification ratio, but its increase within the the process of active image acquisition compared to a random proceeding. Thus, while omitting further details in this work, it should just be mentioned that for experimental results an eigenspace classifier[20] was used.

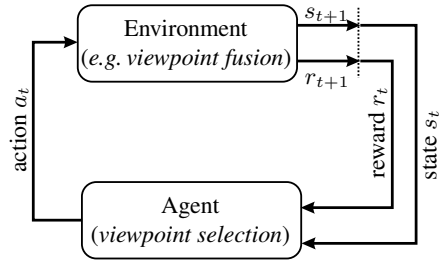


Fig. 1.3. Reinforcement learning loop

Regarding the optimization we decided for a Reinforcement Learning (RL)[21] approach utilizing a training phase, consisting of ϵ episodes with maximally $\tilde{\epsilon}$ steps each. In every single step, a closed loop between sensing s_t and acting $\boldsymbol{a}_t$ is performed (Fig. 1.3). The generally randomly chosen *action* $\boldsymbol{a}_t$ corresponds to the executed cam-

era movement. Accordingly, the *RL-state*

$$s_t := p(\boldsymbol{q}_t | \langle \boldsymbol{o} \rangle_t, \langle \boldsymbol{a} \rangle_{t-1}) \tag{1.15}$$

is the density as given in (1.1). Additionally, the classification module returns a so called *reward* r_t, which measures the quality of the chosen action. Clearly, the definition of the reward is an important aspect as this reward shall model the goal that has to be reached. Section 1.2 named the entropy to be a suitable measure of a distribution's information content, i.e. the discriminability potential in classification tasks. So setting

$$r_t = -H(s_t) = -H\left(p(\boldsymbol{q}_t | \langle \boldsymbol{o} \rangle_t, \langle \boldsymbol{a} \rangle_{t-1})\right) \tag{1.16}$$

we highly reward views that increase the information observed so far in a training episode and thus supports the goal of maximally improving the classification at every time step.

In order to optimize the viewpoint planning in an anticipatory manner, Reinforcement Learning provides the so called *return*:

$$R_t = \sum_{n=0}^{\infty} \gamma^n r_{t+n+1} \quad \text{with } \gamma \in [0;1]\,. \tag{1.17}$$

Instead of the immediate reward r_t, a γ-weighted combination of all rewards arising in later steps n of the episode is applied. During training, this is done subsequently after having finished the episode. Consequently, all acquired combinations of current state s_{t-1}, ensuing action $\boldsymbol{a}_{t-1}$ and resulting return R_t are stored in a training database.

Switching from the training phase to the later evaluation phase, naturally the future rewards cannot be observed at time step t. Thus, the following function, called the *action-value function*

$$Q(s, \boldsymbol{a}) = E\{R_t | s_t = s, \boldsymbol{a}_t = \boldsymbol{a}\} \tag{1.18}$$

is defined. It describes the expected return when starting at time step t in presumption state s with action $\boldsymbol{a}$. Please note that by calculating the expectation value of the γ-weighted and added up entropy in (1.18), $Q(s, \boldsymbol{a})$ is nothing but the conditional entropy which we postulated to be a meaningful optimization criterion in (1.9).

Trying to optimize the camera action in the evaluation phase, the first task is to extract those entries from the database that are relevant for the current state. So, for determining the similarity between the current state and each one in the database, the *extended Kullback-Leibler* **distance function** $d_{\mathrm{EKL}}(s_n, s'_m) = d_{\mathrm{KL}}(s_n, s'_m) + d_{\mathrm{KL}}(s'_m, s_n,)$ with

$$d_{\mathrm{KL}}(s_n, s'_m) = \int p(\boldsymbol{q} | \langle \boldsymbol{o} \rangle_n, \langle \boldsymbol{a} \rangle_{n-1}) \log \frac{p(\boldsymbol{q} | \langle \boldsymbol{o} \rangle_n, \langle \boldsymbol{a} \rangle_{n-1})}{p(\boldsymbol{q} | \langle \boldsymbol{o}' \rangle_m, \langle \boldsymbol{a}' \rangle_{m-1})} d\boldsymbol{q} \tag{1.19}$$

is used. Please note that in general there is no analytic solution for d_{EKL}, but as we represent our densities as sample sets anyway (see section 1.3.1.1) there are well-known ways to approximate d_{EKL} by Monte Carlo techniques.[22]

In order to provide a continuous search space to the optimization problem, we calculate a weighted sum of the action-values $Q(s', \boldsymbol{a}')$ of all previously collected state/action pairs

$(s', \boldsymbol{a}')$:

$$\widehat{Q}(s, \boldsymbol{a}) = \frac{\sum_{(s', \boldsymbol{a}')} K\left(d_{EKL}\left(\theta(s, \boldsymbol{a}), \theta(s', \boldsymbol{a}')\right)\right) \cdot Q(s', \boldsymbol{a}')}{\sum_{(s', \boldsymbol{a}')} K\left(d_{EKL}\left(\theta(s, \boldsymbol{a}), \theta(s', \boldsymbol{a}')\right)\right)} . \tag{1.20}$$

Thereby, the **transformation function** $\theta(s, \boldsymbol{a})$ transforms a presumption state s with a known action $\boldsymbol{a}$ with the intention of bringing a state to a "reference point" (required for the distance function in the next item). Actually, it simply performs a shift of the density according to the action $\boldsymbol{a}$. The **kernel function** $K(\cdot)$ finally weights the calculated distances. A suitable kernel function is, for example, the Gaussian $K(x) = \exp(-x^2/D^2)$ where D denotes the width of the kernel.

Using (1.20), the viewpoint selection problem of finding the optimal action $\boldsymbol{a}^*$ can now be written as a continuous optimization problem

$$\boldsymbol{a}^* = \mathrm{argmax}_{\boldsymbol{a}} \widehat{Q}(s, \boldsymbol{a}) \quad . \tag{1.21}$$

1.3.2. *Active Object Tracking*

For the task of visual object tracking, one is interested in the *motion* of a given object, often called the "target" and treated as a point-sized entity. To acquire this motion, the target is observed by several cameras. Using object tracking on the camera images, each camera effectively generates a two-dimensional observation from the target position. These observations are then used to recover the 3-D position of the target, as well as other indirectly observable motion parameters, such as the velocity and the acceleration. The dimensionality of the observations and the position alone require that more than one camera be used for tracking. In practice, two cameras are sufficient, though more may be used.

