Maximally Informative Interaction Learning for Scene Exploration

Herke van Hoof[1], Oliver Kroemer[1], Heni Ben Amor[1] and Jan Peters[1,2]
{hoof,kroemer,amor,peters}@ias.tu-darmstadt.de

***Abstract*— Creating robots that can act autonomously in dynamic, unstructured environments is a major challenge. In such environments, learning to recognize and manipulate novel objects is an important capability. A truly autonomous robot acquires knowledge through interaction with its environment without using heuristics or prior information encoding human domain insights. Static images often provide insufficient information for inferring the relevant properties of the objects in a scene. Hence, a robot needs to explore these objects by interacting with them. However, there may be many exploratory actions possible, and a large portion of these actions may be non-informative. To learn quickly and efficiently, a robot must select actions that are expected to have the most informative outcomes. In the proposed bottom-up approach, the robot achieves this goal by quantifying the expected informativeness of its own actions. We use this approach to segment a scene into its constituent objects as a first step in learning the properties and affordances of objects. Evaluations showed that the proposed information-theoretic approach allows a robot to efficiently infer the composite structure of its environment.**

I. Introduction

Recognizing and manipulating objects is an essential capability for many robots. Today's industrial robots operate in structured environments with accurate object knowledge provided by human engineers. However, robots in unstructured environments will frequently encounter new objects. Hence, pre-defined object information does not always suffice. Instead, the robot should autonomously learn the properties of objects in its environment using machine learning techniques.

Supervised machine learning techniques, however, often require large amounts of manually annotated data. Furthermore, many techniques need a human expert to fine-tune parameters and features to a specific situation. Such top-down methods, which rely on prior training and human expertise, are usually not suitable for autonomous robots [1].

Alternatively, robots can autonomously collect new knowledge using interactive perception [2, 3]. This bottom-up approach couples perception to physical interactions, such as pushing, grasping, or lifting. Interactive perception allows a robot to learn, for example, the appearance and shape of objects [4–8], their haptic properties [9, 10], kinematic structure [2], and how the state of those objects changes as a result of manipulation [4, 10–12].

For such bottom-up approaches, selection of efficient actions is a challenge. In real world domains, most robots can execute a large variety of actions. Therefore, exploring a scene by executing all possible actions is usually infeasible. Instead, the robot should choose actions that are expected to reveal the most information about the world [9, 13, 14].

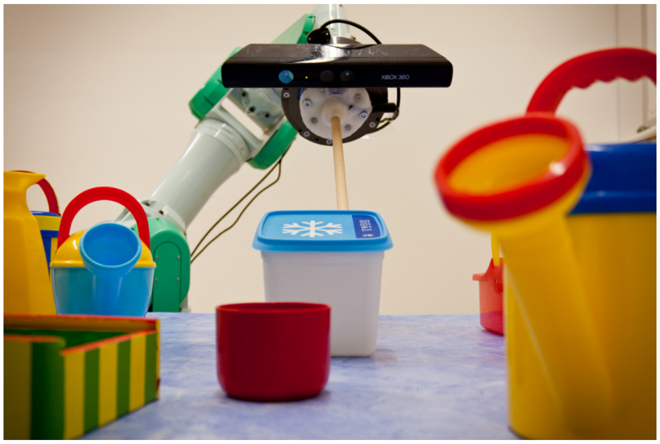


Fig. 1. Our robot learns the structure of its environment by observing the effects of its actions. Based on its observations so far, the robot selects the exploratory actions that are expected to yield maximally informative outcomes, using information-theoretic principles.

In this paper, we propose using a bottom-up approach that requires little prior knowledge to explore novel environments. In our approach, the robot uses its actions to elicit the information required for the specific situations that it encounters, as illustrated in Fig. 1. Rather than finding explorative actions using heuristics, we propose an autonomous system that selects maximally informative actions in a principled manner. The expected informativeness of actions is quantified using the information-theoretic measure of information gain.

We focus on segmenting the perceptual scene into separate objects. Only after segmentation can the robot explore the properties and affordances of individual objects [1, 15]. Methods for bottom-up object segmentation based solely on static images are limited by inherent ambiguities in the observations [3, 4]. For example, two differently-colored adjacent regions may actually be distinct parts of the same object, or belong to different objects. In our approach, we resolve such segmentation ambiguities by testing whether the regions are physically connected.

The knowledge gathered by our system is used to generate a graph-based representation of the observed scene, wherein the edges of the graph represent the probability of pairs of segments belonging to the same object. Our experiments show that the proposed approach efficiently discovers which groups of segments correspond to objects.

