Active Object Recognition on a Humanoid Robot

Björn Browatzki, Vadim Tikhanoff, Giorgio Metta, Heinrich H. Bülthoff and Christian Wallraven

Abstract— **Interaction with its environment is a key requisite for a humanoid robot. Especially the ability to recognize and manipulate unknown objects is crucial to successfully work in natural environments. Visual object recognition, however, still remains a challenging problem, as three-dimensional objects often give rise to ambiguous, two-dimensional views. Here, we propose a perception-driven, multisensory exploration and recognition scheme to actively resolve ambiguities that emerge at certain viewpoints. We define an efficient method to acquire two-dimensional views in an object-centered task space and sample characteristic views on a view sphere. Information is accumulated during the recognition process and used to select actions expected to be most beneficial in discriminating similar objects. Besides visual information we take into account proprioceptive information to create more reliable hypotheses. Simulation and real-world results clearly demonstrate the efficiency of active, multisensory exploration over passive, vision-only recognition methods.**

I. INTRODUCTION

One major difficulty in computational object recognition lies in the fact that a 3D object can be viewed from an infinite number of viewpoints. Indeed, objects with different 3D shapes often share similar 2D views. Humans are able to resolve this kind of ambiguity easily by producing additional views through object manipulation or self movement. In both cases the action made provides proprioceptive information that is closely linked to the visual information retrieved from the obtained views. This mode of exploration can be observed already very early in infants, when they start to interact with objects in their environment. Following this principle, we propose an active method for a humanoid robot that allows an efficient in-hand object exploration and a perception-driven recognition process.

An object is placed in the hand of the robot and during the recognition process it is rotated to produce new views. The object manipulation sequence is *not* predefined and will be different for every object. For a given, unknown object, it is most efficient to look for a view that yields the most additional information for discriminating it from similar ones. Hence, based on the current view, the associated object probabilities, and the history of actions, a motion is selected that is expected to yield the highest information gain. In our case, actions consist of the rotation of an object in an object-centered coordinate system and involve the whole kinematic chain from hand to eye. The actions are executed using a learned inverse kinematics in a 15 degree-of-freedom space.

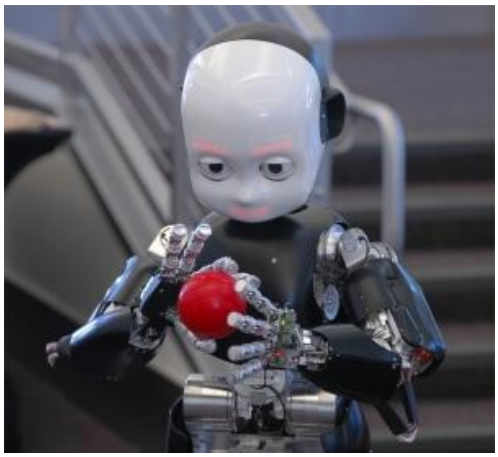

Fig. 1: The iCub humanoid robot. Implementation and evaluation platform of the presented object recognition method.

Hypotheses about the object in question are created and updated as the recognition progresses. We define a hypothesis as an estimate about an object and the viewpoint onto this object, which gives rise to a specific 2D view. A viewpoint is defined as a location on a view sphere centered around the object. We adopt probabilistic Monte Carlo localization methods to maintain a high number of hypotheses in parallel. By running particle filtering, regarding hypotheses as particles, we can take into account the viewpoint changes in the form of proprioceptive information obtained form the robot arm. Object probabilities are calculated based on these hypotheses and an action is selected which is expected to minimize the uncertainty of the current estimate.

II. RELATED WORK

Our work shares the philosophy of the active vision paradigm. It has been shown [1], [2], [3] that by enabling an observer to actively control the sensory input, many vision ambiguities can be resolved. This fundamental idea has soon been adopted by the robotics community and led to a number of systems that implement active object exploration and recognition methods. First approaches tried to move the camera to new locations that yield informative object views [20]. For example, Paletta and Pinz [12] propose a vision-only, camera-based system that is in spirit similar to ours. In

Parts of this research were supported by the European Project POETICON (grant number 215843) and the WCU (World Class University) program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (R31-2008-000-10008-0).

B. Browatzki and H. H. Bülthoff are with the Department of Human Perception, Cognition and Action, Max Planck Institute for Biological Cybernetics, Tübingen, Germany. bjoern.browatzki, heinrich.buelthoff@tuebingen.mpg.de

V. Tikhanoff and G. Metta are with the Department for Robotics, Brain and Cognitive Sciences, Italian Institute of Technology, Genova, Italy. vadim.tikhanoff, giorgio.metta@iit.it

C. Wallraven is with the Department of Brain and Cognitive Engineering, Korea University, Seoul, Korea. wallraven@korea.ac.kr

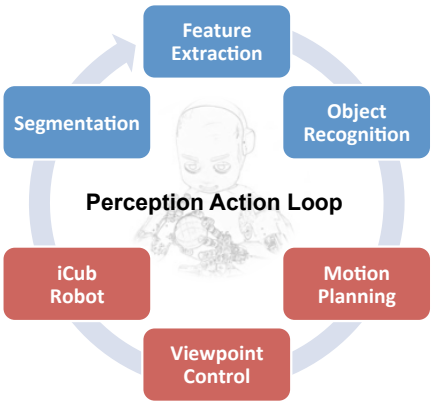

Fig. 2: System components forming a perception action loop.

this work, a recognition sequence is searched that minimizes the number observations needed to achieve a confident object hypothesis. In doing so, the object is placed statically on a turn table and the camera is moved around it on a view sphere.