1.3.2.1. *State and observation representation*

The relatively low-dimensional, single-target nature of object tracking makes it an ideal candidate for the Kalman filter. In our object tracking tasks, we use a Newtonian position-velocity-acceleration motion model. The state vector $\boldsymbol{q}_t \in \mathbb{R}^9$ at time step t, which is part of the discrete-time dynamic system being observed by the Kalman filter, is defined as

$$\boldsymbol{q}_t = (x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z})^{\mathrm{T}} \tag{1.22}$$

where the component triplets correspond to the position, velocity and acceleration of the target, in world coordinates, respectively. The state transition function function $\boldsymbol{f}(\cdot)$, described in (1.3), transforms one state to the next according to

$$\begin{aligned}
x_{t+1} &= x_t + \Delta t \cdot \dot{x}_t + \tfrac{1}{2}(\Delta t^2) \cdot \ddot{x}_t & \dot{x}_{t+1} &= \dot{x}_t + \Delta t \cdot \ddot{x}_t & \ddot{x}_{t+1} &= \ddot{x}_t \\
y_{t+1} &= y_t + \Delta t \cdot \dot{y}_t + \tfrac{1}{2}(\Delta t^2) \cdot \ddot{y}_t & \dot{y}_{t+1} &= \dot{y}_t + \Delta t \cdot \ddot{y}_t & \ddot{y}_{t+1} &= \ddot{y}_t \\
z_{t+1} &= z_t + \Delta t \cdot \dot{z}_t + \tfrac{1}{2}(\Delta t^2) \cdot \ddot{z}_t & \dot{z}_{t+1} &= \dot{z}_t + \Delta t \cdot \ddot{z}_t & \ddot{z}_{t+1} &= \ddot{z}_t
\end{aligned} \tag{1.23}$$

plus an additive white Gaussian noise. In time-discrete systems, such as discussed here, Δt is a unitless factor with value 1. Note that this state and state transition system observes the

Markov property, in that the next state only depends on the current state, and not on past states. This property is necessary for applying the Kalman filter.

Since the state transition function is linear and time-invariant, we can express it as the state transition matrix $\boldsymbol{F}_t \in \mathbb{R}^{9\times 9}$. This matrix is defined as

$$\boldsymbol{F}_t = \begin{pmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{1}{2}(\Delta t^2) & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{1}{2}(\Delta t^2) & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{1}{2}(\Delta t^2) \\ 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{1.24}$$

with, again, Δt being equal to 1 in our time-discrete system. The process noise covariance $\boldsymbol{W}_t \in \mathbb{R}^{9\times 9}$ is set to a diagonal matrix for simplicity, see section section 1.4.2 for an example.

The target is observed by m cameras, each of which produce a 2-D observation: the projection of the point-sized target on each camera image. The observation $\boldsymbol{o}_t \in \mathbb{R}^{2m}$ is defined as the concatenation of all individual 2-D observations at time t:

$$\boldsymbol{o}_t = (o_{x1}, o_{y1}, \ldots, o_{xm}, o_{ym})^{\mathrm{T}} \tag{1.25}$$

with o_{xj} and o_{yj} being the horizontal and vertical coordinates reported by the jth camera and typically measured in pixels. The observation $\boldsymbol{o}_t$ is derived from the state $\boldsymbol{q}_t$ by the observation function (1.4), which is based on the perspective projection in the cameras.

For perspective projection, each camera is parameterized with its *internal* and *external* parameters. The internal parameters are the *focal lengths* ξ_u, ξ_v, the *principal point* σ_u, σ_v and possible skew or distortion parameters (not included here). The external parameters define the affine transformation between the camera coordinates and the world coordinates, given as a rotation matrix $\boldsymbol{\Phi} = (\Phi_{i,j}) \in \mathbb{R}^{3\times 3}$ and a translation vector $(\tau_x, \tau_y, \tau_z)^{\mathrm{T}}$. The actual projection of a 3-D point in world coordinates $(x, y, z)^{\mathrm{T}}$ to 2-D screen coordinates is typically modeled as a matrix multiplication in homogeneous coordinates:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} \xi_u & 0 & \sigma_u \\ 0 & \xi_v & \sigma_v \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \Phi_{0,0} & \Phi_{0,1} & \Phi_{0,2} & \tau_x \\ \Phi_{1,0} & \Phi_{1,1} & \Phi_{1,2} & \tau_y \\ \Phi_{2,0} & \Phi_{2,1} & \Phi_{2,2} & \tau_z \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{1.26}$$

where the final observation is derived by

$$\begin{pmatrix} o_x \\ o_y \end{pmatrix} = \begin{pmatrix} \frac{u}{w} \\ \frac{v}{w} \end{pmatrix} \tag{1.27}$$

Since this function is not linear (due to the division), we use the extended Kalman filter and obtain the observation matrix $\boldsymbol{G}_t \in \mathbb{R}^{2m\times 9}$ as the derivative of the observation

function about the estimated state $\widehat{\boldsymbol{q}}_t^-$, shown here for $m = 1$:

$$\boldsymbol{G}_t = \begin{pmatrix} \frac{\xi_u \cdot (\eta_z \cdot \Phi_{0,0} - \eta_x \cdot \Phi_{2,0})}{\eta_z^2} & \frac{\xi_u \cdot (\eta_z \cdot \Phi_{0,1} - \eta_x \cdot \Phi_{2,1})}{\eta_z^2} & \frac{\xi_u \cdot (\eta_z \cdot \Phi_{0,2} - \eta_x \cdot \Phi_{2,1})}{\eta_z^2} \; 0 \cdots 0 \\ \frac{\xi_v \cdot (\eta_z \cdot \Phi_{1,0} - \eta_x \cdot \Phi_{2,0})}{\eta_z^2} & \frac{\xi_v \cdot (\eta_z \cdot \Phi_{1,2} - \eta_x \cdot \Phi_{2,1})}{\eta_z^2} & \frac{\xi_v \cdot (\eta_z \cdot \Phi_{1,2} - \eta_x \cdot \Phi_{2,2})}{\eta_z^2} \; 0 \cdots 0 \end{pmatrix} \tag{1.28}$$

where $\boldsymbol{\eta} = (\eta_x, \eta_y, \eta_x)^T$ are the target world coordinates rotated and translated into the camera coordinate system, i.e. $\boldsymbol{\eta} = \boldsymbol{\Phi}\widehat{\boldsymbol{q}}_t^- + \boldsymbol{\tau}$. The zeroes to the right of the matrix correspond to the non-observable parts of the state.