[1] TU Darmstadt, FB Informatik, FG IAS, Darmstadt, Germany
[2] Max Planck Institute for Intelligent Systems, Tübingen, Germany

II. Related work

Segmentation is an important step in finding and learning the properties and affordances of unknown objects in the robot's environment. Section II-A discusses how interaction can be used to obtain such segmentations. Subsequently, Section II-B discusses how to select maximally informative actions. The application of informative action selection in interactive settings is discussed in Section II-C.

A. Interactive scene segmentation

Fitzpatrick and Metta [4] did early work on using a robot's actions to segment visual scenes with unknown objects, by detecting the motion resulting from sweeping the arm across the workspace. Li and Kleeman [15] refined this method by using short, accurate pushes in cluttered environments. Kenney et al. [3] proposed to accumulate information over time to increase the accuracy of the segmentation.

The methods in the previous paragraph relied on image differencing to detect movement. Problems with this method, e.g. handling textureless objects, were addressed by Beale et al. [1], who started from a low-level over-segmentation of the image. Instead of estimating for every pixel whether it belongs to the object, object membership was estimated per segment. If visual features on the objects can be reliable tracked, these features can also be used to perform movement detection in the context of interactive segmentation [8, 16].

In all of those methods, the robot used actions that were either fixed, selected at random, or chosen by a heuristic. For a large part, they considered scenes containing only one object. However, if a robot has access to many possible exploratory actions, trying them at random is not efficient. Heuristics, on the other hand, rely on human insights and are likely to fail in unforeseen situations.

Other segmentation methods focused on settings where the robots held an object and used actions to segment it from the background and learn its properties [6, 7, 12, 17, 18]. These methods could further inspect objects after they have been segmented and, hence, these approaches would be a good complement to our work.

B. Selecting informative actions

Rather than using fixed actions or heuristics, the robot should adapt to the current situation by evaluating the informativeness of different actions. For instance, in the work of Denzler and Brown [13], the camera parameters that provide the largest mutual information are selected. Schneider et al. [9] proposed acquiring data with a tactile sensor at the height expected to produce the largest reduction in entropy. Hsiao et al. [19] minimized the expected costs of object pose estimation by selecting an optimal n-length sequence of actions. They found that usually a search depth of $n = 1$ is already sufficient. These active perception approaches change the perceptual parameters, but do not try to cause changes in the environment. Furthermore, these methods are usually applied after prior training.

C. Informed interaction

In contrast to the methods in the previous section, interactive perception approaches attempt to cause changes in the environment. By observing the effects of actions, object knowledge can be obtained without prior supervised training. This interactive approach also benefits from maximally informative actions. For instance, Krainin et al. [5] used a next-best-view algorithm based on information gain to select the best viewpoint and re-grasps for in-hand object modeling. Another way to estimate the informativeness of an action is by comparing the current situation to similar situations in the past, and selecting the action that was the most successful in uncovering new object properties in those situations. This approach was used by Katz et al. [14] to estimate the value of actions for exploring the kinematic structure of articulated objects using Q-learning.

In summary, informed action selection has been shown to outperform random actions in different active and interactive perception scenarios [9, 13, 14]. However, the discussed approaches for selecting informative actions did not address the problem of learning about completely novel objects. Most of these approaches required detailed models or training data of the objects before being able to efficiently discriminate between them [9, 13, 19]. Other approaches needed to already have sufficient knowledge to pick up the object [5] or required a human-crafted, domain-specific representation to generalize past experiences to the current situation [14].

III. Object segmentation using maximally informative interaction

The approaches for object learning and segmentation reviewed in Section II depend largely on prior training or human domain knowledge. In contrast, we propose to enable robots to develop such knowledge autonomously by interacting with novel objects. As a first step towards this goal, a robot needs to efficiently decompose the scene it perceives into its constituent objects.

The starting point of our segmentation approach is an over-segmentation of the visual scene into coherent segments. This process is described in Section III-A. Subsequently, the robot actively explores the scene to determine which segments constitute coherent objects. Efficient exploration is accomplished by selecting actions that are expected to maximize information gain. The action selection and execution methods are described in Sections III-B to III-F. The robot observes the resulting state of its environment as described in Section III-G. An overview of this process is given in Fig. 2, and the entire algorithm is summarized in Algorithm 1.

