Probabilistic Segmentation and Targeted Exploration of Objects in Cluttered Environments

Herke van Hoof, Oliver Kroemer, and Jan Peters

***Abstract*—Creating robots that can act autonomously in dynamic, unstructured environments requires dealing with novel objects. Thus, an off-line learning phase is not sufficient for recognizing and manipulating such objects. Rather, an autonomous robot needs to acquire knowledge through its own interaction with its environment, without using heuristics encoding human insights about the domain. Interaction also allows information that is not present in static images of a scene to be elicited. Out of a potentially large set of possible interactions, a robot must select actions that are expected to have the most informative outcomes to learn efficiently. In the proposed bottom-up, probabilistic approach, the robot achieves this goal by quantifying the expected informativeness of its own actions in information-theoretic terms. We use this approach to segment a scene into its constituent objects. We retain a probability distribution over segmentations. We show that this approach is robust in the presence of noise and uncertainty in real-world experiments. Evaluations show that the proposed information-theoretic approach allows a robot to efficiently determine the composite structure of its environment. We also show that our probabilistic model allows straightforward integration of multiple modalities, such as movement data and static scene features. Learned static scene features allow for experience from similar environments to speed up learning for new scenes.**

I. Introduction

Many tasks require the recognition and manipulation of objects. Therefore, it is essential that robots assisting humans have such capabilities. Since human environments are unstructured, open-ended and dynamic, relying on a pre-specified database of possible objects is most likely insufficient. Rather, robots should learn about novel objects they encounter.

Supervised learning methods have been used to acquire object models, e.g. [1–4]. However, such methods rely on a prestructured data set provided by human teachers, limiting the applicability to newly encountered objects. Additionally, such objects often occur in clutter, whereas classical approaches require the object to be isolated, or segmented by a (human) teacher [2–5]. This requirement is usually not fulfilled when encountering objects in real, cluttered environments where such segmentations are not available.

Therefore, methods that can segment cluttered scenes are required as starting point for learning about the objects at hand [6, 7]. Such methods should be robust to noise and uncertainty (e.g., as induced by sensor- or manipulation failures and occlusions). Furthermore, minimizing the use of heuristics and hand-tuning makes the system more autonomous, by reducing the dependency on human time and effort.

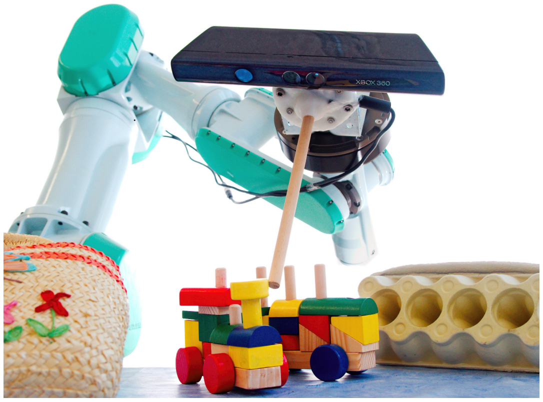


Figure 1: We use a Mitsubischi PA-10 robot arm equipped with a force-torque sensor and an RGBD camera. The robot autonomously interacts with its environment to segment a scene into objects.

In this paper, we use a part-based approach to object segmentation. We use the robot's interaction with its environment to resolve segmentation ambiguities and to model objects robustly. Whereas most current work on robotic scene segmentation focuses on finding a single most likely segmentation, our approach retains a probability distribution over possible segmentations. We will evaluate this approach on real-world segmentation problems.

This probabilistic approach makes segmentation more robust to noise, and endows the robot with knowledge about the uncertainty in its model. Such knowledge can be exploited to choose the most informative action out of many possible actions using information-theoretic insights, whereas most recent work relied on prior training or human-crafted heuristics and domain knowledge. We will show that this criterion helps the robot to learn efficiently.

Extending our past work [8], we define a new probabilistic model that allows motion clues to be integrated with static visual clues in a principled manner. Parameters of the likelihood model of the visual clues are learned from data rather than tuned by hand, allowing the robot to transfer knowledge from previous scenes to the current task. We will show that using static visual clues in addition to motion information will enable our system to determine scene segmentations substantially faster.

H. van Hoof, O. Kroemer and J. Peters are with the IAS institute at TU Darmstadt, Germany (email: hoof,kroemer,peters@ias.tu-darmstadt.de). J. Peters is also with the Max Planck Institute for Intelligent Systems, Tübingen, Germany. Please address correspondence to H. van Hoof.

This is a preliminary version of van Hoof, H.; Kroemer, O.; Peters, J., "Probabilistic Segmentation and Targeted Exploration of Objects in Cluttered Environments," *Robotics, IEEE Transactions on*, vol.30, no.5, pp.1198–1209, 2014. The project received funding from the European Community's Seventh Framework Programme under grant agreement no. ICT-248273 GeRT, no. ICT-270327 CompLACS and no. ICT-610967 TACMAN.

II. Related work

To learn the properties of novel objects in cluttered scenes, segmenting those images is usually a required first step [6, 7]. In this section, we will review various approaches to the scene segmentation problem and contrast our approach to this prior work. First, we will discuss methods that work with one or more static images. Subsequently, we will discuss several interactive approaches to solving this problem. Finally, we highlight a few important aspects of the problem and how these are addressed in prior work: dealing with noise and uncertainty, integrating visual with interactive clues, and efficient exploration.

A. Non-interactive visual segmentation

Traditional segmentation approaches take a single image as input, using clues such as contrast, texture, or color [9]. If 3D-data is available, 3D features such as surface normals or curvature might additionally be exploited [10, 11]. However, visual or spatial boundaries need not always correspond to object boundaries [12, 13], so not all ambiguities can be resolved [12, 14–16]. An alternative is to look at video streams [17]; however, in real-robot setups there may be too much (self-)occlusion for this strategy to be viable. Alternatively, a set of images containing the same objects [18–20] is analyzed together. For example, co-segmentation and co-recognition methods [19, 20] find image segments that occur in multiple images, usually with different backgrounds. Multi-scene analysis [18], on the other hand, finds objects that moved between scenes with the same background.

In these approaches, the robot or system is a passive observer: it uses static images or waits for the environment to change. To learn autonomously, however, it is beneficial for a robot to cause its own changes in the environment.

B. Interactive perception for object segmentation

Physical interaction with objects through pushing, grasping, or lifting enables a robot to learn about them. For learning how actions change the state of objects [13, 15, 21–24], interaction is even required, as the necessary information is not present in static (visual) data.