Active object tracking parameterizes the observation function with an action vector $\boldsymbol{a}_t$ for each time t. This action directly affects the internal parameters, such as changing the focal length, or the external parameters, such as panning and tilting of the camera. For example, for a purely zooming camera, $\boldsymbol{a}_t = (a_1)$ is a one-dimensional factor for the focal lengths, i.e.

$$\xi_u = a_1 \cdot \xi_{u0} \tag{1.29}$$

$$\xi_v = a_1 \cdot \xi_{v0} \tag{1.30}$$

for starting focal lengths ξ_{u0} and ξ_{v0}. For a camera on a pan-tilt unit, $\boldsymbol{a}_t = (a_{\text{pan}}, a_{\text{tilt}})$ would describe the pan and tilt angles a_{pan} and a_{tilt}, respectively. These angles change the rotation matrix $\boldsymbol{\Phi}$ and possibly the translation vector $\boldsymbol{\tau}$. The observation matrix $\boldsymbol{G}_t$ is changed equivalently.

For systems with more than one camera, as is usually the case, the corresponding observation matrix is achieved by vertical concatenation of the single-camera observation matrix shown above.

1.3.2.2. *Optimal action selection*

Given the above definitions of $\boldsymbol{F}_t$ and $\boldsymbol{G}_t$, the motion of the target can be reconstructed for any action $\boldsymbol{a}_t$ by use of the Kalman filter. More specifically, this allows us to *predict* the effect any given action will have on the uncertainty of the estimate, measured by the *a posteriori* state covariance matrix $\boldsymbol{P}_t^+$ after observation $\boldsymbol{o}_t$ has been integrated into the estimate, since $\boldsymbol{P}_t^+$ does not depend on $\boldsymbol{o}_t$, seen in (1.6).

As mentioned before, we find the optimal action $\boldsymbol{a}_t^*$ by minimizing the expected entropy of the state estimate. Since the state estimate is in the form of a normal distribution, $\boldsymbol{q}_t \sim \mathcal{N}(\widehat{\boldsymbol{q}}_t, \boldsymbol{P}_t^+)$, its conditional entropy has the closed form

$$H(\boldsymbol{q}_t|\boldsymbol{a}_t) = \frac{n}{2} + \frac{1}{2}\log(2\pi^n|\boldsymbol{P}_t^+|), \tag{1.31}$$

where $|\cdot|$ denotes the determinant of a matrix. Since the covariance matrix $\boldsymbol{P}_t^+$ as calculated in eq. (1.6) depends on $\boldsymbol{a}_t$ but *not* on $\boldsymbol{o}_t$, we can simplify eq. (1.9) by pulling $H(\boldsymbol{q}_t|\boldsymbol{a}_t)$ out of the integral. The remaining integral now integrates a probability density function and is therefore 1. If we further disregard constant terms and factors, the optimality criterion is

$$\boldsymbol{a}^* = \underset{\boldsymbol{a}_t}{\operatorname{argmin}} \log|\boldsymbol{P}_t^+| \,. \tag{1.32}$$

The logarithm could even be dropped due to its monotony. Due to the independence of $\boldsymbol{P}_t^+$ from $\boldsymbol{o}_t$, we can find the optimal action *before* the associated observation is made.

1.3.2.3. *Visibility*

However, even though the actual value of $\boldsymbol{o}_t$ is not relevant, the *presence* of an observation is. If no complete observation can be made at a certain time step, the Kalman update step cannot be performed. In this case, the *a posteriori* state estimate uncertainty is unchanged from the *a priori* state estimate uncertainty. In other words, $\boldsymbol{P}_t^+ = \boldsymbol{P}_t^-$.

In many cases, this availability of an observation depends on the camera action. Consider the classic focal length dilemma. Using a large focal length (zooming in) gives the best view of an object, but the object risks moving outside the field of view of the camera. Using a small focal length (zooming out) reduces the risk of losing the object, but the object is now very small in the image, and a tracking error of one pixel translates to a much larger world coordinate distance. So the optimal focal length is most likely in between: small enough not to lose the object, but large enough to gain the most information.

In object tracking, each observation is a point on the image plane of a camera (or concatenation of several such points). Since the camera sensor is finite, there are points on this plane which do not lie on the sensor. We will call observations on the camera sensor *visible* observations, and those outside the sensor *non-visible* observations. The impact of this classification is that states that the observation function maps to non-visible observations would not generate any observation at all, i.e. the update step would be skipped.

Assume that we could partition the set of observations into the sets of *visible* observations $\mathcal{O}_v$ and *non-visible* observations $\mathcal{O}_{\neg v}$, and revisit equation (1.9):

$$H(\boldsymbol{q}_t|\boldsymbol{o}_t, \boldsymbol{a}_t) = \int p(\boldsymbol{o}_t|\boldsymbol{a}_t) H(\boldsymbol{q}_t|\boldsymbol{a}_t) \mathrm{d}\boldsymbol{o}_t \tag{1.33}$$

$$= \int\limits_{\boldsymbol{o}_t \in \mathcal{O}_v} p(\boldsymbol{o}_t|\boldsymbol{a}_t) H(\boldsymbol{q}_t|\boldsymbol{a}_t) \mathrm{d}\boldsymbol{o}_t + \int\limits_{\boldsymbol{o}_t \in \mathcal{O}_{\neg v}} p(\boldsymbol{o}_t|\boldsymbol{a}_t) H(\boldsymbol{q}_t|\boldsymbol{a}_t) \mathrm{d}\boldsymbol{o}_t \tag{1.34}$$

due to the summation rule of integers. Given that $H(\boldsymbol{q}_t|\boldsymbol{a}_t)$ is independent of $\boldsymbol{o}_t$ in the Kalman filter case, *except* for the membership of $\boldsymbol{o}_t$ in $\mathcal{O}_v$ or $\mathcal{O}_{\neg v}$, the entropy $H(\boldsymbol{q}_t|\boldsymbol{a}_t)$ can only have (or rather, be proportional to) one of two values:

$$H(\boldsymbol{q}_t|\boldsymbol{a}_t) \propto \begin{cases} \log|\boldsymbol{P}_t^+| & \text{if a visible observation occurs,} \\ \log|\boldsymbol{P}_t^-| & \text{otherwise.} \end{cases} \tag{1.35}$$

This simplifies equation (1.34) to

$$H(\boldsymbol{q}_t|\boldsymbol{o}_t, \boldsymbol{a}_t) \propto \int\limits_{\boldsymbol{o}_t \in \mathcal{O}_v} p(\boldsymbol{o}_t|\boldsymbol{a}_t) \log|\boldsymbol{P}_t^+| \mathrm{d}\boldsymbol{o}_t + \int\limits_{\boldsymbol{o}_t \in \mathcal{O}_{\neg v}} p(\boldsymbol{o}_t|\boldsymbol{a}_t) \log|\boldsymbol{P}_t^-| \mathrm{d}\boldsymbol{o}_t \tag{1.36}$$

$$= \log|\boldsymbol{P}_t^+| \cdot \int\limits_{\boldsymbol{o}_t \in \mathcal{O}_v} p(\boldsymbol{o}_t|\boldsymbol{a}_t) \mathrm{d}\boldsymbol{o}_t + \log|\boldsymbol{P}_t^-| \cdot \int\limits_{\boldsymbol{o}_t \in \mathcal{O}_{\neg v}} p(\boldsymbol{o}_t|\boldsymbol{a}_t) \mathrm{d}\boldsymbol{o}_t \tag{1.37}$$

$$= w \cdot \log|\boldsymbol{P}_t^+| + (1-w) \cdot \log|\boldsymbol{P}_t^-| \tag{1.38}$$

with w being the probability that the to-be-acquired observation will be visible.