In [9] the robot acquires information on objects in its vicinity by using its body to explore and probe the environment. Omrčen et al. [11] address the basic sensorimotor processes that have to be provided to allow a dexterous exploration of an unknown object. In [19] and especially [18] active methods have been studied that are comparable to our approach in the sense that objects are inspected from multiple view points to resolve ambiguities and to distinguish between objects. However, these systems lack the direct interaction with the object and are therefore not able to take into account the additional cues active exploration on a humanoid robot offers. The benefit of incorporating proprioceptive cues, for example, is demonstrated in Sec. VI.

III. Perception Driven Object Recognition

The recognition system we propose can be split into two fundamental parts. One part is the perception side which comprises components for data acquisition and reasoning and the other part is the action side that consists of the robot hardware and controllers. We link perception and action directly together and one is determined by the outcome of the other. This principle is often referred to as Perception-Action-Loop and is shown in Figure 2. In the following we discuss some of the components that need to be provided in order to close this loop. Subsequently the object learning and recognition modules will be presented in Sec. IV and V.

A. The iCub robot platform

The iCub (Fig. 1) is an open-source humanoid robot designed as a result of the RobotCub project, a collaborative European project aiming at developing a new open-source cognitive robotics platform. Measuring 105cm in total height, the iCub robot is approximately the same size as a three year old child. The iCub is the ideal platform to undertake research in cognitive systems [7], [14], [15], as it has fully articulated hands, which allow for dexterous manipulations, as well as a head-and-eye system, which permits very precise and accurate movements that are required for vision. Furthermore, the iCub robotic platform is equipped with visual, vestibular (for balance and spatial orientation), auditory, and haptic sensor capabilities.

B. Viewpoint Control

To explore an object systematically one needs to be able to describe already seen viewpoints and desired viewpoints in an efficient and compact way. However, the joint space of a humanoid robot that needs to be controlled in order to achieve a desired viewpoint is usually high dimensional. In our case, the whole kinematic chain that can be used to manipulate an object within the field of view consists of 15 degrees of freedom (DOFs): 7 DOFs for the arm and wrist, 3 DOFs for the torso, and 5 DOFs for head and eye [13]. In principle, one could fix some of these joints and, for example, only move the wrist while keeping a steady gaze onto the hand. However, due to motor constraints the space of accessible viewpoints would be very limited. In our approach we therefore incorporated all possible DOFs to increase the range of motion as much as possible and to ensure that we can generate a high number of object views without re-grasping.

We describe a viewpoint in terms of two parameters, azimuth φ and elevation θ. They are defined in respect to the reference frame N of the robot hand and the gaze vector G between hand and eye as

$$\theta = acos(G \cdot N_z), \tag{1}$$

$$\varphi = acos(G \cdot N'_x). \tag{2}$$

N'_x denotes the vector N_x after tilting the plane $N_x \times N_y$ to be orthogonal to G. To improve readability, in the following we will refer to (θ, φ) as a viewpoint ϕ.

We employ a linear-weighted nearest neighbor search to map from viewpoint ϕ to joint states $\mathbf{q}$. Sample points for the search are collected through random movement. Each sample consist of the current gaze angles and the joint angles of the DOFs we are interested in.

To obtain a configuration in joint space that leads to a desired viewpoint we search through the recorded samples and calculate the weighted joint average

$$\hat{\mathbf{q}} = \sum_{i}^{N} W_i \mathbf{q}_i, \tag{3}$$

from the N nearest neighbors according the sample weights W_i given by:

$$W_i = w_\Phi \Phi \left\langle \phi_i, \phi \right\rangle + w_X \left\| \mathbf{x}_i - \mathbf{x} \right\| + w_q \left\| \mathbf{q}_i - \mathbf{q} \right\|, \tag{4}$$

where $\Phi \left\langle \cdot \right\rangle$ denotes the angle between two gaze vectors. The second and third terms are optional and comprise additional optimization tasks. The middle term optimizes for a low euclidian distance to a desired location in 3D space. The last term penalizes large changes in joint space to reduce jerkiness. These optimization goals are assigned individual weights w_Φ, w_X, and w_q.

The calculations are simple and run at real-time performance

even for a large number of samples. To speed up computation even further we select a subset of unweighted samples solely based on viewpoint ϕ using an approximate nearest neighbor search to only take these samples into account for weighting. The resulting poses $\hat{\mathbf{q}}$ produce viewpoints with a deviation of usually less than 3° from the desired viewpoint. The accuracy is sufficient since viewpoint changes below this margin are not expected to cause relevant changes in the image.

C. Image segmentation and feature extraction

The object in the hand of the robot occupies only a small area of the image obtained from the robot camera. Using the whole image for recognition would result in a poor recognition performance, as the scene background would introduce a high amount of noise. However, since the gaze is permanently updated to keep the object in focus, we can safely assume that the object is always located approximately in the center of image. An obvious improvement therefore presents itself in cropping a sub image around the image center. To remove background area from the image we train a Gaussian Mixture Model (GMM) (see e.g. [4]) on a small region around the selected sub image. GMMs are commonly used for background removal tasks. They are specified by K normal distributions $\mathcal{N}_k(\mu, \Sigma)$ with mean μ and covariance Σ as well as a weight w_k. In contrast to most background removal approaches, we do not create a background model for each image location. Since the camera is moved we need to be able to deal with a quickly changing background. Therefore we create a model on each incoming image and directly apply it to the current view.