For learning the appearance and shape of individual objects [15, 25–28], interaction can also be helpful. Through interaction, a robot can obtain multiple views of a scene (for approaches like [18]) autonomously. Besides, knowing what action was performed helps the robot to interpret ambiguous observations, such as movement of an object of interest when there is background movement as well [7].

To avoid the segmentation problem, objects can be physically separated from clutter by grasping [22, 27–30]. Autonomously grasping novel objects, however, is non-trivial by itself and frequently requires knowledge of the object's geometry. Such knowledge is not present for novel objects.

An alternative to grasping is non-prehensile manipulation such as pushing. Employing non-prehensile manipulation for object segmention was pioneered by Fitzpatrick and Metta [15]. Their robot swept its arm across its workspace to detect objects using image differencing. Li and Kleeman [6] refined this method using short, accurate pushes targeted at near-symmetrical objects. The accuracy of segmentations can be increased by accumulating information over time [14].

These image differencing techniques estimate object membership per pixel. To estimate movement direction, a part based representation using an initial over-segmentation can be used [7]. In a subsequent stage, movement and object membership can be estimated for each of these parts. Alternative approaches use algorithms such as iterative closest point (ICP) to determine whether the tracked point cloud is a single rigid object [12, 31], or estimate the movement of trackable visual features [13, 16, 25, 31–33].

Interactive approaches offer powerful clues and enhanced robot autonomy. Therefore, we will employ an interactive segmentation method in our approach.

C. Dealing with noise and clutter

Data coming from robot sensors is frequently noisy and may be incomplete. The possibility of co-movement of multiple objects in cluttered scenes means that observed movement data may often be ambiguous. Thus, with a limited amount of experience, there is uncertainty about the true segmentation. Many of the discussed approaches do not handle occlusion and co-movement as they deal with only one object of interest, e.g., [6, 7, 15]. Noise is often ignored [14, 15] or handled by requiring objects to move as rigid bodies during one or multiple actions [12, 16, 19, 33]. In these approaches, it is not clear how to deal with uncertainty, e.g., from occlusions or with pushes resulting in multiple adjacent objects moving as according to the same homogeneous transform.

The minimization of inconsistent movement is an alternative approach [18]. This approach assumed objects are connected components, which does not always hold in the presence of clutter and occlusions. Beale et al. [7] employed a probabilistic method for correlating a segment's movement with that of the end-effector, but considered only a single object of interest.

Defining a probability distribution over the complete set of possible segmentations of all objects in the scene means that uncertainty can be quantified properly. We expect such an approach to result in robustness in the presence of failures and occlusion. Knowing the uncertainty also allows actions to be selected more robustly. For example, when grasping an object, parts whose object membership is uncertain can be avoided. Therefore, in our approach, we will define a probability distribution over all possible segmentations of the scene to deal with noise and uncertainty. We will show that such an approach can be effectively used for scene segmentation.

D. Combining interaction and visual clues

Interaction can yield powerful clues, but might take a substantial amount of time. In contrast, visual segmentation clues may be ambiguous but are instantly available. Combining both kinds of clues can potentially reduce the interaction time needed to obtain good segmentations.

For example, Katz et al. [13] used hand-tuned predictors based on compactness and appearance, in addition to co-movement of features. Bergström et al. [16] combined rigid motion clues with color- and disparity clues, whereas Schiebener et al. [25] validated hypotheses based on proximity and shared parametric surfaces using co-movement. Hausman et al. [33] used visual features to generate hypotheses, and to reconstruct a dense model from clustered feature points.

The methods discussed in the previous paragraph used visual features to create hypotheses that were tested by interaction [16, 25, 33] or to modify binary potentials between parts [13]. Instead, a *probabilistic approach* offers a principled way to integrate noisy clues from multiple sources. Likelihood terms based on different clues can than be integrated in a principled manner. Learning the parameters of the likelihood model from past experience enables knowledge transfer between different scenes and avoids hand-tuning.

In this paper, we will consequently extend the approach proposed in Sec. II-C with factors corresponding to static scene properties detected by the visual system. We will show that this information speeds up the segmentation process substantially.

E. Efficient learning with informed exploration

In cluttered environments, many different explorative actions are possible. Not all of these actions are equally informative, so that carefully selecting informative actions may decrease the amount of interaction time needed. Surprisingly, all methods discussed in Sec. II-B employed actions that were selected at random, or according to fixed schemes and heuristics. Fixed actions were applied in scenarios with only a single object, and hence such approaches cannot deal with cluttered environments. Heuristics rely on human insights into the problem domain, and as such are not at all guaranteed to work in different domains or unforeseen situations. An exception is the approach of Hausman et al. [33], where actions are chosen based on the probability that the proposed object is the result of over- or under-segmentation. This probability is determined based only on the number of segments assigned to that object, and does not take the likelihood of plausible alternatives into account.

Rather than relying on human cleverness to tweak heuristics, it would be preferable to use general principles that allow the robot to adapt to new situations autonomously. Principles from information theory, for example, can be used to quantify the informativeness of possible actions. For example, perceptual parameters can be chosen to maximize the informativeness of observations [34–37], often after prior training on a specific set of objects or using a physics simulator.

There are several examples of information-theoretic approaches to interactive perception. For example, Krainin et al. employed a next-best-view algorithm based on information gain to select the best viewpoint for in-hand object modeling [26]. This approach assumes the robot knows the objects in its environment sufficiently well to grasp them. Similarly, Sushkov and Sammut [38] selected interactions using the expected information gain, based on discrete sets of possible actions and models provided by a human and training in a physics simulator.

Informative actions can also be found by selecting actions that were successful in uncovering new object properties in similar situations in the past. Katz et al. [39], for example, estimated the value of actions for exploring the kinematic structure of articulated objects using Q-learning. Their approach used a human-crafted, domain-specific representation to generalize past experiences to the current situation.

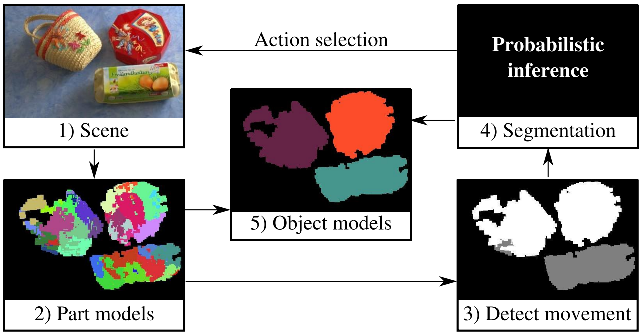


Figure 2: Every scene is processed using our part-based approach: (1,2) Known parts are recognized using stored appearance characteristics and new parts are extracted from regions that have not been seen yet. (3) Movement of known parts (shown in gray) is detected based on the distance between their current and past locations. (4) By merging part models according to the inferred partitioning, a partitioning of parts is determined, corresponding to a segmentation of the scene.