Obviously, w depends on $\boldsymbol{a}_t$. The probability w can be calculated for each $\boldsymbol{a}_t$ by regarding the observation estimate. In the Kalman filter case, the observation follows a

normal distribution, with $\boldsymbol{o}_t \sim \mathcal{N}(\boldsymbol{g}(\widehat{\boldsymbol{q}}_t^-, \boldsymbol{a}_t), \boldsymbol{S}_t)$. The probability w is then the integral over the area of visible observations:

$$w = \int_{\boldsymbol{o}_t \in \mathcal{O}_v} p(\boldsymbol{o}_t|\boldsymbol{a}_t)\mathrm{d}\boldsymbol{o}_t = \int_{\boldsymbol{o}_t \in \mathcal{O}_v} \mathcal{N}(\boldsymbol{g}(\widehat{\boldsymbol{q}}_t^-, \boldsymbol{a}_t), \boldsymbol{S}_t)\mathrm{d}\boldsymbol{o}_t \tag{1.39}$$

In object tracking, $\mathcal{O}_v$ is a rectangular area in the observation space (for each camera). A closed solution exists for this problem. For more than one camera, we will assume that the Kalman update step can only be performed if all cameras produce a visible observation. In this case, w is the product of all individual visibility probabilities.

1.3.2.4. *Multi-step action selection*

The above method selects the optimal view if all sensor actions are equally valid and reachable. However, in real world systems, the range of actions available for the next camera image may be considerably reduced. For example, in zoom planning, the speed of the zoom motor in the camera determines how far the zoom settings can be changed in the time between two camera images.

Generally, we associate a *cost* with each action. If the costs of different actions are not equal, and depend on the previous action (available zoom settings depend on the current motor position, for example), the above method may not yield the optimal settings. Instead, we must evaluate a *sequence* of future views. Planning a sequence of actions, especially given computation time constraints, is discussed in more detail in Deutsch et al.[23]

1.3.3. *Active Object Reconstruction*

We study the problem of finding the next best view in 3-D reconstruction from intensity images, using the above introduced information theoretical algorithm. We show how the general algorithm can be adopted to the special task and discuss some boundary conditions.

1.3.3.1. *State and observation representation*

Similar to active object tracking (cf. section 1.3.2), we use a Kalman filter approach.[24] Therefore, we use the same notations. The 3-D reconstruction is represented by a list of i 3-D points, concatenated to the *state vector* $\boldsymbol{q}_t \in \mathbb{R}^{3i}$. Since the coordinates of the reconstructed 3-D points are constant in time, the state transition matrix $\boldsymbol{F}_t$ is the identity matrix $\boldsymbol{I} \in \mathbb{R}^{3i}$ and there is no noise in this process, i. e. the noise covariance $\boldsymbol{W}_t = \boldsymbol{0}$. Further, the state does not depend on time: $\boldsymbol{q} = \boldsymbol{q}_t$.

The state estimate is represented by the state vector, as defined above, and the covariance $\boldsymbol{P}_t$. We assume each estimate of the 3-D point coordinates is independent of the other ones, so the covariance $\boldsymbol{P}_t$ has a block diagonal structure with 3×3 blocks. As we will see below this allows a efficient evaluation of a certain view.

The observation $\boldsymbol{o}_t$ is a concatenation of the 2-D projections of the 3-D points in each time step t. In contrast to section 1.3.2, we do not have one object, which is observed by m cameras, but we have i points, which are observed by one camera. So the dimension of $\boldsymbol{o}_t$

Fig. 1.4. SCORBOT (left) and turn table (right)

is $\mathbb{R}^{2i}$. The observation is assumed to be noisy, with an additive Gaussian Noise with zero mean and covariance $\boldsymbol{R}_t$. The observation function $\boldsymbol{g}(\boldsymbol{q}_t, \boldsymbol{a}_t)$ depends on the modifiable camera parameters $\boldsymbol{a}_t$ (focal length, rotation and translation of the camera), non modifiable ones (principal point, skew and distortion parameters), which are not denoted explicitly, and projects the vector of the 3-D points $\boldsymbol{q}_t$ to the 2-D image plane by the perspective projection model. Therefore, we can use equations (1.26) and (1.27) to evaluate $\boldsymbol{g}(\cdot, \cdot)$.

Since $\boldsymbol{g}(\cdot, \cdot)$ is a nonlinear function, we have to use the extended Kalman filter, which uses a first order Taylor approximation to linearize $\boldsymbol{g}(\cdot, \cdot)$. Thus, we need the Jacobian $\boldsymbol{G}$ of $\boldsymbol{g}(\cdot, \cdot)$, which is derived analytically from equations (1.26) and (1.27). Incidently, it is easy to show that the first order Taylor approximation and the paraperspective projection model, well known in computer vision, is equivalent.

1.3.3.2. *Optimal action selection*

The goal of next best view selection is to find the optimal next view point $\boldsymbol{a}_t^*$ to improve the reconstruction accuracy. One optimality criterion is to reduce the uncertainty in the state estimation, which is measured in information theory by its entropy $H(\boldsymbol{q}|\boldsymbol{a}_t)$. This entropy, however, has to be calculated *a priori* to optimize the view before obtaining a new image.

Therefore, we need to determine the expected entropy $H(\boldsymbol{q}|\boldsymbol{o}_t, \boldsymbol{a}_t)$. The expected entropy is the mean of the entropy of $\boldsymbol{q}$ over all observations and was introduced in equation (1.9). The optimality criterion is the determination of the view $\boldsymbol{a}_t^*$ which maximizes the mutual information (cf. eq. (1.11)), which is equivalent to minimizing the conditional entropy. As in (1.32), this corresponds to minimizing the logarithm of the determinant of $\boldsymbol{P}_t$. Since $\boldsymbol{P}_t$ is a block diagonal matrix with blocks $\boldsymbol{P}_t^{(k)}, k = 1, \ldots i$ the calculation can be

simplified to

$$\boldsymbol{a}^* = \underset{\boldsymbol{a}_t}{\operatorname{argmin}} \log \prod_{k=1}^{i} |\boldsymbol{P}_t^{(k)}| = \underset{\boldsymbol{a}_t}{\operatorname{argmin}} \sum_{k=1}^{i} \log |\boldsymbol{P}_t^{(k)}|. \tag{1.40}$$

So the high computational complexity of calculation of the determinant of a $3i \times 3i$ covariance can be reduced to i calculations of the determinant of 3×3 matrices.