In Fig. 3 the training area is depicted. We take pixel intensities in CIE L*a*b* color space as input samples for calculating $\mathcal{N}_{1,..,K}(\mu, \Sigma)$. The optimization is carried out using Expectation-Maximization (we rely on the C++ implementation from the OpenCV library [6]). As we only use a low number of pixels, training is completed within approximately 50ms on a dual-core 2.5 GHz mobile CPU. The trained GMM is then applied to the sub image containing the object. The probability P_{BGR} of an image pixel I being considered as background is computed by evaluating the weighted PDF of the multivariate distribution defined by:

$$P_{BGR}(I) = \sum_{k=1}^{K} \frac{w_k}{\sqrt{|2\pi\Sigma_k|}} \cdot \exp^{\frac{1}{2}(I-\mu)^T \Sigma_k^{-1}(I-\mu)} \quad (5)$$

The resulting probability map is then scaled to the interval $[0, 1]$. This is done because we assume that there is always an object present in the image. Hence, some part of the image always should to be classified as object. We consider all image locations with background probability smaller than a fixed threshold as occupied by the object. Finally, morphological operations are applied to remove clutter and artifacts.

To describe the image content we extract two types of features from the segmented object images. We rely on the well known Pyramids of Histograms of Oriented Gradients feature (PHOG) [5] (2 levels, 20 orient. bins) to capture information on object shape and use color histograms in CIE L*a*b* space (15x15 bins for a* and b*) as a basic appearance descriptor.

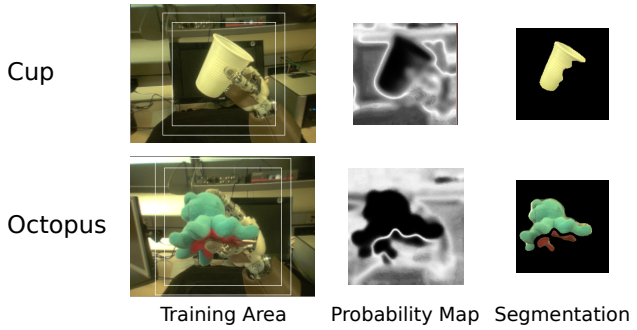

Fig. 3: Segmentation process: A background model is trained on the area between the rectangles and applied to the area inside the smaller rectangle. Low values in the resulting probability map indicate the presence of an object.

IV. Object exploration and learning

For view-based three-dimensional object recognition, sample views need to be acquired from various viewpoints. These collections of views are then converted into representations of objects or object classes. It is desirable that the acquisition of these views is performed in an efficient way in terms of exploration time and motor actions. We address this point by keeping track of already seen viewpoints and selecting target positions that minimize

$$\phi_{t+1} = argmin_{\phi'} e(\phi', \phi_t) \quad (6)$$

with

$$e(\phi', \phi_t) = \alpha\tau(\phi') \cdot (1-\alpha)\Phi\left\langle \phi', \phi_t \right\rangle. \quad (7)$$

Eq. (7) results in low values for viewpoints that are close to the current viewpoint but add as much new information as possible. The parameter α controls how fast the object is moved to new orientations. Low values result in a slow motion in which the object is examined carefully. High values, in contrast, lead to a coarse but fast exploration. The function $\tau(\cdot)$ in Eq. (7) defines how well a certain view has been seen before and is defined as

$$\tau(\phi) = \max_{v \in V} 1 - \frac{\Phi\left\langle \phi, \phi_v \right\rangle}{fov}, \quad (8)$$

with fov denoting the angle of the field-of-view and V being all previously visited positions on the view sphere. We obtain values in the interval [0,1] with 0 for a completely unknown view and 1 for an exact viewpoint match.

Recording the whole input stream without filtering would result in a vast amount of data that is intractable, both in terms of memory consumption and computational effort. However, keeping too few views would lead to a low recognition performance. To select only a representative subset of views from the continuous stream of input images we extract certain keyframes [17] based on the amount of visual change occurring. We measure the change by converting the image to a lower dimensional feature representation and

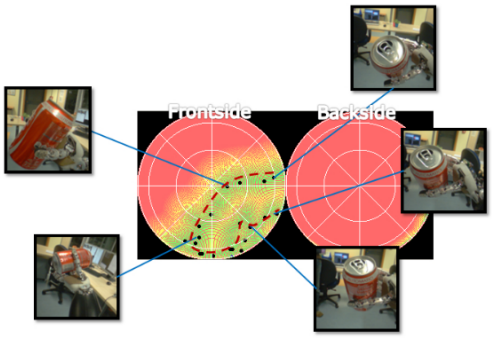

Fig. 4: Example of an exploration sequence executed to learn a new object. The dashed red line shows the exploration path on the view sphere. Keyframes are marked by black dots.

then by computing the feature distance between subsequent images. This representation should be insensitive to clutter and artifacts introduced through preprocessing steps such as segmentation. We obtained good results by converting the images to CIE L*a*b* color space and selecting high energy frequencies of the Fourier-transformed color planes. A frame is marked as keyframe if the feature distance to the last keyframe drops below a predefined threshold. Each keyframe is annotated with the viewpoint is was extracted on. A few dozens of these keyframes are usually sufficient to represent the appearance of an object.