A. Finding object candidates using visual over-segmentation

As described in Section II-A, movement detection by image differencing has several disadvantages. Therefore, our approach uses visual over-segmentation to create segments containing pixels close together in Euclidean and color space, similar to the work of Beale et al. [1]. Subsequently, we observe whether movement has occurred for each segment, rather than for each pixel. Objects can consist of multiple

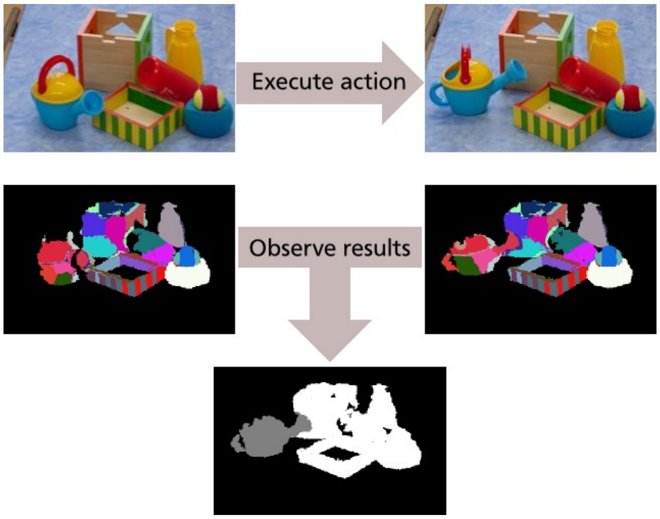


Fig. 2. Based on the results of previous actions, the action that is expected to yield the maximum information gain is chosen. This action results in some segments moving together, which increases the confidence that those segments form an object.

regions, each of which is coherent in space and color. Hence, we assume that a true segmentation of the objects in the scene can be obtained by merging segments.

The point clouds to be segmented are obtained using an RGBD camera mounted at the robot's end effector, as shown in Fig. 1. This set-up allows the robot to move the camera to different positions. Calibration of the camera allows merging of the point clouds and the removal of any points that do not correspond to objects on the table in front of the robot.

To obtain an over-segmentation, points are clustered according to six-dimensional feature vectors containing the color in CIELAB space [20] and the location in Euclidean space using the k-means algorithm. This is very similar to the SLIC algorithm [21], which operates in image space rather than on three-dimensional point clouds. Examples of segmentations are shown in Fig. 2. Segments are tracked over time by initializing the cluster centers in the current time step using the centers in the previous time step, similar to [22, 23].

B. Graph-based scene representation

We represent the over-segmented visual scene as a graph $G = (V, E)$ with V being the set of vertices representing the segments and E being the set of edges connecting every pair of vertices that belong to the same object. We consider every graph $G \in \mathcal{G}$, with $\mathcal{G}$ denoting the set of all graphs corresponding to partitionings of the set of all segments. For every pair of vertices (i, j), we define $g_{ij} = 1$ if the vertices belong to the same object ($\{i, j\} \in E$), and zero otherwise.

If a segment is pushed, all segments belonging to the same object as the pushed segment should move as well. The segments observed to move at time step t are represented by $\mathbf{o}^t$, with $o_j^t = [\mathbf{o}^t]_j = 1$ if vertex j was observed as moving, and zero otherwise.

When two segments are moving synchronously, the confidence that they belong to the same object increases. On the other hand, when only one of the segments moves, the confidence that they belong to different objects increases. The degree of confidence depends on the pushed vertex v. The data gathered about the edge between vertices i and j, up to time step t, will be denoted by $\mathcal{D}_{ij}^t = \begin{bmatrix} n_{ij}^t & N_{ij}^t & m_{ij}^t & M_{ij}^t \end{bmatrix}$ with n_{ij}^t the number of times both vertices moved out of the N_{ij}^t times that one of those vertices was pushed, and m_{ij}^t the number of times both vertices moved out of the M_{ij}^t times at least one of them moved but neither was pushed.

C. Finding informative actions for exploration

We want to find out which pairs of segments belong to the same objects, using as few actions as possible. Therefore, we want to find the vertex v to be pushed at time t that maximizes the expected information gain, that is given by the Kullback-Leibler divergence

$$\mathrm{KL}(\mathrm{P} \| Q) = \sum_i P(i) \ln \frac{P(i)}{Q(i)}$$

between the conditional distribution over graphs after the push $p(G|\mathcal{D}^{t+1})$ and before the push $p(G|\mathcal{D}^t)$.

In order to evaluate the expected information gain of every push, the probability distribution of a scene structure given observation data $p(G|\mathcal{D}^t)$ needs to be calculated. According to Bayes' rule, and assuming independence of observations given g_{ij}, the conditional probability $p(G|\mathcal{D}^t)$ is proportional to the joint probability

$$p(G, \mathcal{D}^t) = p(G) \prod_{i;j>i} p(\mathcal{D}_{ij}^t | g_{ij}), \tag{1}$$

where the product is over all pairs of vertices (i, j) with $j > i$. The probability $p(o_i = 1 | o_j = 1)$ depends on whether i and j are part of the same object, i.e. $g_{ij} = 1$, and whether j was pushed. As these probabilies are not known, we need to estimate the parameter vectors

$$\boldsymbol{\theta}_h = \begin{bmatrix} \theta_{h,0} \\ \theta_{h,1} \end{bmatrix} = \begin{bmatrix} p(o_i = 1 | j = v, o_j = 1, g_{ij} = h) \\ p(o_i = 1 | j \neq v, o_j = 1, g_{ij} = h) \end{bmatrix}, \tag{2}$$

for $h \in \{0, 1\}$ together with the conditional distribution over G. The parameters are re-estimated after every action, according to the observed results. The estimate of the parameters Θ at time step t will be denoted by Θ^t.