Some constraints on the modifiable camera parameters $\boldsymbol{a}_t$ must be considered. Not every optimal view point, in the sense of (1.32), results in a usable image. Some examples of effects that can make a view point completely or partly unusable are:

- Field of view: all 3-D points to be reconstructed have to be visible in the image, otherwise they cannot observed by the camera. We can ensure this by backprojecting the mean of 3-D estimate of the points to the image plane. If this projection is in the image, we assume that this point is visible.
- Occlusion: again, the 3-D points must be visible in the image. But this constraint may fail for a point, because the point is occluded by parts of the object itself or by the robot arm. This condition is independent of the upper one, since the projection of one point can be in the image, but it is not visible if it lies behind another surface. This constraint is not modeled for the experiments, because we analyze only flat objects. So self occlusions do not occur in this case.
- Reachability: the view point must be reachable by the robot. To ensure this, we use the 4 by 4 *Denavit-Hartenberg* matrix,[25] which depends on the angles of the rotation axes and the distances between the joints, to calculate the transformation to a fixed world coordinate system. Since the lengths are fixed, only the angles are relevant.

These constraints determine the search space for our optimization. We are able now to search for the optimal view point $\boldsymbol{a}_t^*$ with an exhaustive search over the discretely sampled action space. The action space in our case is the space with all reachable angles of the joints of the robot. If the expected observation contains image points outside the field of view, we discard this sample. The best-rated undiscarded sample is the next best view.

1.4. Experiments

1.4.1. *Evaluation for Active Object Recognition*

In order to show the benefit of active object recognition we should be able to point out that—compared to an unplanned proceeding—we gain an enhancement in classification results after the same number of views. To satisfy the demands of arbitrary applications for our approach, non-synthetic objects with ambiguities are favorable to be evaluated. So we decided on the toy manikins shown in Fig. 1.5, provided with a quiver, a lamp, a bib, or any of the eight possible combinations of these equipments, consequently arranging a classification problem with eight classes. Image acquisition was done by fixing the manikins on a turntable while getting images from a camera located at a fixed position. Moving the

turntable, we cover a circular action space with $J = 1$, as mentioned in section 1.3.1.1. For providing perfect ambiguities to our algorithm at times, we work on fixed croppings of the original images, which can be regarded as zoomed-in image acquisition. This way, decisive equipments can just drop out of the scope, thus from time to time even the best classifier cannot decide reliably on the object class when given a single image.

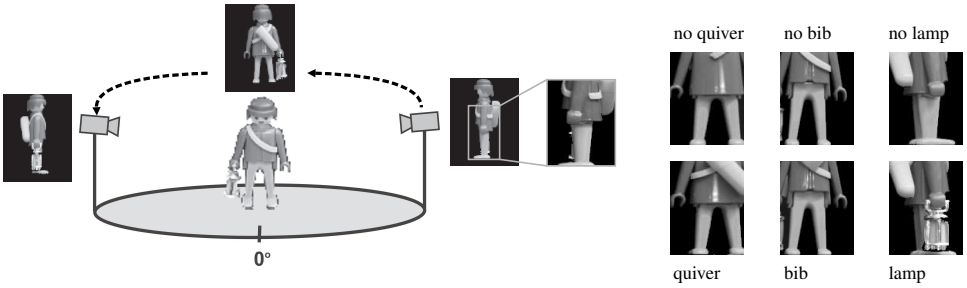


Fig. 1.5. Views of the toy manikin object classes

For our purpose, we chose steps of 1.0 degree in the horizontal direction to gain a fundamental image set of 360 entries per class. Taking every other image and calculating its features (see 1.3.1.2), we can construct the underlying model.

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 that already appear in the model representation.

During the Reinforcement Learning training phase, for each class in the database we now provide ν episodes of randomly chosen sensor actions and resulting images to the algorithm. Each episode contains at most eight steps of image retrieval and consecutive information fusion. Following the intent of this paper, we consequently applied the entropy reward (1.16) during Reinforcement Learning for rating positions, i.e. camera actions. Handling the two-dimensional space, the density representation depends on $M = 2880$ particles altogether, that is 360 particles per class.

Concerning the influence of variable parameters, in a first instance we chose two different values for the number of training episodes $\nu \epsilon \{3, 50\}$ which provides us with two differently reliable databases. Additionally we tested two variations of the weighting $\gamma \epsilon \{0, 0.5\}$ and two kernel parameters $D \epsilon \{2, 5\}$. Fig. 1.6 shows the corresponding classification results in each step, compared to those generated by unplanned sensor action. Results of the planned and random sequences were computed relying on 250 episodes with a maximum of eight steps for each parameter combination and object class.

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 when not performing arbitrary sensor movements. Especially early steps within an episode $(t = 2, 3)$ partially gain a benefit of more then 10% in classification rate.

Furthermore, it is observable that the kernel parameter D and the step influence param-