V. Active Object Recognition

Active recognition in the context of a humanoid robot differs from static (multi-view) object recognition scenarios by offering two valuable benefits. First, the object in question can be manipulated. Thus additional information can be generated by creating new views. Second, by incorporating proprioceptive information from joint states we do not have to rely on a unrelated set of views but can also take into account the viewpoint differences that caused the change in appearance.

Each view can be treated as an observation that adds information about object probabilities. Sequences of observations can be combined to form joint distributions of object probabilities. However the questions arises of how to calculate object probabilities and how to select actions that lead to short sequences with discriminative views. This problem is addressed in the following.

A. Recognition by localization

Object recognition in this context can be regarded as a localization problem in which the goal is to find the most probable location on the view spheres of the objects. We define such a location as $x = (o, \phi)$, with o determining the object and ϕ defining the view angle. We estimate the probability distribution over all positions x at every recognition iteration t. We can calculate the posterior probability of x given a sequence of actions a and observations z as follows:

$$\begin{aligned} p(x_t) &= p(x_t|z_{1:t}, a_{1:t}) \\ &= p(z_t|x_t) \cdot \int p(x_t|x_{t-1}, a_t) \cdot p(x_{t-1}) dx_{t-1}. \end{aligned} \quad (9)$$

Equation (9) defines the posterior of a recursive Bayesian filter. As the exact solution is intractable, we employ a particle filter as a Monte Carlo approximation. The particle filter achieves a discrete estimation of the true distribution using a large set of weighted samples, or particles $\{x^i, w^i\}_{i=1}^N$. The approximated posterior can be stated as

$$p(x_t) \approx p(z_t|x_t) \sum_{i=1}^{N} w_t^i \cdot p(x_t|x_{t-1}^i, a_t). \quad (10)$$

Our particle filter implementation is based on the bootstrap filter proposed in [10] and outlined in Alg. 1. We predict new particle positions by taking into account the viewpoint change from the last observation. It is important to note that this information is acquired through proprioception by the active robot. After adding Gaussian noise $\mathcal{N}(\mu, \Sigma)$, the particle update rule is given by

$$x_t \leftarrow q(x_t^i|x_{t-1}^i, a_t) = x_{t-1}^{i,\phi} + a_t + \mathcal{N}(\mu, \Sigma). \quad (11)$$

New particle weights are computed based on the expected view at a certain location on the view sphere. This view is estimated from the extracted keyframes (see Sec. IV). To obtain an estimate on all possible particle positions we interpolate keyframes on points that were not directly observed. The interpolation is done by calculating a weighted average from the k-nearest neighbors. The weight is set proportional to the angular distance to the neighboring keyframes. The interpolation needs to be done only once per object and can be precomputed offline. Using the resulting maps of view estimates $V_{est}[\cdot]_{k=1}^K$ as lookups, the likelihood of an observation with feature vector z given a particle x^i can efficiently be calculated by

$$p(z|x_t^i) = (V_{est}[x_\phi]_{x_o} - z)^2, \quad (12)$$

and the particle weight updated by

$$w_t^i = w_{t-1}^i \cdot p(z|x_t^i). \quad (13)$$

Since the view sphere is limited due to motor constraints, we need to decide on how to handle particles that leave the part for which we have gathered data. We set these particles to low weights but do not discard them entirely. It is always possible that the object is located differently in the hand of robot and we currently observe views that were not visible during training. Assigning lower weights, however, is necessary since otherwise we would maintain too many hypotheses that will not be updated in the future. Low weights will eventually result in a higher probability of being discarded during resampling.

To remove particles with low weights we resample the particles in each iteration. We follow the sampling and importance resampling (SIR) procedure [10] and replace each particle with another particle that is picked proportional

to its likelihood of giving rise to the current observation (Eq. 12). We found that it is important to base the *resampling* only on the current view instead of taking into account past iterations. This way it is avoided that, in the absence of discriminative input, the filter converges to an arbitrary mode. After resampling, the object probabilities $P_{1,..,K}$ are calculated by integrating the weights of the particles associated with object k,

$$P_t^k = \frac{\sum_{i=1}^{N} \delta_{k,\pi_i} w_t^i}{\sum_{i=1}^{N} w_t^i}. \tag{14}$$

In Eq. (14) $\delta_{i,j}$ refers to the Kronecker delta returning 1 for $i = j$ and 0 otherwise. We can now calculate the entropy H_t of the current object probability distribution,

$$H_t = \sum_{k=1}^{K} P_t^k log P_t^k. \tag{15}$$

H_t describes the uncertainty of our current predictions and serves as an indicator of when a confident assumption can be made. We furthermore use H_t as optimization target for determining the next action as described in the following subsection.