Conditioning on the parameters, and assuming observations are independent over time, (1) can be written as

$$\begin{aligned} & p(G|\mathcal{D}^t, \Theta^t) \propto p(G, \mathcal{D}^t | \Theta^t) \\ & = p(G) \prod_{i;j>i} p(\mathcal{D}_{ij}^t | g_{ij} = h, \boldsymbol{\theta}_h^t) \\ & \propto p(G) \prod_{i;j>i} \mathrm{Bin}(n_{ij} | N_{ij}, \theta_{h,0}) \mathrm{Bin}(m_{ij} | M_{ij}, \theta_{h,1}), \end{aligned} \tag{3}$$

where $\mathrm{Bin}(n|N, p)$ is the probability mass according to the binomial distribution. Equation (3) is used to evaluate the distribution over graphs G given the data available before and after a potential action. These distributions are then used to calculate the expected gain in information due to pushing vertex v

$$\mathbb{E}_{\mathbf{o}^{t+1}}\big[\mathrm{KL}(p(G|\mathcal{D}^t,\mathbf{o}^{t+1},\Theta^t)\|p(G|\mathcal{D}^t,\Theta^t))|\mathcal{D}^t,v,\Theta^t\big]$$
$$= I(G;\mathbf{o}^{t+1}|\mathcal{D}^t,v,\Theta^t).$$

To evaluate this expectation over all possible observations $\mathbf{o}^{t+1}$, we need to compute

$$p(\mathbf{o}^{t+1}|\mathcal{D}^t,v,\Theta^t)$$
$$= \textstyle\sum_{G\in\mathcal{G}} p(\mathbf{o}^{t+1}|G,v,\Theta^t)p(G|\mathcal{D}^t,\Theta^t), \qquad (4)$$

where we assume that $\mathbf{o}^{t+1}$ is conditionally independent of $\mathcal{D}^t$ given G, and G is independent of the selected action v. Assuming that movement of vertex j is conditionally independent on the movement of other vertexes and graph edges given pushed vertex v and g_{jv}, the conditional distribution

$$p(\mathbf{o}^{t+1}=1|G,v,\Theta^t)$$
$$= \textstyle\prod_j p(o_j^{t+1}=1|g_{jv}=k,v,\Theta^t) = \prod_j \theta_{k,0}.$$

D. Estimating the model parameters

The unknown parameters Θ that represent the probability of observing different events can be estimated together with G using expectation-maximization, by considering G to be a latent variable. Expectation-maximization is an iterative technique for finding maximum likelihood solutions in the presence of hidden variables.

Starting from an initial setting of the parameters Θ^0, the algorithm iterates expectation and maximization steps. In an expectation step, the conditional probability $P(G|\mathcal{D},\Theta)$ is calculated, which is used to calculate a lower bound $Q(\Theta|\Theta^k)$ on the log-likelihood function. $Q(\Theta|\Theta^k)$ expresses the expected log-likelihood of parameters Θ conditioned on the parameters in the previous iteration Θ^k, with Θ^k considered fixed. A maximization step then selects parameters Θ^{k+1} that maximize this expectation. As all variables in this section refer to the variables at the current time step t, time indexes are omitted for clarity.

Assuming observations are conditionally independent given graph G, and assuming $p(G)$ is independent of parameters Θ, the joint likelihood

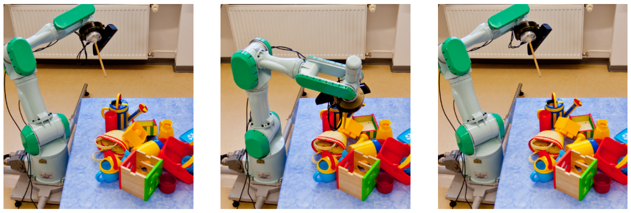
$$p(G,\mathcal{D}|\Theta) = \textstyle\prod_{i;j>i} p(\mathcal{D}_{ij}|G,\Theta)p(G).$$