In our approach, knowledge of the uncertainty of the scene's segmentation is captured in a probabilistic model. We can exploit this knowledge to calculate the (approximate) expected information gain of possible actions, without additional hand-tuning. We will use this criterion as a principled way to select informative explorative actions, and show that using such actions improves performance in an example task.

III. Probabilistic segmentation and modeling

In environments cluttered with novel objects, the correct segmentation of the scene into its constituent objects is initially unknown. Hence, appearance characteristics cannot be attributed to the correct object, posing challenges to tracking. We rather attribute appearance characteristic and movement to local regions called 'parts'. This part-based approach allows the resulting motion of an action to be estimated at the object level, which prevents problems associated with estimating such movement at a pixel level [7]. In Sec. III-A and III-B, our approach to obtaining and tracking such regions will be explained. By using the part-based model, the segmentation problem is reduced to finding out how the parts are grouped into objects. How this partitioning is learned by interacting with objects is detailed in Sec. III-C and IV-D. Finally, Sec. III-E describes how actions can be selected to maximize the expected information gain. Fig. 2 illustrates this process.

A. Part extraction and description

We initialize parts using a three-dimensional grid covering the observed point cloud. Each of these grid points ('center points') defines a part consisting of all points within a fixed radius of these center points. Parts may overlap each other. As objects are pushed, previously obscured points of the object surface can become visible. If such points are not within the radius of existing parts, a new part center can be generated at its location. If the radius is set too large, a single part could span multiple objects. On the other hand, setting it too small results in unnessecarily many parts, driving up computational requirements. A radius of 6 cm worked sufficiently well in our experiments, as this radius corresponds roughly to the smallest dimension of our objects.

The parts are tracked using local key point descriptors within their radius. Key points are a sparse set of points that can be detected reliably from multiple views. An expressive description makes sure that correspondences between key points extracted from different views can be matched reliably. As a sparse set of key points is used, calculating and matching descriptors can be done relatively fast.

Our approach does not depend on the particular kind of descriptors employed. In our experiments, key points and descriptors were obtained using the Scale Invariant Feature Transform algorithm [40]. We considered to add Maximally Stable Color Regions [41], as in [25]. However, we found that adding these features did not make a practical difference in tracking for our set of objects (Sec. IV-A), as the observed median difference in part locations was less than 1 cm.

B. Part recognition and movement detection

Our approach requires the detection of the movement of each of the parts resulting from actions taken by the robot. The inference of the scene segmentation is independent of the method employed to detect these movement. We employed an 'eye in hand' set-up, and compare the locations of key-points [40] before and after a push to detect part motion.

Occasionally, false matches occur, even with powerful descriptors. Although we do not assume *objects* are rigid, we do approximate the movement of the object's *parts* by homogeneous transformations, which helps to filter out false matches. We use the random sample consensus (RANSAC) algorithm [42] to robustly find the rigid 3D transformation of the local coordinate frame that explains most key-point matches. In each iteration, a homogeneous transform is fitted using singular value decomposition. In our experiments, we used 100 iterations of the algorithm.

The transformation with the highest number of inliers is accepted if it exceeds a fixed threshold (in our experiments, a quarter of all matches with a minimum of six). We refit he transformation using all found inliers for stability. The refitted transformation is accepted if it has a higher number of inliers.

If the found transformation includes a translation of more than a threshold of three cm (i.e., half of the distance that the robot tries to push the object), we conclude the part has moved. We considered adding a criterion based on pure rotations, but this did not improve the performance of our method. If no rigid transformation with sufficient inliers is found, we conclude that the part is not visible in the scene, for example due to occlusion of the object.

In our current set-up, we are limited to low-frequency change detection instead of continuous tracking because of the minimum distance to the scene (80 cm) required by our sensor. In a hardware set-up with an independently positioned sensor, continuous tracking might yield more robust results, and could deal with textureless objects as well [43].

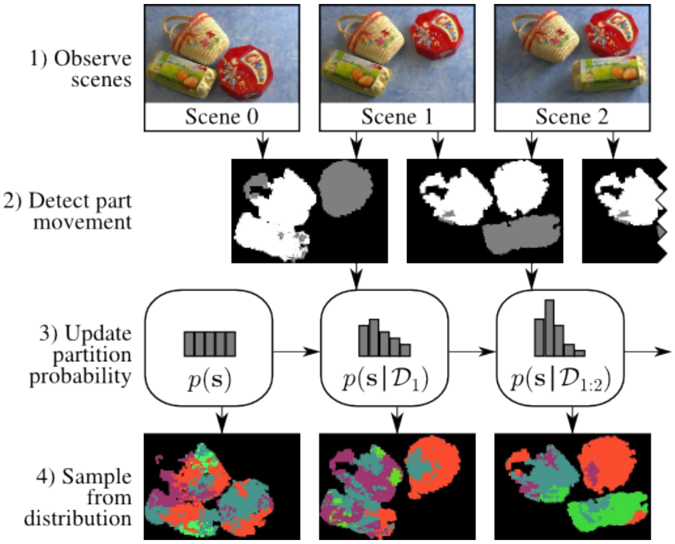

1) Observe scenes
Scene 0
Scene 1
Scene 2
2) Detect part movement
3) Update partition probability
$p(\mathbf{s})$
$p(\mathbf{s}\,|\,\mathcal{D}_1)$
$p(\mathbf{s}\,|\,\mathcal{D}_{1:2})$
4) Sample from distribution

Figure 3: Overview of our probabilistic segmentation approach. After every action, the resulting scene is observed (1) and moving (gray) and non-moving (white) parts are identified (2). This data set $\mathcal{D}$ is subsequently used to update the probability distribution over partitions $\mathbf{s}$ (3). This distribution is approximated using samples (4).

C. Interaction for probabilistic segmentation

To obtain a scene segmentation, the extracted object parts have to be partitioned into groups corresponding to the objects in the scene. Clues for this partitioning are provided by interacting with the environment, which results in the movement of objects. Observing this movement can resolve segmentation ambiguities [14, 15].

Seeing parts move together is more probable when these parts belong to the same objects. However, observed joint movement does not guarantee that features belong to the same object, as the robot's sensors are noisy and objects that push each other also result in joint movement. Therefore, we retain a distribution over possible segmentations, rather than choosing the most likely segmentation. As the movement resulting from consecutive actions is observed, this distribution will generally become narrower, and will have more probability mass at the true segmentation. This process is illustrated in Fig. 3.