Algorithm 1 Particle filter object recognition

for $k = 1 \rightarrow K$ **do** ▷ Initialization
  for $n = 1 \rightarrow N/K$ **do**
    $i \leftarrow k \cdot n + n$, $\pi_i \leftarrow k$
    Draw particle x_i randomly from view sphere of object k.
    Assign initial weights:
      $w_0^i = g(y_0|x_0^i)$
  end for
end for
$t \leftarrow 0$
repeat ▷ Iteration
  $t \leftarrow t + 1$
  Execute action a_t.
  Aquire observation z_t.
  Update particles:
  for $i = 1 \rightarrow N$ **do**
    $x_t^i \leftarrow q(x_t^i|x_{t-1}^i, a_t)$
    $w_t^{'i} \leftarrow w_{t-1}^i p(z_t|\tilde{x}_t^i)$
  end for
  Normalize weights:
    $\tilde{w}_t^i \leftarrow \tilde{w}_{t-1}^{'i} / \sum_{j=1}^{N} \tilde{w}_t^{'j}, \qquad i = 1 \rightarrow N$
  Resample particles:
    Select N particle indices $j_i = 1 \rightarrow N$ proportional to particle likelihood of current observation:
      $j_i \leftarrow l \propto \frac{p(z_t|\tilde{x}_t^l)}{\sum_{j=1}^{N} p(z_t^j|\tilde{x}_t^j)}, \qquad l = 1 \rightarrow N$
    $x_t^i \leftarrow \tilde{x}_t^{j_i}, w_t^i \leftarrow \tilde{w}_t^{j_i}$
  Calculate object probabilities:
    $P_t^k = \sum_{i=1}^{N} w_t^i \delta_{k,\pi_i} / \sum_{i=1}^{N} w_t^i, \quad k = 1 \rightarrow K$
  Calculate entropy:
    $H_t = \sum_{k=1} P_t^k log P_t^k$
until $H_t < H_{des}$

B. Action selection

We are looking for an action a that is expected to lead to a viewpoint that maximizes the expected information gain. This can be formulated as

$$a = \arg\max_{\tilde{a} \in A} E[I_{\tilde{a}}] = H_t - E[H_{t+1}^{\tilde{a}}], \tag{16}$$

where I_a denotes the information gain by executing action $\tilde{a}$. H_t is the current entropy across object probabilities. The entropy $H(X) = -\sum_{x \in X} x \log x$ of a random variable X can be utilized to indicate the information content of our current estimate. To maximize the information gain we need to find an action that minimizes the expected entropy. The expected entropy can be calculated by integrating over the range of expected observations m produced by action $\tilde{a}$

$$E[H_{t+1}^{\tilde{a}}] = \int_m H(P_t(m)) dm. \tag{17}$$

We approximate Eq. (17) by sampling observations $z_{1,..,M}$ at view sphere locations $\tilde{x} = g(\tilde{x}|x_t, \tilde{a})$ from our view estimation map V_{est}. Consequently $E[H_{t+1}(\tilde{a})]$ can be stated as

$$E[H_{t+1}(\tilde{a})] \approx -\sum_{m}^{M} H(P_t(m)) \frac{p(z_m|\tilde{a})}{\sum_m p(z_m|\tilde{a})}. \tag{18}$$

Object probabilities $P_t(m)$ defined previously in Eq. (14) are dependent on the set of particles $\tilde{x}$ propagated by action $\tilde{a}$ and the sampled view z_m:

$$P_t^k(m) = \frac{\sum_i^N \delta_{k,\pi_i} w_i p(z_m|\tilde{x}_i)}{\sum_i^N w_i p(z_m|\tilde{x}_i)}. \tag{19}$$

We assume an observation z_m to occur with a probability only dependent on the position on the view sphere and not being dependent on the action performed. Thus we set $p(z_m|\tilde{a}) = 1/M$ for all z_m. This, however, does not imply that actions do not affect the expected observations. The sampling position of the observation still depends on the action and its start position. Finally, after inserting Eq. (19) into Eq. (17) we obtain

$$E[H_{t+1}(\tilde{a})] \approx -\frac{1}{M} \sum_{m}^{M} \sum_{k}^{K} \frac{\sum_i^N \delta_{k,\pi_i} w_i p(z_m|\tilde{x}_i)}{\sum_i^N w_i p(z_m|\tilde{x}_i)} \cdot \log \frac{\sum_i^N \delta_{k,\pi_i} w_i p(z_m|\tilde{x}_i)}{\sum_i^N w_i p(z_m|\tilde{x}_i)} \tag{20}$$

Unfortunately, the evaluation of Eq. (20) is time-consuming. We do, however, not necessarily need to know the exact entropy values to find a favorable action. Instead, we follow the reasoning in [8] and search for an action a that leads to views that introduce a high amount of variance across different objects

$$a = \arg\max_{\tilde{a} \in A} D_{\tilde{a}} \approx E[D_{\tilde{a}}] = \sum_{i}^{\hat{N}} \sum_{j}^{\hat{N}} (\tilde{z}_i - \tilde{z}_j) \cdot \kappa_{i,j} \tag{21}$$

with

$$\kappa_{i,j} = \begin{cases} \alpha & \text{if } i = j \\ \beta & \text{else} \end{cases} \tag{22}$$

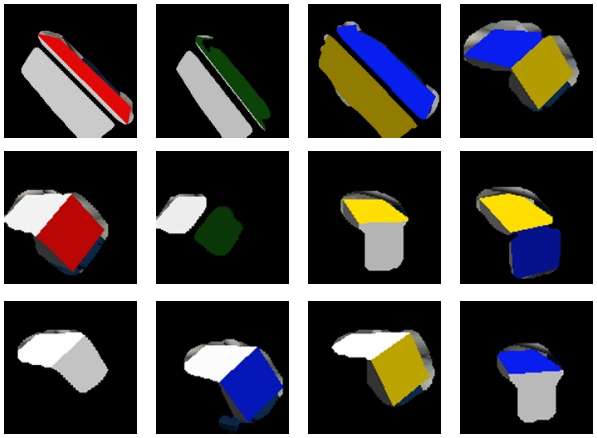

Fig. 5: Objects used for evaluation in the iCub simulator. 12 rectangular boxes with differently colored sides. Shown are segmented keyframes used in the recognition process.