Hence, the lower bound of the expected log-likelihood

$$\begin{aligned}Q(\Theta|\Theta^k) &= \sum_{G\in\mathcal{G}} p(G|\mathcal{D},\Theta^k)\log(p(G,\mathcal{D}|\Theta))\\ &= \sum_{G\in\mathcal{G}} p(G|\mathcal{D},\Theta^k)\log\left(\prod_{i;j>i} p(\mathcal{D}_{ij}|G,\Theta)p(G)\right)\\ &= \sum_{G\in\mathcal{G}} p(G|\mathcal{D},\Theta^k)\sum_{i;j>i}\log(p(\mathcal{D}_{ij}|G,\Theta)p(G)).\end{aligned}$$

Algorithm 1 Maximally informative interaction algorithm.

```
t ← 0
loop
    estimate $\boldsymbol{\Theta}^t$ and $P(G|\mathcal{D})$
    $v^* \leftarrow \arg\max_v I(G; o^{t+1}|\mathcal{D}^t, v, \boldsymbol{\Theta}^t)$
    execute action $v^*$
    $t \leftarrow t+1$
    observe movement $\mathbf{o}^t$
end loop
```



(a) Scene observation (b) Action execution (c) Effect observation

Fig. 3. Before every action, the robot observes from different viewpoints (a). The observations so far are used to predict the action with the highest information gain. The robot executes the selected action (b), thereby changing the state of its environment. The resulting change is observed (c) and the robot is ready to choose its next action.

In the maximization steps, the lower bound on the log-likelihood function is to be maximized to get the parameter values for the next iteration $\Theta^{k+1} = \arg\max_{\Theta} Q(\Theta|\Theta^k)$. Analytically solving $\nabla_\Theta Q(\Theta|\Theta^k) = \mathbf{0}$ yields

$$\boldsymbol{\theta}_{h,0}^{k+1} = \frac{\displaystyle\sum_{i;j>i} n_{ij}\sum_{G\in\mathcal{G}} p(g_{ij}=h|G)p(G|\mathcal{D},\Theta^k)}{\displaystyle\sum_{i;j>i} N_{ij}\sum_{G\in\mathcal{G}} p(g_{ij}=h|G)p(G|\mathcal{D},\Theta^k)}. \qquad (5)$$

For the second element of the parameter vectors $\boldsymbol{\theta}_{h,1}^{k+1}$ ($h \in \{0,1\}$) the result is similar, but uses m_{ij} and M_{ij} instead of n_{ij} and N_{ij}, respectively. The estimate of the prior $P(G)$ should be updated according to its parametrization.

E. Approximating the information gain

The number of partitions grows at a rate greater than exponentially in the number of vertices. Hence, the evaluation of sums over all $\mathcal{G}$ becomes intractable for high numbers of vertices. However, the expectation of the information gain can be approximated by assuming that

$$P(G) = \prod_{i;j>i} P(g_{ij}).$$

In this case, the expected information gain can be evaluated in polynomial time.

Under this assumption, the Kullback-Leibler divergence between the probability distributions over the graphs before and after a push becomes a sum of the Kullback-Leibler divergences of individual edges. The number of possible observations that we have to take the expectation over also grows faster than exponentially. However, under the assumption of edge independence, we can sample from this distribution in order to efficiently approximate the expectated information gain numerically.

The assumption of edge independence also allows efficient estimation of parameters Θ together with latent variables g_{ij} using expectation-maximization, as explained in Section III-D. In this case, the parameters are $\Theta = \{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\pi}\}$ with $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ as defined in (2) and the prior probabilities specified by $\boldsymbol{\pi}_h = P(g_{ij} = h)$ for $h \in \{0, 1\}$.

The sums in (5) calculate $p(g_{ij} = h|\mathcal{D}, \Theta_k)$. Still assuming the vertex connections g_{ij} are independent, an approximation can be calculated in closed form as

$$\begin{aligned}&\tilde{p}(g_{ij} = h|\mathcal{D}_{ij}, \Theta_k)\\ &\propto \boldsymbol{\pi}_h \prod_{i;j>i} \mathrm{Bin}(n_{ij}|N_{ij}, \boldsymbol{\theta}_{h,0})\mathrm{Bin}(m_{ij}|M_{ij}, \boldsymbol{\theta}_{h,1}).\end{aligned}$$

This approximation is inserted in (5) to estimate the parameters $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$. By solving $\nabla_{\boldsymbol{\pi}} Q(\Theta|\Theta^k) = \mathbf{0}$, we obtain

$$\boldsymbol{\pi}_0 = (1 - \boldsymbol{\pi}_1) = \sum_{i;j>i} p(g_{ij} = 1|\mathcal{D}, \Theta^k)/|E|,$$

with $|E|$ being the number of edges. This approximate inference algorithm has a time complexity cubic in the number of vertices.