Graphical Model and Notation: We represent a segmentation by a vector $\mathbf{s} = [s_1, \ldots, s_N]^T$, with the elements s_i indicating the object containing each of the N parts. For example, the vector $\mathbf{s} = [1, 1, 2]^T$ would indicate a segmentation where the first two parts are assigned to the same object, while the third part is assigned to a different object.

To infer a scene's segmentation the robot has access only to data set $\mathcal{D}_T = \{a_t, \mathbf{o}_t | t \leq T\}$, where a_t is the index of the pushed part at time t and $\mathbf{o}_t$ the resulting observation vector. Its j^{th} value $\mathbf{o}_t[j]$ is equal to 1 if part j was observed moving at time step t, or 0 if it was observed to be still.

We assume that the probability that an object moves is an unknown constant θ_{p} if pushed, or θ_{np} if not pushed. The movement of part n at time t is represented with a latent binary variable $m_{n,t}$. This variable $m_{n,t}$ has the same value for all parts belonging to the same object: $s_n = s_j \implies m_{n,t} = m_{j,t}$. Part n is assumed to be observed moving at time t with an unknown, fixed probability θ_{m} if the object to which it belongs moved ($m_{n,t} = 1$) or θ_{nm} if it did not. The corresponding graphical model is shown in Fig. 4.

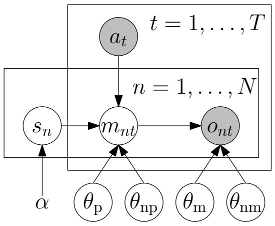


Figure 4: Graphical model for probabilistic segmentation. Shown are parameters, hidden variables (open circles) and observed variables (shaded circles).

Sampling approach for segmentation inference: The number of possible segmentations grows exponentially as the number of parts increases. Hence, calculating the probability of each segmentation quickly becomes computationally intractable. Nevertheless, we can approximate this distribution using samples, which can be drawn using Markov chain Monte Carlo methods such as the Gibbs sampler [44]. Given data set $\mathcal{D}_T$, this approach produces samples from a joint distribution over latent variables by iteratively selecting one latent variable, and assigning it a value according to the assignment probability conditioned on the assignment of all other sampled variables. If we keep only every k^{th} sample, we obtain independent samples for sufficiently large k.

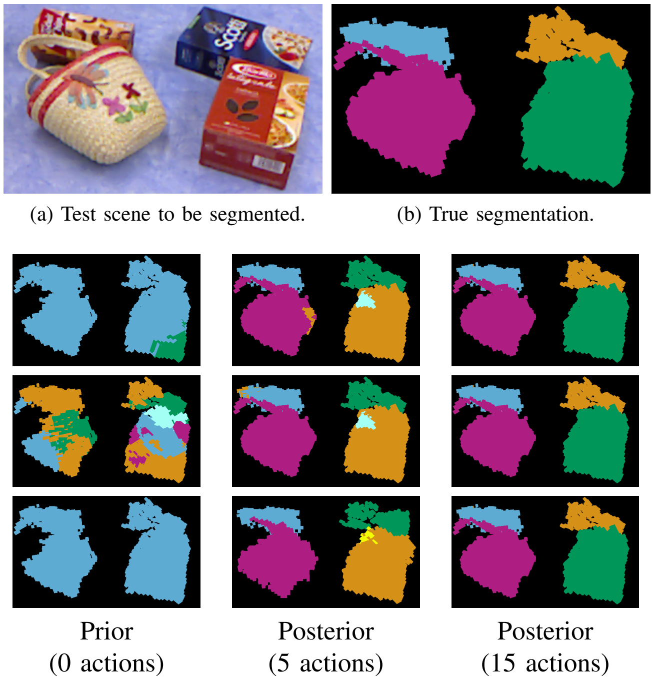
The parameters $\theta_{\mathrm{p}}, \theta_{\mathrm{np}}, \theta_{\mathrm{m}}$ and θ_{nm} can be marginalized in closed form for conjugate priors. Elements s_i of $\mathbf{s}$ and $m_{n,t}$ of $\mathbf{m}$ are sampled according to proposal distributions depending on the current sampled values of all other elements of those vectors, denoted by $s_{\setminus i}$ and $m_{\setminus n,t}$:

$$p(s_i|s_{\setminus i}, \mathbf{a}, \mathbf{m}, \alpha) \propto p(\mathbf{m}|\mathbf{a}, \mathbf{s}) p(s_i|s_{\setminus i}, \alpha), \text{ and} \tag{1}$$

$$\begin{aligned} p(m_{n,t}|\mathcal{D}_T, \mathbf{s}, m_{\setminus n,t}) &\propto p(\mathbf{m}|\mathcal{D}_T, \mathbf{s}) \\ &\propto p(\mathbf{o}|\mathbf{m}) p(\mathbf{m}|\mathbf{a}, \mathbf{s}), \end{aligned} \tag{2}$$

exploiting the conditional independences expressed in the graphical model (Fig. 4). The following subsections will explain how we define the prior $p(s_i|s_{\setminus i}, \alpha)$, the movement model $p(\mathbf{m}|\mathbf{a}, \mathbf{s})$ and the observation model $p(\mathbf{o}|\mathbf{m})$ needed to evaluate these expressions. Examples of samples drawn using this approach are shown in Fig. 5.

Prior distribution over segmentations: The number of objects in the scene is not assumed to be known. Hence, a suitable non-parametric prior distribution $p(s_i|s_{\setminus i})$ over partitionings $\mathbf{s}$ of n parts is given by the Chinese restaurant process [45]. Given the assignment of the other parts $s_{\setminus i}$, part i is assigned to an existing object j with a probability dependent on the number of parts n_j already assigned to that object: $p(s_i = j|s_{\setminus i}) = n_j/(\alpha + n - 1)$. However, with probability $p(s_i = J|s_{\setminus i}) = \alpha/(\alpha + n - 1)$ the part can also be assigned to a new object J, to which no other part has been assigned yet. Due to the non-parametric nature of the Chinese restaurant process, the number of objects does not need to be set in advance, but is learned from the data. The free parameter α controls how often new objects are created by the generative process. In our experiments, α was set to 1.


(a) Test scene to be segmented.

(b) True segmentation.

(c) Samples of the distribution over segmentations $\mathbf{s}$, coloring the parts according to the object they are assigned to. The columns contain three independent samples from the distribution after observing the effect of 0, 5 and 15 actions, respectively.