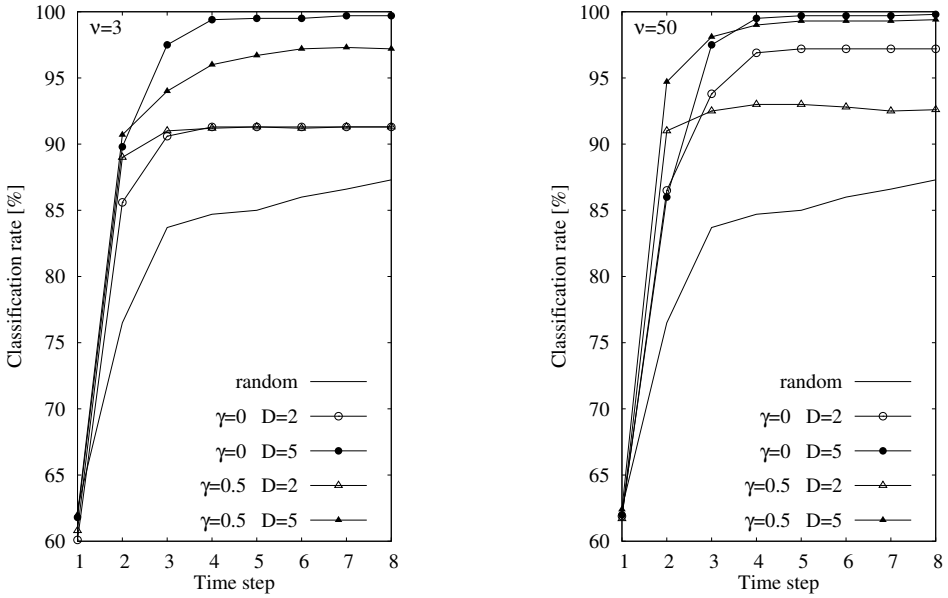


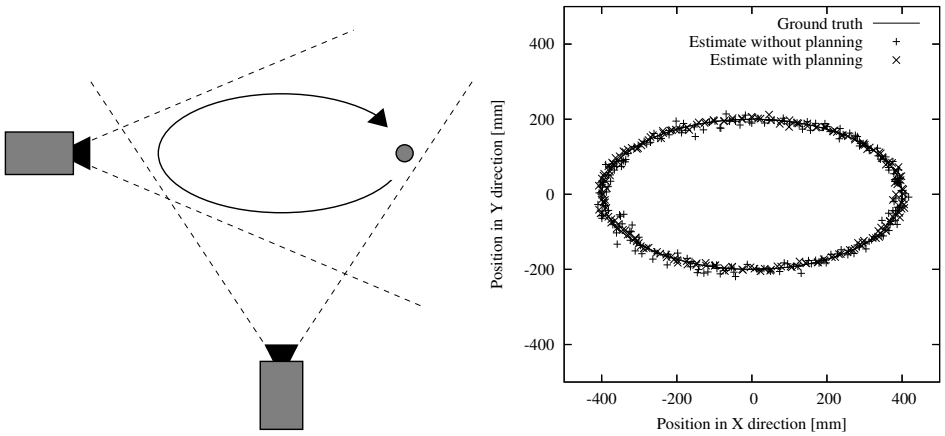
Fig. 1.6. Classification rate after t time steps of planned and unplanned viewpoint selection. For the planned variation combinations of the free parameters ν, γ and D are evaluated. Ratios at the first time step differ, because initial viewpoints were selected randomly.

eter γ can be altered within an adequate range of values without loosing the view planning benefit compared to the random proceeding. Additionally, a quite small database with only $\nu = 3$ training episodes per class already causes a significant advantage of the planned proceeding. Thus, a possibly desired adaption of training time because of computation time constraints appears to be feasible within a very wide range.

1.4.2. *Evaluation for Active Object Tracking*

The next best view planning for active object tracking is evaluated in a simulation. Object tracking is a relatively easy task to simulate, and ground truth and repeatability are also present. The experimental setup is visible in figure 1.7(a). Two cameras, at right angles, observe a point-shaped tracking target moving in an ellipsis in the center of the scene. The target is tracked by a Kalman filter using a polynomial motion model, as described in section 1.3.2.1. The axes have lengths 400mm and 800mm, respectively. The state transition noise matrix $\boldsymbol{W}_t \in \mathbb{R}^{9\times 9}$ is a constant diagonal matrix, corresponding to a standard deviation in the position of 100mm, in the velocity of $100\text{mm}/\Delta t$ and in the accelleration of $10\text{mm}/(\Delta t)^2$.

The generated observations are perturbed by white Gaussian noise with zero mean and a known covariance matrix $\boldsymbol{R}_t \in \mathbb{R}^{4\times 4}$, corresponding to a standard deviation of 1% of the image width and height for both cameras. If a perturbed observation lies outside the simulated camera field-of-view, it is discarded and is not used to update the Kalman filter.



(a) simulation setup (b) ground truth and estimations

Fig. 1.7. Overview of the object tracking simulation (1.7(a)). Two cameras with variable focal lengths observe an object moving on an elliptical path. Fig. 1.7(b) shows the ground truth object path for two full cycles and the estimated positions without and with view planning.

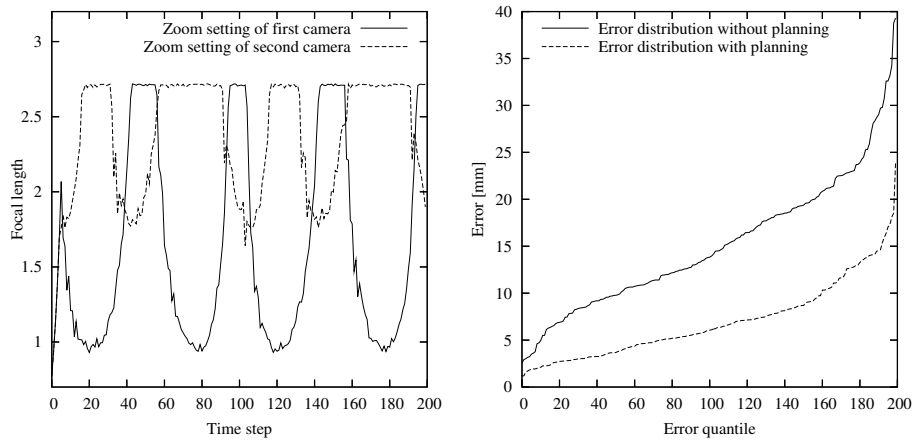
A time step without update results in the *a posteriori* state probability being the same as the *a priori* state probability, with a much larger state entropy $H(\boldsymbol{q}_t|\boldsymbol{a}_t)$ and a larger expected estimation error. This reflects the fact that we have lost information about the target due to the noisy state transition, but not regained it with up-to-date observations. This forces the view optimization system to incorporate the expected visibility in order to avoid such information loss. The target is (potentially) reacquired in the next time step.

Each camera can change its focal length within a certain range, i.e. $\boldsymbol{a}_t = (a_1, a_2)^{\mathrm{T}}$ with a_j the focal length factor for camera j. For comparison, we also run the experiment with a fixed focal length, chosen in such a way that the object is always visible.

Figure 1.7(b) shows the ground truth path of the target and the estimated positions with and without zoom planning. It can be seen that the estimation is error-prone due to the observation noise. The average error was 15.11mm without planning vs. 6.93mm with planning. This is a reduction to 45.9% of the original error.

Figure 1.8(a) shows the zoom levels each camera assumes at each of 200 time steps (two full object cycles) during zoom planning. Higher focal length values correspond to a narrower field of view. Each camera follows the object, keeping the expected projection close to its image borders. This allows the maximal focal length to be used at all times, minimizing the effect of the observation noise on the estimation. One should note that this is entirely emergent behavior, resulting solely from the minimization of expected entropy.