F. Action selection and execution

Once the expected information gain of pushing each action is evaluated, the segment that is expected to yield the highest one-step information gain when pushed is selected. A one-step look-ahead is commonly used [5, 9, 13], and Hsiao et al. [19] found that, in their framework, one-step lookahead is usually sufficient.

If the expected information gain for multiple actions is equal, as it is at the beginning of learning, the segment to be pushed is selected at random. The robot pushes the center of the selected segment in a direction corresponding to a projection of the segment's normal in the table plane. Infeasible or dangerous actions, e.g., pushes outside the workspace, are not executed. Instead, the next most informative action is selected. Algorithm 1 summarizes our approach to maximally informative interaction.

G. Observing the effect of the action

After an action has been executed, the resulting scene is observed. Similar to the original observation, the new scene is over-segmented using the methods described in Section III-A. In order to track segments over different observations, the segment centers are initialized using the previous segment centers, as suggested by Heisele et al. [22, 23].

After clustering, the locations of the new segment centers are compared to the locations of the corresponding old segment centers. Any centers that had a displacement larger than a predefined threshold are considered to have moved as a result of the action.

IV. Experiments

Our approach was evaluated on a robot platform together with a system for uninformed action selection for comparison. This section will first describe the experimental set-up and subsequently present our results and a discussion thereof.

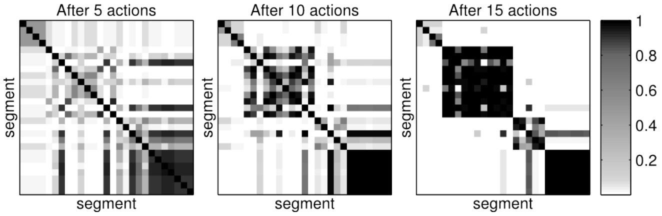


Fig. 4. The matrices show, for every pair of segments, the system's estimate of the probability that they belong to the same object. Results after 5, 10, and 15 actions are shown. The segments have been re-ordered according to the object they belong to. Ideally, this matrix would be a block diagonal.

A. Experimental set-up

The platform used for evaluation is a Mitsubishi PA-10 robot arm. Mounted on this arm are a force-torque sensor, an RGBD camera and a rod for pushing the objects, as shown in Fig. 1. In front of the robot is a table supporting the objects to be segmented. The camera is calibrated such that point clouds can be transformed to the robot's coordinate system. In this manner, point clouds corresponding to different viewpoints of the same scene can be merged. Observing from different view points avoids changing the segmentation too much when the object is pushed. Furthermore, parts of the point clouds that do not correspond to the objects in the robot's workspace can be automatically removed.

The force-torque sensor enables the robot to detect collisions, so that it can operate safely without human supervision. Hence, experiments can be run in an entirely autonomous manner.

This experimental set-up is shown in Fig. 3. It illustrates how the robot observes its environment and subsequently selects and executes the action that it expects to have the highest information gain, based on its current internal representation of the environment. This maximally informative interaction method is described in detail in Section III. We compare this method to an uninformed method that chooses vertices to be pushed at random with a uniform probability.

Initially, four objects are put closely together on the table in front of the robot. They are represented using $k = 30$ segments. In this set-up, selecting the most informative action takes less than a second, a small fraction of the time needed to push and observe the result. Both rigid and non-rigid objects were used in the experiment. After each action, the robot outputs its estimate of the probability that each pair of segments belongs to the same object $p(g_{ij}|\mathcal{D}^{1:t}, \boldsymbol{\Theta}^t) \propto p(\mathcal{D}_{ij}^{1:t}|g_{ij}, \boldsymbol{\Theta}^t)$ as given in (3). These probabilities are calculated in the same manner for the informed algorithm. A human annotation of the segments in the last frame is used as a ground truth.

The average error between the ground truth of g_{ij} and the estimated probability is used as performance measure. This measure is calculated after every action. Segments to which no points, or a large number of points from multiple objects, have mistakenly been assigned are excluded from this calculation. Similar initial configurations were used for both methods.

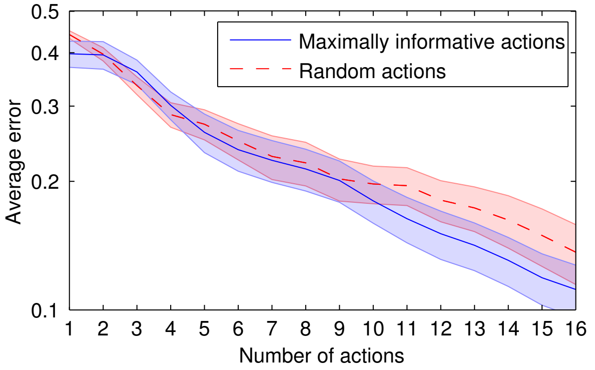


Fig. 5. Number of actions the robot takes in its environment plotted versus the average error per edge for both action selection methods on a log scale. Mean and standard error over 10 trials for each method are shown.