Figure 5: Samples of the distribution over segmentations of a test scene. A priori, the number of objects as well as the segmentation are unknown. Therefore, samples of the prior are very different from each other. Over time, the number of objects and the correct segmentation are inferred. The growing consistency of the samples indicates decreasing uncertainty.

The movement model: To evaluate Eqs. (1) and (2), we define a movement model

$$\begin{aligned} p(\mathbf{m}|\mathbf{a}, \mathbf{s}) &= \int_0^1 \int_0^1 p(\mathbf{m}|\mathbf{s}, \theta_{\mathrm{p}}, \theta_{\mathrm{np}}) p(\theta_{\mathrm{p}}, \theta_{\mathrm{np}}|\mathbf{s}) \mathrm{d}\theta_{\mathrm{p}} \mathrm{d}\theta_{\mathrm{np}} \\ &\propto \int_0^1 \theta_{\mathrm{p}}^{\alpha'_{\mathrm{p}}} (1 - \theta_{\mathrm{p}})^{\beta'_{\mathrm{p}}} \mathrm{d}\theta_{\mathrm{p}} \int_0^1 \theta_{\mathrm{np}}^{\alpha'_{\mathrm{np}}} (1 - \theta_{\mathrm{np}})^{\beta'_{\mathrm{np}}} \mathrm{d}\theta_{\mathrm{np}} \\ &\propto \frac{\alpha'_{\mathrm{p}}! \beta'_{\mathrm{p}}!}{(1 + \alpha'_{\mathrm{p}} + \beta'_{\mathrm{p}})!} \frac{\alpha'_{\mathrm{np}}! \beta'_{\mathrm{np}}!}{(1 + \alpha'_{\mathrm{np}} + \beta'_{\mathrm{np}})!}, \end{aligned}$$

with factorizing conjugate prior

$$p(\theta_{\mathrm{p}}, \theta_{\mathrm{np}}|\mathbf{s}) = \mathrm{Beta}(\theta_{\mathrm{p}}|\alpha_{\mathrm{p}}, \beta_{\mathrm{p}}) \mathrm{Beta}(\theta_{\mathrm{np}}|\alpha_{\mathrm{np}}, \beta_{\mathrm{np}}),$$

and α'_{p}, α'_{np} are the corresponding α_{p}, α_{np} plus the number of times an object moved given that it was pushed (α'_{p}) or not pushed (α'_{np}). Similarly, β'_{p}, β'_{np} are the corresponding β_{p}, β_{np} plus the number of times an object did not move given that it was pushed (β'_{p}) or not pushed (β'_{np}). We set the hyper-parameters $\alpha_{\mathrm{p}} = \beta_{\mathrm{np}} = \alpha_{\mathrm{np}} = \beta_{\mathrm{p}} = 1$ to encode a uniform prior.

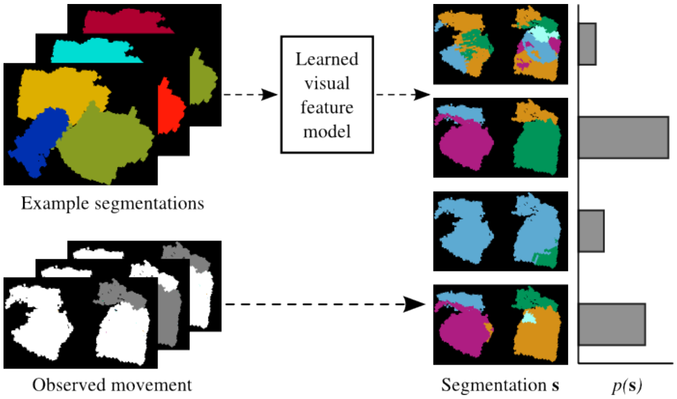


Figure 6: Statistical properties of visual features in previously-seen scenes can be modeled. The visual likelihood of segmentations of a new scene is combined with movement likelihood to improve calculated segmentation probabilities.

The observation model: Analogous to the movement model, we define the observation model

$$\begin{aligned} p(\mathbf{o}|\mathbf{m}) &= \int_0^1\int_0^1 p(\mathbf{o}|\mathbf{m},\theta_\mathrm{m},\theta_\mathrm{nm})p(\theta_\mathrm{m},\theta_\mathrm{nm})\mathrm{d}\theta_\mathrm{m}\mathrm{d}\theta_\mathrm{nm} \\ &\propto \int_0^1 \theta_\mathrm{m}^{\alpha'_\mathrm{m}}\left(1-\theta_\mathrm{m}\right)^{\beta'_\mathrm{m}}\mathrm{d}\theta_\mathrm{m}\int_0^1 \theta_\mathrm{nm}^{\alpha'_\mathrm{nm}}\left(1-\theta_\mathrm{nm}\right)^{\beta'_\mathrm{nm}}\mathrm{d}\theta_\mathrm{nm} \\ &\propto \frac{\alpha'_\mathrm{m}!\beta'_\mathrm{m}!}{(1+\alpha'_\mathrm{m}+\beta'_\mathrm{m})!}\frac{\alpha'_\mathrm{nm}!\beta'_\mathrm{nm}!}{(1+\alpha'_\mathrm{nm}+\beta'_\mathrm{nm})!}, \end{aligned}$$

with factorizing conjugate prior

$$p(\theta_\mathrm{m},\theta_\mathrm{nm}) = \mathrm{Beta}(\theta_\mathrm{m}|\alpha_\mathrm{m},\beta_\mathrm{m})\mathrm{Beta}(\theta_\mathrm{nm}|\alpha_\mathrm{nm},\beta_\mathrm{nm}).$$

The variables α'_m, α'_nm, β'_m, β'_nm express the number of parts that were observed moving or not moving given movement of the corresponding object, added to the respective hyper-parameters $\alpha_\mathrm{m} = \beta_\mathrm{nm} = \alpha_\mathrm{nm} = \beta_\mathrm{m} = 1$. In subsequent sections, we will need the data likelihood

$$p(\mathcal{D}_T|\mathbf{s}) = \mathbb{E}_\mathbf{m}\left[p(\mathcal{D}_T|\mathbf{m})|\mathbf{s}\right] \approx J^{-1}\sum_{j=1}^{J}p(\mathcal{D}_T|\mathbf{m}_j),$$

using the conditional independence of $\mathcal{D}_T$ from $\mathbf{s}$ and approximating the expectation with samples $\mathbf{m}_j \sim p(\mathbf{m}|\mathbf{s})$. Furthermore, we will need the observation probability

$$p(\mathbf{o}|a,\mathbf{s},\mathcal{D}_T) = \mathbb{E}_\mathbf{m}\left[p(\mathbf{o}|\mathbf{m})|\,a,\mathbf{s},\mathcal{D}_T\right] \approx J^{-1}\sum_{j=1}^{J}p(\mathcal{D}_T|\mathbf{m}_j),$$

using the conditional independence of $\mathbf{o}$ and samples $\mathbf{m}_j \sim p(\mathbf{m}_j|a,\mathbf{s},\mathcal{D}_T)$.