Figure 1.8(b) shows the distribution of the Euclidian distance between the estimated position and the ground truth position, the estimation error. These distributions were obtained by acquiring the error at each time step, and then sorting all errors by magnitude. This



(a) zoom levels during tracking

(b) tracking error distribution

Fig. 1.8. The zoom levels assumed assumed by the two cameras during zoom planning. (1.8(a)) and the distribution of the Euclidian estimation error for simulations with planned and non-planned views, sorted by error value (1.8(b)).

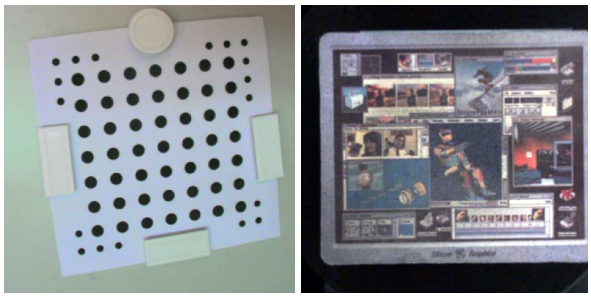
representation allows us to see that view planning gives an error which is generally lower than without view planning. However, the error can rise almost high as the non-planned case (near the right side of the graph), in the event of an object loss in one or both cameras.

1.4.3. *Evaluation for Active Object Reconstruction*

We verify our approach for next best view planning for 3-D reconstruction with real world experiments. We use a Sony DFW-VL500 firewire camera, whose intrinsic parameters were calibrated by Tsai's algorithm.[26] The camera is moved by the SCORBOT in the first experiment and by the turn table with tilting arm in the second one (cf. Fig. 1.4). The extrinsic parameters are calculated by the Denavit-Hartenberg matrix and the hand-eye transformation, which is acquired by the algorithm of Schmidt.[27] The first experiment reconstructs a calibration pattern, the second a mouse pad.

In both experiments, we start with an initial estimation, obtained by triangulation from an image pair from two view points. This gives us an initial estimate of $\boldsymbol{q}$. The initial covariance matrix $\boldsymbol{P}_0$ is set to a diagonal matrix $\mathrm{diag}(10, \ldots, 10)$, as we assume that the uncertainty is equal in each direction.

To evaluate the expected uncertainty, we calculate the determinant of $\boldsymbol{P}_t$ (eq. (1.32)). The Jacobian $\boldsymbol{G}_t(\boldsymbol{a}_t)$ of the observation function depends on the axis values of the robot and must be calculated for each candidate view point. The computation time for the next best view for the SCORBOT (5 degrees of freedom, due to its 5 axes, 384000 view points analyzed, 49 3-D points) is about 9 minutes on a system with an Pentium IV processor



(a) calibration pattern (b) mouse pad

Fig. 1.9. Images taken during the experiments (1.9(a) from first and 1.9(b) from second)

with 3 GHz, and 1 GB RAM, and about 45 seconds for the turntable (2 degrees of freedom, 2000 view points analyzed, 50 3-D points). The computation time is linear in the number of points.

1.4.3.1. *Reconstructing a Calibration Pattern*

A calibration pattern (cf. Fig. 1.9(a)) is viewed from the top of the SCORBOT. The pattern simplifies the acquisition of 2-D points, and allows us to compare our results with ground truth data. After the initialization, we start the optimization process to take new images from the optimal view point.

Table 1.1 shows the results for the first 5 iterations in the optimized case and a non-optimized one. The images for the non-optimized view points were taken by alternating between the two initial positions.

By construction, the determinant of $\boldsymbol{P}_t$ is reduced faster in the optimized case than in the non-optimized case. Additionally, the mean of the errors of all points decreases after each time step, except for some outliers. This rise in error is not a contradiction to the decrease in uncertainty, since the Kalman filter cannot judge the quality of an observation.

The view points are shown in Fig. 1.10. After the initialization steps (middle top) the optimized view points lie as expected: the cameras are opposite each other and the angle between each line of sight is approx. 90 degrees.

1.4.3.2. *Reconstructing a Mouse Pad*

In this experiment we use a mouse pad (cf. Fig. 1.9(b)), requiring us to track feature points during movement, using the algorithm of Zinsser.[28] However, only the tracked points from the optimal positions are used to update the state estimation. Integration of the points tracked *en route* to the optimal positions is possible, but this would prevent a comparison of two view point sequences due to a diverging number of integrated observations.

Table 1.2 shows the root mean square error between the reconstructed 3-D points and

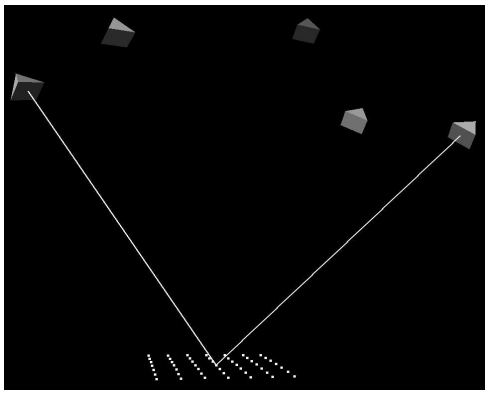


Fig. 1.10. View points for reconstruction of the calibration pattern, with two lines of sight for one point from different view points. We can observe, the angle between the lines of sight is approximately 90 degrees.

Table 1.1. First experiment: μ_t is the mean of the difference between reconstructed points and the ground truth data in mm, σ_t is the standard deviation of this error, $|\boldsymbol{P}_t|$ is the determinant of the covariance matrix. We display the values for the optimized and a non-optimized view point sequence, which is taken by the SCORBOT.

	optimized			non-optimized		
t	μ_t	σ_t	$\lvert\boldsymbol{P}_t\rvert$	μ_t	σ_t	$\lvert\boldsymbol{P}_t\rvert$
1	0.132	0.080	7.281	0.132	0.080	7.281
2	0.128	0.079	1.762	0.125	0.072	3.338
3	0.115	0.062	0.705	0.128	0.073	1.468
4	0.108	0.062	0.385	0.129	0.074	0.905
5	0.107	0.061	0.244	0.127	0.074	0.531

their regression plane, as well as the trend of the covariance matrix $\boldsymbol{P}_t$, for the first 5 iterations. We compare the values from the optimized view points to an experiment with view points uniformly distributed on a circle perpendicular to the rotation axis of the turn table, and to one completely random view point sequence on the half sphere. The error decreases fastest in the optimized case, signifying a measurable benefit from view point optimization.

1.5. Summary

We have described an information theoretic framework for selecting the next best view in a computer vision task, based on mutual information. We have applied this framework to three typical computer vision tasks: object recognition, tracking and reconstruction.