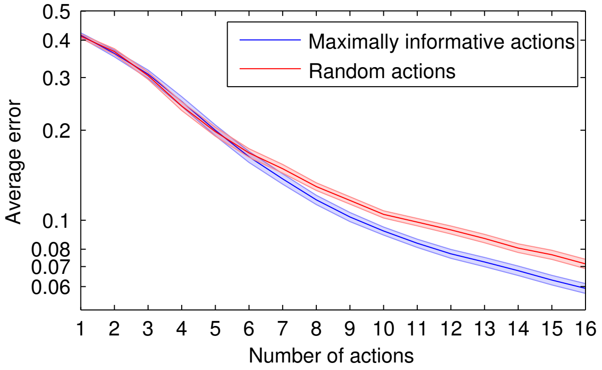


Fig. 6. Number of actions plotted versus the average error per edge. Mean and standard error over 100 simulated trials are shown for each method.

B. *Experimental results*

Each action selection systems was evaluated in ten complete trial runs on a real robot. Examples of the system's estimation of the probability that pairs of segments belong to the same object during one such run is shown in Fig. 4. This figure shows the uncertainty about which vertices belong to the same object. This uncertainty reduces as the results of an increasing number of actions are observed. After 15 actions, the interactive system has come close to a perfect segmentation of the four objects. Fig. 5 shows the average error of the estimated scene structure over all ten runs.

As the extend of evaluations that can be obtained on the robot is limited, we also evaluated the methods on 100 simulated trial runs using twice as many objects and segments. In simulation, all vertices that are part of the pushed objects were considered to move together, but to simulate noise, each element of the observation vector was flipped with a 10% probability. Results of this simulation are shown in Fig. 6. Similar to the real robot experiments, the system needs more random actions than maximally informative actions to reduce the error to the same value.

The data from these simulated runs was analyzed using a paired-sample two-tailed t-test. After performing eight random actions, the error is significantly different from the error after six maximally informative actions, $t(99) = -2.3$, $p < 0.05$, with informative actions resulting in lower errors. This difference becomes larger and more significant as the number of executed actions increases.

C. *Discussion*

Regardless of the action selection method, within 15 actions the system learns about the structure of its environment, with the error between the system's prediction and the true state of the world going down steadily. After 15 actions the error is still decreasing, which indicates a better prediction could be made if the system was allowed to perform more actions. The error seems sufficiently low for the system to make useful predictions about interaction with its environment, which shows that our interactive approach and the chosen graphical representation were effective in this scenario.

By representing the environment using an oversegmentation of the raw sensory information, we could confine possible exploratory behaviors to a relatively small discrete set of actions. Because the set of actions was limited in this case, exploring at random was not an unreasonable strategy.

Even though random action selection was a feasible strategy in the current set-up, the number of actions needed to reduce the error to a certain value was still smaller when maximally informative actions were chosen. For instance, 16 random actions are needed to reduce the error to the value that is obtained by performing just 13 maximally informative actions. However, compared to the variance among trial runs, the advantage of selecting maximally informative actions is limited. We expect this difference to be much larger in high-dimensional or continuous action spaces.

One reason for the large inter-trial variance is that our vision system did not always find a suitable over-segmentation and tracking of segments sometimes failed. Another source of variance is the unmodelled interaction between objects. For example, sometimes objects topple over or consistently bump into each other, while in other trials such events did not take place. However, modelling such interactions would be possible only after obtaining a segmentation, or with the use of additional prior information.

It is interesting to see that the random action selection method seems to have a small initial advantage, both on the real robot and in simulation. This advantage could be due to the fact that after only a few actions the estimate of the parameters is far from the actual values due to a lack of data. Hence, the expected value of the information gain of possible actions will not be reliable. If the information gain cannot be predicted reliably, this lack of knowldedge can cause the action selection system to choose sub-optimal actions. A possible solution to this bootstrapping problem is to initialize the system using a few random actions.

V. Conclusion

In this paper, we have proposed a bottom-up approach to object learning using interaction with the robot's environment. Maximally informative interaction allows robots to efficiently develop knowledge about their environment.

Our interactive approach was applied on a real robot to learn a decomposition of the scene into objects. This segmentation is an important first step in learning about objects, as it allows subsequent exploration to focus on single objects rather than the cluttered scene.

Our experiments show that it is possible to develop this important knowledge completely autonomously. Except for setting up the objects in the robot's workspace, no human supervision at all was needed while the robot explored its environment. This makes it feasible for the robot to learn over longer periods of time.