D. Integrating visual clues

In case we have access to vison data set $\mathcal{D}_\mathrm{vis}$ besides interaction data set $\mathcal{D}_T$, it is straightforward to adapt our model

$$p(\mathbf{s}|\mathcal{D}_T,\mathcal{D}_\mathrm{vis}) \propto p(\mathcal{D}_T|\mathbf{s})p(\mathcal{D}_\mathrm{vis}|\mathbf{s})p(\mathbf{s}),$$

under the assumption that data sets $\mathcal{D}_T$ and $\mathcal{D}_\mathrm{vis}$ are conditionally independent (see Fig. 6). One important clue obtained from vision is the spatial distribution of the parts in the scene.

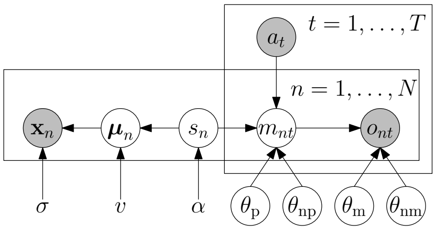


Figure 7: Graphical representation of the extended model.

If we assume objects to be spatially compact, the locations $\{\mathbf{x}_j|s_j = k\}$ of parts object k are clustered around some location $\boldsymbol{\phi}_k$. We assume $p(\mathbf{x}_j|\boldsymbol{\phi}_{s_j}) = \mathcal{N}(\mathbf{x}_j;\boldsymbol{\mu}_j,\Sigma)$ to be a normal distribution with $\boldsymbol{\mu}_j = \boldsymbol{\phi}_{s_j}$. The variables $\boldsymbol{\phi}_i$ are latent; a conjugate prior over latent variables $\boldsymbol{\phi}_k$:$p(\boldsymbol{\phi}_k) = \mathcal{N}(\boldsymbol{\phi}_k;\mathbf{0},V)$ allows marginalization in closed form. The desired likelihood

$$p(\mathcal{D}_\mathrm{vis}|\mathbf{s}) = \prod_{k\in\mathcal{K}}\left(\iint_{\Re^2}p(\boldsymbol{\phi}_k)\prod_{j\in\mathcal{J}_k}p(x_j|\boldsymbol{\phi}_{s_j})\mathrm{d}\boldsymbol{\phi}_k\right),$$

where $\mathcal{K}$ is the set of objects in the current segmentation and $\mathcal{J}_k = \{j|1 \le j \le N, s_j = k\}$ is the set of indexes of parts that belong to object k. The extended model is shown in Fig. 7.

For simplicity, we assume the normal distributions to be isotropic ($\Sigma = \sigma I$, $V = vI$) and train the parameters σ and v on a data set of already-segmented scenes using the method of maximal marginal likelihood using a gradient ascent approach. No additional tuning is required.

Our modeling approach is quite general: different kinds of features (e.g., based on color or shape) can be integrated in the model in the same manner. Only a suitable distribution over the properties of the objects and their parts is required. We chose the location feature due to its generality, as spatial compactness holds for a wide variety of objects and scenes.

E. Maximizing mutual information for directed exploration

Our approach generates part models and approximates a distribution over segmentations regardless of the chosen actions. However, if our robot deliberately chooses informative actions, we expect useful object models to be learned faster [35–37, 46]. Hence, we choose actions to maximize the mutual information $I\left(\mathbf{s};\mathbf{o}|a,\mathcal{D}_T\right)$, where $\mathbf{s}$ is the partition of the parts into objects and $\mathbf{o}$ is the observed outcome of the action targeted at part a at $t = T+1$. The mutual information is equal to the expected information gain of observing the result of pushing part a given by

$$\begin{aligned} I\left(\mathbf{s};\mathbf{o}|a,\mathcal{D}_T\right) &= \mathbb{E}_\mathbf{o}\left[D_\mathrm{KL}(p(\mathbf{s}|\mathbf{o},a,\mathcal{D}_T)||p(\mathbf{s}|\mathcal{D}_T))|a,\mathcal{D}_T\right] \\ &= \mathbb{E}_{\mathbf{s},\mathbf{o}}\left[\log\left(\frac{p(\mathbf{s},\mathbf{o}|a,\mathcal{D}_T)}{p(\mathbf{o}|a,\mathcal{D}_T)p(\mathbf{s}|a,\mathcal{D}_T)}\right)\middle|\,a,\mathcal{D}_T\right], \end{aligned}$$

where D_KL is the Kullback-Leibler divergence. The argument of the logarithm is computed using

$$\begin{aligned} \frac{p(\mathbf{s},\mathbf{o}|a,\mathcal{D}_T)}{p(\mathbf{o}|a,\mathcal{D}_T)p(\mathbf{s}|a,\mathcal{D}_T)} &= \frac{p(\mathbf{o}|\mathbf{s},a,\mathcal{D}_T)p(\mathbf{s}|a,\mathcal{D}_T)}{p(\mathbf{o}|a,\mathcal{D}_T)p(\mathbf{s}|a,\mathcal{D}_T)} \\ &= \frac{p(\mathbf{o}|\mathbf{s},a,\mathcal{D}_T)}{\mathbb{E}_{s'}\left[p(\mathbf{o}|\mathbf{s}',a,\mathcal{D}_T)|\mathcal{D}_T\right]}, \end{aligned}$$

as $p(\mathbf{s}|a,\mathcal{D}_T) = p(\mathbf{s}|\mathcal{D}_T)$. The spaces S and O of possible partitions and observations grow exponentially as the number of parts increases. Hence, evaluating these expectations exactly is infeasible. We can approximate these expectations using samples $(\mathbf{s}_{(j)}, \mathbf{o}_{(j)}) \sim p(\mathbf{s},\mathbf{o}|a,\mathcal{D}_T)$, $j \in \{1,\dots,J\}$ and samples $\mathbf{s}_{(k)} \sim p(\mathbf{s}|\mathcal{D}_T)$, $k \in \{1,\dots,K\}$, i.e., by computing

$$I\left(\mathbf{s};\mathbf{o}|a,\mathcal{D}_T\right) \approx \frac{1}{J}\sum_j \log\left(\frac{p(\mathbf{o}_{(j)}|\mathbf{s}_{(j)},a,\mathcal{D}_T)K}{\sum_k p(\mathbf{o}_{(j)}|\mathbf{s}_{(k)},a,\mathcal{D}_T)}\right). \tag{3}$$