In object tracking, the state is the position, velocity and acceleration of an object. This object is observed by several cameras, whose internal or external parameters can change. Next best view planning selects the optimal parameters for the estimation of the object state. We have shown the benefit of next best view planning for object tracking in a simulation

Table 1.2. Second experiment: μ_t is the mean of the root mean square error of the points to their regression plane in mm, $|\boldsymbol{P}_t|$ the determinant of the covariance matrix after each iteration. The optimized, one uniform and one random view point sequence are shown.

	optimized		circle		random	
t	μ_t	$\lvert\boldsymbol{P}_t\rvert$	μ_t	$\lvert\boldsymbol{P}_t\rvert$	μ_t	$\lvert\boldsymbol{P}_t\rvert$
1	0.073	8.62	0.073	8.62	0.073	8.65
2	0.050	1.75	0.041	1.98	0.054	2.76
3	0.033	0.636	0.038	0.845	0.043	1.20
4	0.030	0.315	0.038	0.428	0.041	0.479
5	0.026	0.175	0.041	0.235	0.041	0.329

with cameras with a changeable focal length. Using next best view planning, the tracking error is noticably lower than the same task with fixed focal lengths. In our case, we were able to reduce the estimation error to 45.9% of the error in the fixed focal length setup.

In 3-D reconstruction, the state consists of the 3-D coordinates of the reconstructed points and the observation consists of tracked feature points. Additional constraints, field of view, occlusions, and reachability have to be considered to get a feasible next best view. In two real world experiments, we have shown that optimal selected view points reduce the reconstruction error significantly. In one experiment the error was reduced to 63.4% of the error of unplanned views.

References

1. J. Denzler and C. Brown, An information theoretic approach to optimal sensor data selection for state estimation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **24**(2), 145–157, (2002).
2. M. A. Sipe and D. Casasent, Feature space trajectory methods for active computer vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **24**(12), 1634–1643, (2002).
3. S. J. Dickinson, Active object recognition integrating attention and viewpoint control, *Computer Vision and Image Understanding*. **67**(3), 239–260, (1997).
4. S. Kovačič, A. Leonardis, and F. Pernuš, Planning sequences of views for 3-D object recognition and pose determination, *Pattern Recognition*. **31**(10), 1407–1417, (1998).
5. T. Arbel and F. Ferrie, Entropy-based gaze planning, *Image and Vision Computing*. **19**(11), 779–786, (2001).
6. X. S. Zhou, D. Comaniciu, and A. Krishnan. Conditional feature sensitivity: A unifying view on active recognition and feature selection. In *Proceedings of the 9th International Conference on Computer Vision*, pp. 1502–1509, Nice, France, (2003).
7. C. Laporte, R. Brooks, and T. Arbel. A fast discriminant approach to active object recognition and pose estimation. In *Proceedings of the 17th International Conference on Pattern Recognition*, pp. 91–94, Cambridge, (2004).
8. B. J. Tordoff and D. W. Murray, Reactive Control of Zoom while Fixating Using Perspective and Affine Cameras, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **26**(1), 98–112, (2004).
9. C. Micheloni and G. L. Foresti. Zoom On Target While Tracking . In *International Conference on Image Processing - ICIP'05* , pp. 117–120, Genua, Italy, (2005).

10. B. J. Tordoff and D. W. Murray, A method of reactive zoom control from uncertainty in tracking, *Computer Vision and Image Understanding*. **105**(2), 131–144, (2007).
11. A. Davison. Active search for real-time vision. In *Proceedings of the International Conference on Computer Vision (ICCV)*, vol. 1, pp. 66–73, Beijing, China, (2005).
12. J. E. Banta, L. R. Wong, C. Dumont, and M. A. Abidi, A next-best-view system for autonomous 3-d object reconstruction., *IEEE Transactions on Systems, Man, and Cybernetics, Part A*. **30**(5), 589–598, (2000).
13. R. Pito, A solution to the next best view problem for automated surface acquisition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **21**(10), 1016–1030, (1999).
14. W. Scott, G. Roth, and J.-F. Rivest, View planning for automated three-dimensional object reconstruction and inspection, *ACM Computing Surveys*. **35**(1), 64–96, (2003).
15. K. N. Kutulakos and C. R. Dyer, Recovering shape by purposive viewpoint adjustment, *International Journal of Computer Vision*. **12**(2), 113–136, (1994).
16. E. Marchand and F. Chaumette, Active vision for complete scene reconstruction and exploration, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **21**(1), 65–72, (1999).
17. W. Niem, *Automatische Rekonstruktion starrer dreidimensionaler Objekte aus Kamerabildern*. (VDI Verlag GmbH, 1999).
18. A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. (McGraw-Hill, Inc, Singapore, 2002), 4-th edition.
19. M. Isard and A. Blake, CONDENSATION — Conditional Density Propagation for Visual Tracking, *International Journal of Computer Vision*. **29**(1), 5–28, (1998).
20. H. Murase and S. Nayar, Visual learning and recognition of 3-D objectsfrom appearance, *International Journal of Computer Vision*. **14**, 5–24, (1995).
21. R. S. Sutton and A. G. Barto, *Reinforcement Learning*. (A Bradford Book, Cambridge, London, 1998).
22. F. Deinzer, J. Denzler, C. Derichs, and H. Niemann. Aspects of optimal viewpoint selection and viewpoint fusion. In *Computer Vision – ACCV 2006*, pp. 902–912, Hyderabad, India, (2006).
23. B. Deutsch, S. Wenhardt, and H. Niemann. Multi-Step Multi-Camera View Planning for Real-Time Visual Object Tracking . In *Pattern Recognition - 28th DAGM Symposium* , pp. 536–545, Berlin, (2006).
24. R. Kalman, A new approach to linear filtering and prediction problems, *Journal of Basis Engineering*. **82**, 35–44, (1960).
25. J. J. Craig, *Introduction to Robotics: Mechanics and Control*. (Prentice Hall, Upper Saddle River, USA, 2004), 3-rd edition.
26. R. Y. Tsai, A Versatile Camera Calibration Technique for High–Accuracy3D Machine Vision Metrology Using Off–the–ShelfTV Cameras and Lenses, *IEEE Journal of Robotics and Automation*. **RA–3**(4), 323–344, (1987).
27. J. Schmidt, F. Vogt, and H. Niemann. Vector Quantization Based Data Selection for Hand-Eye Calibration. In *Vision, Modeling, and Visualization 2004*, pp. 21–28, Stanford, USA, (2004).
28. T. Zinßer, C. Gräßl, and H. Niemann. Efficient Feature Tracking for Long Video Sequences. In *Pattern Recognition, 26th DAGM Symposium*, pp. 326–333, Tübingen, Germany, (2004).