Useful representations could be learned through exploration using either random or maximally informative action selection. However, these representations improved faster when maximally informative actions were used.

We plan to use our approach to scene segmentation as a stepping stone for learning object properties. Besides intrinsic properties such as shape and mass, it is important for a robot to learn which actions an object affords and how these actions change the state of the objects in the environment. Ideally, a robot would discover autonomously how to change the state of its environment to a desired goal state.

Acknowledgment

The project receives funding from the European Community's Seventh Framework Programme under grant agreement no. ICT-248273 GeRT.

References

[1] D. Beale, P. Iravani, and P. Hall, "Probabilistic models for robot-based object segmentation," *Robotics and Autonomous Systems*, vol. 59, no. 12, 2011.

[2] D. Katz and O. Brock, "Manipulating articulated objects with interactive perception," in *ICRA*, 2008.

[3] J. Kenney, T. Buckley, and O. Brock, "Interactive segmentation for manipulation in unstructured environments," in *ICRA*, 2009, pp. 1377–1382.

[4] P. Fitzpatrick and G. Metta, "Grounding vision through experimental manipulation," *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 361, no. 1811, pp. 2165–2185, 2003.

[5] M. Krainin, B. Curless, and D. Fox, "Autonomous generation of complete 3D object models using next best view manipulation planning," in *ICRA*, 2011.

[6] W. Li and L. Kleeman, "Interactive learning of visually symmetric objects," in *IROS*, 2009, pp. 4751–4756.

[7] A. Ude, D. Omrčen, and G. Cheng, "Making object learning and recognition an active process," *International Journal of Humanoid Robotics*, vol. 5, no. 2, pp. 267–286, 2008.

[8] D. Schiebener, A. Ude, J. Morimotot, T. Asfour, and R. Dillmann, "Segmentation and learning of unknown objects through physical interaction," in *International Conference on Humanoid Robots*, 2011, pp. 500 –506.

[9] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard, "Object identification with tactile sensors using bag-of-features," in *IROS*, 2009, pp. 243–248.

[10] J. Sinapov, T. Bergquist, C. Schenck, U. Ohiri, S. Griffith, and A. Stoytchev, "Interactive object recognition using proprioceptive and auditory feedback," *The International Journal of Robotics Research*, vol. 30, no. 10, pp. 1250–1262, 2011.

[11] J. Modayil and B. Kuipers, "The initial development of object knowledge by a learning robot," *Robotics and Autonomous Systems*, vol. 56, no. 11, 2008.

[12] D. Kraft, R. Detry, N. Pugealt, E. Başeski, F. Guerin, J. Piater, and N. Krüger, "Development of object and grasping knowledge by robot exploration," *Transactions on Autonomous Mental Development*, vol. 2, no. 4, pp. 368–383, 2010.

[13] J. Denzler and C. Brown, "Information theoretic sensor data selection for active object recognition and state estimation," *PAMI*, pp. 145–157, 2002.

[14] D. Katz, Y. Pyuro, and O. Brock, "Learning to manipulate articulated objects in unstructured environments using a grounded representation," in *Robotics: Science and Systems IV*, 2008, pp. 254–261.

[15] W. Li and L. Kleeman, "Autonomous segmentation of near-symmetric objects through vision and robotic nudging," in *ICRA*, 2008, pp. 3604–3609.

[16] L. Chang, J. Smith, and D. Fox, "Interactive singulation of objects from a pile," in *ICRA*, 2012.

[17] L. Natale, F. Orabona, G. Metta, and G. Sandini, "Exploring the world through grasping: a developmental approach," in *International Symposium on Computational Intelligence in Robotics and Automation*, 2005.

[18] D. Kraft, N. Pugeault, E. Başeski, M. Popović, D. Kragić, S. Kalkan, F. Wörgötter, and N. Krüger, "Birth of the object: Detection of objectness and extraction of object shape through object action complexes," *International Journal of Humanoid Robotics*, vol. 5, no. 2, pp. 247–265, 2008.

[19] K. Hsiao, L. P. Kaelbling, and T. Lozano-Pérez, "Robust grasping under object pose uncertainty," *Autonomous Robots*, vol. 31, no. 2, pp. 253–268, 2011.

[20] Commission Internationale de l'Eclairage, "Colorimetry, second edition," Tech. Rep. 15.2, 1986.

[21] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels," EPFL, Tech. Rep. 149300, 2010.

[22] B. Heisele, U. Kressel, and W. Ritter, "Tracking nonrigid, moving objects based on color cluster flow," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 257–260.

[23] B. Heisele and W. Ritter, "Segmentation of range and intensity image sequences by clustering," in *Information Intelligence and Systems, International Conference on*, 1999, pp. 223–225.