Fig. 6: Objects used for evaluation on the iCub. Six identical brown plastic cups, of which five are modified with colored tape.

For $\alpha = 1$ and $\beta = 0$ we only take into account views that lie on different objects. By modifying κ, however, we can steer the algorithm towards discriminating between views within the same objects. This is useful when the task consists less of identifying objects but rather of determining the specific object pose. However, even when the object pose is not of interest it is beneficial to set β to a positive value. When the recognition is approaching saturation, most particles will then be located on one object.

VI. EVALUATION

As the actual grasping is not part of this study, we start all experiments with the object already grasped and located in the hand of the robot. To demonstrate the ability of our system to distinguish even between highly similar objects, we conducted evaluations in simulation and on the real robot on objects that share many views. The objects used in the experiments are depicted in Fig. 5 and Fig. 6 respectively. Objects were learned as described in Sec. IV. Since looking at the back of the hand is hardly desirable we restricted the exploration to the upper hemisphere. Furthermore, as noted earlier, the robot is not able to bring the hand into all orientations a human is able to do. For example, it is not possible to have the robot look directly onto its fingertips from the front. We therefore limited the accessible area of the view sphere for exploration as well as for recognition to $[0°, 90°]$ elevation and $[140°, 270°]$ azimuth. During the exploration phase, between 80 and 130 keyframes were recorded per object in the automatic exploration mode.

For both experiments we conducted 20 recognition trials for each object. 10 trials with planned actions using the variance maximization scheme discussed in Sec. V and 10 trials with viewpoints randomly selected on the previously defined part of the view sphere. We should note that picking random positions on the view sphere does not lead to views that are as random as setting random joint angles within a certain range. By selecting view sphere positions, the same poses will be available that were potentially obtained during learning and can be set by the motion planning algorithm. Using completely random joint or task space positions would make object recognition even more difficult and would not represent a viable baseline. After each trial, a random position was set to ensure independence between individual trials.

The recognition process was stopped when the entropy dropped below 0.1 or 15 iterations were completed. The particle filter was initialized with 75 particles per object—uniformly sampled on the view sphere of each object. Motion planning was performed taking into account the 300 highest weighted particles. We distinguish actions up to a resolution of $\pm 1°$ on the view sphere and computation was speeded up using a grid search on a recursive raster of 9x14 elevation and azimuth cells. The total planning time was approximately 400ms per iteration.

To compare our approach against pure visual recognition, not taking into account proprioceptive information, we performed k-nearest neighbor matchings with the recorded keyframes. This was done for all trials, in parallel to the particle filtering. We performed matchings using 1,3 and 5 neighbors. As the differences are marginal we only plot results for 3 nearest neighbors.

A. Evaluation in simulation

To test the validity of our approach under controlled conditions, we conducted an experiment on the iCub simulator [16]. We created 12 rectangular boxes (Fig. 5) using common 3D modeling tools and assigned different colors to three of the sides. Thus, all objects contain sides that are identical to sides of other objects. This means that single views can easily be confused. As the objects, however, are distinguishable from other viewpoints, we expect that after some exploration time enough information is gathered to correctly identify most of them. Results are shown in Fig. 7 and Fig. 8. We see that planned motion leads to more correct results and more confident predictions. The latter is indicated by the much faster decreasing entropy in Fig. 8. These findings could be confirmed by the following real-world experiment on the iCub robot.

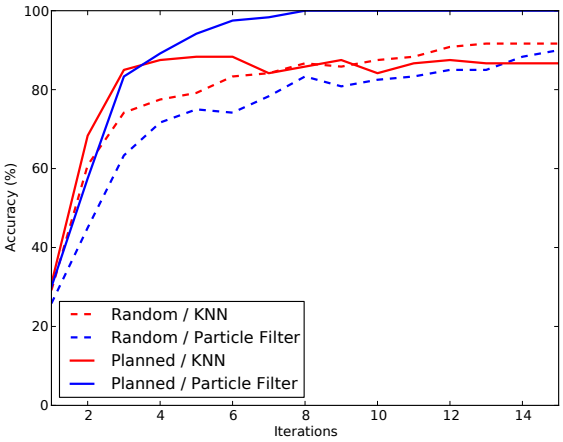

Fig. 7: Iteration accuracy in simulation.