The samples from $p(\mathbf{s}|\mathcal{D}_T)$ can be obtained using the Gibbs sampler described in Sec. III-C. Now, we sample from the conditional $p(\mathbf{o}|\mathbf{s},a,\mathcal{D}_T)$ to get a sample from $p(\mathbf{o},\mathbf{s}|a,\mathcal{D}_T)$. Furthermore, we need the conditional observation probability

$$p(\mathbf{o}|a,\mathbf{s},\mathcal{D}_T) = \frac{p(\mathbf{o},\mathcal{D}_T|\mathbf{s},a)}{p(\mathcal{D}_T|\mathbf{s})},$$

as $p(\mathcal{D}_T|\mathbf{s},a) = p(\mathcal{D}_T|\mathbf{s})$. We determine $p(\mathcal{D}_T|\mathbf{s})$ as described in Sec. III-C, and compute $p(\mathbf{o},\mathcal{D}_T|\mathbf{s},a)$ similarly after adding the potential action a and observation $\mathbf{o}$ to the actual actions and observations in $\mathcal{D}_T$.

IV. Evaluation

In the proposed approach, a robot uses probabilistic inference to segment a cluttered scene based on interaction data. In this section, we will first introduce our general experimental set-up in Sec. IV-A. Subsequently, in Sec. IV-B we compare our probabilistic segmentation method to alternative approaches on data gathered by a real robot. In Sec. IV-C, we consider a scenario where action selection according to the mutual information criterion is needed to learn efficiently, and compare that strategy to random action selection. Finally, we evaluate the inclusion of a likelihood term based on the visually observed spatial distribution of object parts in addition to interaction data in Sec. IV-D.

A. Experimental set-up and quality measure

We evaluated our approach using a 7 degrees of freedom Mitsubishi PA-10 robot arm. A RGBD camera, a force-torque sensor, and a rod used to manipulate the objects were mounted on the arm's end effector (see Fig. 1). Hence, the robot could move the camera to observe the scene from different perspectives. The force-torque sensor allowed the robot to register forces exerted on the rod, which allowed the robot to autonomously stop its motion in case of unexpected collisions. The camera was calibrated. Consequently, observations taken from different points of view could be aligned straightforwardly by transforming them to the robot's coordinate frame. Since the table location is known, observed parts not belonging to the scene on the table could automatically be removed.

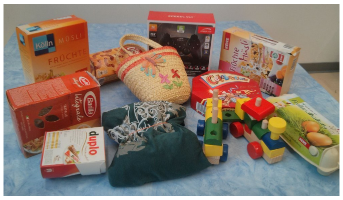
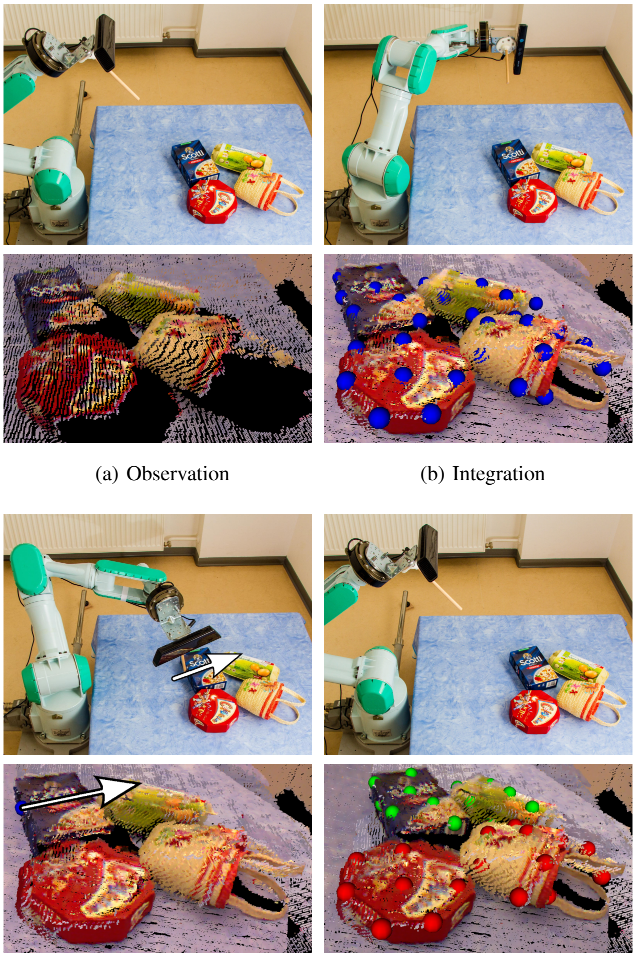


Figure 8: The set of 12 everyday objects used in our experiments. We included objects of different shapes and an articulated object (train), a deformable object (cloth bundle) and a flexible object (basket).



(a) Observation

(b) Integration

(c) Push

(d) Observation

Figure 9: Illustration of the exploration phase. (a) The robot observes the scene, obtaining an incomplete point cloud from one perspective. (b) Percepts from multiple perspectives are integrated and patches are extracted (part centers shown as blue spheres). (c) A push is selected (bottom, blue sphere) and executed. (d) The resulting scene is observed and the patch centers are registered as moving (green) or non-moving (red).

The robot was presented with a cluttered scene of novel objects taken from the set shown in Fig. 8. These objects were set up on a table next to the robot. The robot interacted with the scene, pushing the selected part near its center in a direction based on its estimated surface normal. After every action, the scene was observed from three different view points in order to update the distribution over partitions of the parts into objects (Sec. III-C). The entire procedure is illustrated in Fig. 9. Performing this procedure took about one minute for each action, most of which was used by actual movement of the robot. Our robot was not moving at maximum speed to keep operation controlled and safe.[1]

Quality measure: After every action, the robot updates its posterior probability distribution over segmentations. Parts that belong to the same object according to the ground truth

[1] A video showing our set-up is available at: http://youtu.be/GQYP2eYaGks.