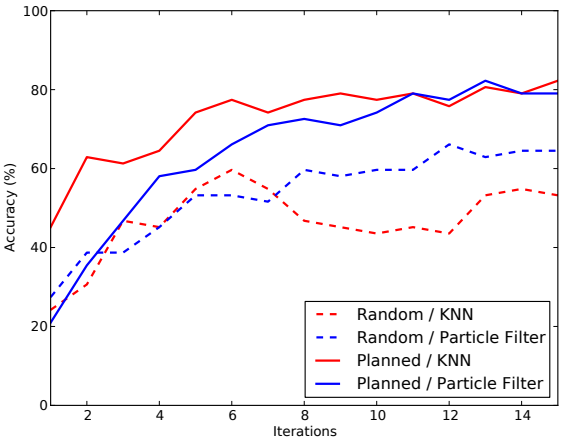

Fig. 9: Iteration accuracy using the iCub.

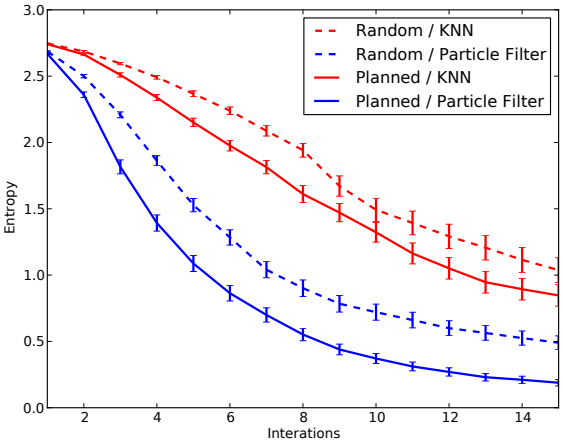

Fig. 8: Iteration entropy in simulation.

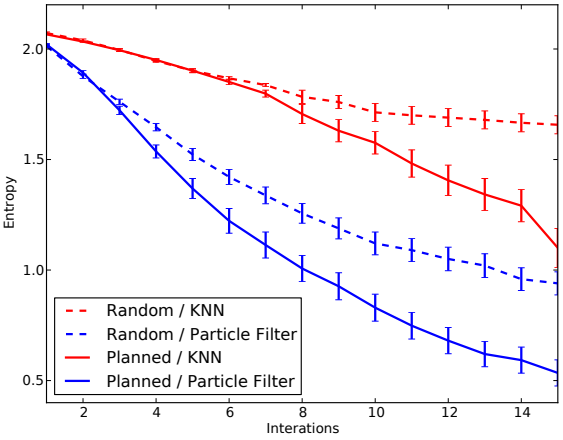

Fig. 10: Iteration entropy using the iCub.

B. *Real-world evaluation*

For the experiment on the iCub we chose six brown plastic cups with five of them marked at one location with colored tape. This way they were distinguishable from the other cups only from a very limited set of viewpoints. For two of the cups, for example, we placed the marker inside the cups. These cups can only be recognized by looking directly inside—a challenging task when the movement is not pre-programmed.

In Fig. 9 the recognition accuracy for all 6 objects is plotted for the 15 iterations. We clearly see that planned motion results in much higher recognition rates than those achieved by random exploration. Explicitly taking into account proprioception yields an increase in accuracy only in the case of random motion. This makes sense as the planned exploration often does not perform much motion at all; for many objects, an optimal viewpoint is achieved already after a few iterations, and this view is maintained until recognition is completed. We see, however, the benefit of the particle filter that considers proprioceptive information when we look at Fig. 10. Since accuracy does not tell us anything about *the speed and the confidence* with which a result was obtained, we also plot the entropy of object probabilities for all iterations. Especially in the case of Fig. 10, it is clearly visible how these two features are significantly improved by going beyond visual comparisons only.

If we look more closely at the results of individual objects, we find that for particular objects the differences are substantial. In Fig. 11 the accuracy for each object after 10 iterations is shown. The unmarked brown cup ('Normal cup'), for example, was recognized correctly only one time using random, vision only exploration (chance level = 16.7%). We found this effect also in the simulation experiments for the object (white box) that bore no apparent mark with which it could be identified. From the way, the experiment is set up, it becomes clear that unmarked objects can only be recognized by *rejecting alternatives*. This, however, can only be achieved by reasoning across multiple observations and regarding subsequent views in relation to each other.

Furthermore, it is very interesting to investigate which viewpoints were visited during the recognition of individual objects. In Fig. 12 we marked the view sphere positions for each object during the 10 planned recognition trials. Dots are scaled according to iteration number with the first iterations represented by the smallest dots. We see that the distribution is different for all objects. The upper (left) area in the plots corresponds to positions in which a side view of the object is obtained (here, for example, the side stickers could be seen) the bottom righthand area contains the viewpoints that allow to view into the cups. The region directly above indicates

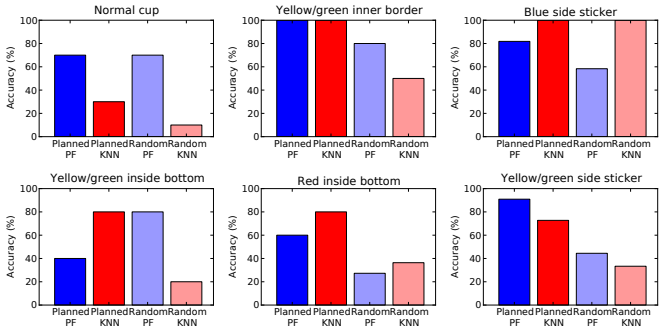

Fig. 11: Recognition performances for the six plastic cups after ten iterations.

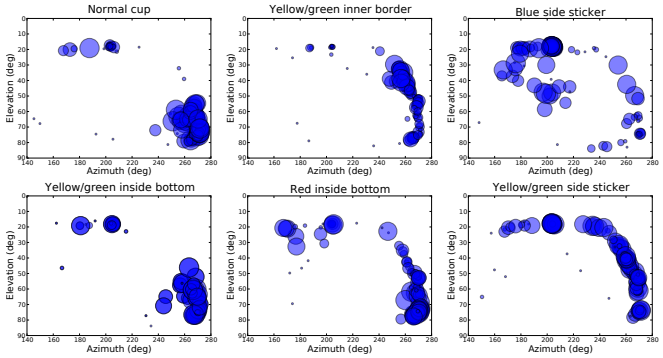

Fig. 12: Observers viewpoints during 10 recognition trials.

poses in which the bottom of cup could no longer be seen, but the inside rim was well visible. The patterns look similar for the normal cup and for the two cups with modified inside bottoms. This means that these objects were mostly inspected by looking inside. In contrast, for the cup with the blue sticker on the side, the robot only looked inside a few times during the first iterations (indicated by small dots). After rejecting the according hypotheses it concentrated on observing the cup from viewpoints that quickly put the blue sticker into focus.

VII. CONCLUSIONS

In this work we have implemented a perception-driven object recognition process that allows a humanoid robot to recognize even highly similar objects by actively resolving ambiguities. We have demonstrated in simulation and in real-life experiments that by predicting optimal viewpoints objects can be identified much faster and more reliably. In addition, we have shown that some difficult objects can only be recognized by rejecting all possible alternate hypothesis—again, this would not be achievable without our active viewpoint planning. Finally, the incorporation of proprioceptive information (that is, that we can work in the joint angle space of the robot), also resulted in a significant improvement over visual-only comparisons by speeding up the recognition process considerably. In future work, we will optimize this framework to deal with a large number of concurrent hypotheses—after all, in typical use cases, the robot will be expected to deal with a large number of objects (and object categories). Finally, it might be possible to extend our framework to a more abstract search space, in which objects are disambiguated not by viewpoints but by performing certain actions (such as taking an object, and trying to fit it into one of several differently-shaped slots). Perception-action skills like this will allow the robot to become an active explorer and to learn about its environment by interacting with it—similarly to the stages in infant development.

VIII. ACKNOWLEDGMENTS

The authors wish to acknowledge Lorenzo Natale for his contributions to this work.

REFERENCES

[1] J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. *Intl Journal of Computer Vision*, 1:333–356, 1988. 10.1007/BF00133571.

[2] R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, May 1988.

[3] D. Ballard. Animate vision. *Artificial intelligence*, 48(1):57–86, 1991.

[4] J. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. *International Computer Science Institute*, 4(510):126, 1998.

[5] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *International conference on image and video retrieval*, pages 401–408. ACM, 2007.

[6] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[7] F. Broz, H. Kose-Bagci, C. Nehaniv, and K. Dautenhahn. Learning behavior for a social interaction game with a childlike humanoid robot. In *Social Learning in Interactive Scenarios Workshop, Humanoids*, 2009.

[8] N. Fairfield and D. Wettergreen. Active localization on the ocean floor with multibeam sonar. In *OCEANS 2008*, pages 1–10. IEEE, 2008.

[9] P. Fitzpatrick and G. Metta. Grounding vision through experimental manipulation. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 2003.

[10] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, 1993.

[11] D. Omrčen, A. Ude, K. Welke, T. Asfour, and R. Dillmann. Sensorimotor processes for learning object representations. In *Humanoid Robots*, pages 143–150. IEEE, 2007.

[12] L. Paletta and A. Pinz. Active object recognition by view integration and reinforcement learning. *IEEE International Conference on Robotics and Autonomous Systems*, 2000.

[13] U. Pattacini, F. Nori, L. Natale, G. Metta, and G. Sandini. An Experimental Evaluation of a Novel Minimum-Jerk Cartesian Controller for Humanoid Robots. *Dynamical Systems*, pages 1668–1674, 2010.

[14] G. Sandini, G. Metta, and D. Vernon. The iCub cognitive humanoid robot: An open-system research platform for enactive cognition. *50 years of artificial intelligence*, pages 358–369, 2007.

[15] V. Tikhanoff, A. Cangelosi, and G. Metta. Integration of Speech and Action in Humanoid Robots: iCub Simulation Experiments. *Autonomous Mental Development*, 3(1):17–29, 2011.

[16] V. Tikhanoff, P. Fitzpatrick, F. Nori, L. Natale, G. Metta, and A. Cangelosi. The iCub Humanoid Robot Simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, 2008.

[17] C. Wallraven and H. H. Bülthoff. *Object Recognition, Attention, and Action*, chapter Object Recognition in Man and Machine, pages 89–104. Springer, Tokyo, Japan, 2007.

[18] K. Welke, T. Asfour, and R. Dillmann. Object separation using active methods and multi-view representations. In *IEEE International Conference on Robotics and Automation*, pages 949–955, 2008.

[19] K. Welke, T. Asfour, and R. Dillmann. Active multi-view object search on a humanoid head. In *IEEE International Conference on Robotics and Automation*, pages 417–423. IEEE, 2009.

[20] D. Wilkes and J. Tsotsos. Active object recognition. In *Computer Vision and Pattern Recognition*, pages 136–141. IEEE, 1992.