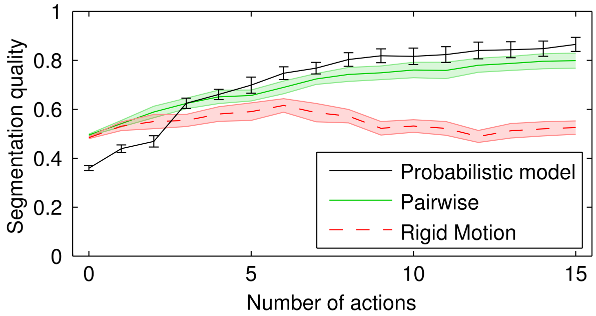


Figure 10: Inference of scene segmentations using different inference methods. Parameter learning increases initial uncertainty in the probabilistic model, which reduces segmentation quality initially. The error bars and shaded areas show the standard error.

(human annotation), should be assigned likewise by the robot. The robot decides the parts should belong to the same object if the majority of samples assigns them so (and vice versa).

Following Fowlkes and Mallows [47], the quality measure we used was the correspondence

$$B = \frac{|P \cap Q|}{\sqrt{|P||Q|}}, \tag{4}$$

where Q and P are the set of pairs of parts that belong to the same object according to the human annotation (Q) or according to the model's prediction (P), and $|\cdot|$ denotes a set's cardinality. This correspondence is zero if ground truth and prediction do not agree on any pair of parts. Conversely, the correspondence is one if they agree on all pairs.

B. Interactive segmentation experiment

During interaction with its environment, the robot obtained information that allowed it to narrow down its distribution over possible segmentations. The number of objects and the way the parts should be assigned to those objects were inferred simultaneously with the parameters $\theta_{\mathrm{p}}, \theta_{\mathrm{np}}, \theta_{\mathrm{m}}$ and θ_{nm}.

We compared our probabilistic model to two heuristics commonly employed in the field of interactive segmentation. Neither of these baselines directly re-implements a particular approach from the literature: interactive segmentation approaches usually have a strong interdependency between set-up, sensing, representation, inference and action selection, making it difficult to execute such a direct comparison in a fair manner. These baselines are:

Rigid motion approach: If parts do not follow the same rigid transformation, they need to belong to different objects.

Pairwise approach: Two parts belong together if co-movement is observed more often than separate movement, independently from any other parts.

The set of twelve different objects (Fig. 8) was used to create fifteen initial set-ups consisting of four objects each. The robot explored these objects using fifteen random actions. Only motion data was used in this experiment. The same data set was used for all methods.

Discussion: In the segmentation task, we evaluated the quality of the segmentation the robot inferred through interaction. The robot learned continuously from its own experience, without an annotated training set or reward signals.

The experiment required inferring the segmentation of scenes composed of rigid and non-rigid objects using just movement data. Considering these circumstances, both the pairwise method and our full model did quite well. Both comparison methods were outperformed by our probabilistic segmentation approach (see Figs. 10 and 11). Our approach attained an average quality of 0.86 in contrast to 0.80 for the pairwise method. This difference is a relevant step towards perfect segmentations (1.00). The rigid motion method is sensitive to tracker errors, and attained an average quality of only 0.52. After 15 actions all methods except for the rigid motion method appear to have converged. Our probabilistic approach needed only nine actions for half of the trial runs to attain a segmentation quality of at least 0.85, while this quality was not reached within 15 actions for the pairwise and rigid motion methods (see Figs. 10 and 11).

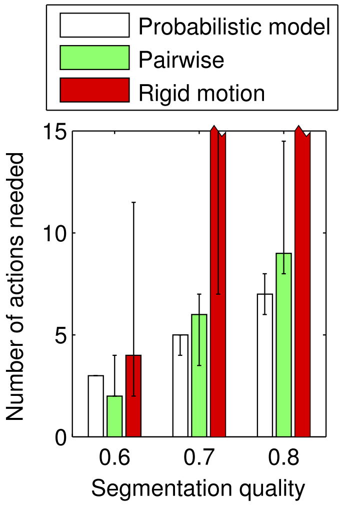


Figure 11: Scene segmentations using different inference methods. Bars show the median number of actions needed, errorbars indicate the interquartile range where it is different from the median. Trials were cut off after 15 actions.

Our scenario is different from that of related work in terms of the available knowledge. Related work often focused on scenario's where knowledge of the objects' movement type (e.g., rigid transformation) were given to the robot [7, 12, 16, 19, 33], whereas in our case only the occurence of movement was used. This additional flexibility has the cost that we usually need more actions than these related approaches. For example, parameters of our probabilistic model ($\theta_{\mathrm{p}}, \theta_{\mathrm{np}}, \theta_{\mathrm{m}}$ and θ_{nm}) are learned rather than tuned which increases uncertainty in the beginning of the experiment, reducing performance of our learned model relative to manually specified models initially. The baseline methods, on the other hand, do not optimally exploit the information in larger data sets. One cause for occasional failure in all of the methods was that the tracker system occasionally loses certain parts (see Sec. IV-C).

A qualitative advantage of our method over the pairwise method is that it respects the transitivity of the 'belongs to the same object' relation: if part i and j belong together and so do j and k, the same necessarily holds for i and k. The pairwise method cannot guarantee this consistency.

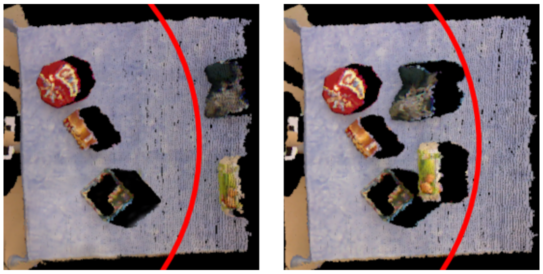


Figure 12: Top-view of the set-up, with the robot located outside of the illustration on the left-hand side. In the initial set-up, two objects are visible to the robot but outside of its workspace (indicated by the red arc). After the tenth action (left), those objects are moved manually within the robot's workspace (right). This movement is not used in inference. After observing the resulting situation, the robot continues with the eleventh action.

C. Action selection experiment

In a second experiment, we evaluate how much the robot gains by exploiting its knowledge of the segmentation uncertainty to select more informative actions. When all objects can be manipulated equally easily, random action selection performs fairly well as it tends to distribute actions evenly over all objects. However, objects are not always consistently reachable. Some objects might even be entirely out of the robot's workspace, and can only be manipulated later.

Therefore, we used a set-up similar to the previous experiment, however, every scene included five objects of which two were initially placed outside of the robot's workspace. The objects were selected from the set shown in Fig. 8, and ten independent trials were executed. After training on the three remaining objects for ten actions (using a random action selection strategy), these objects were placed such that they became reachable to the robot (Fig. 12).

Subsequently, five more actions were executed using one of two action selection strategies: selecting actions at random or according to the maximal mutual information criterion (Sec. III-E). To speed up computation, we set the parameters $\theta_{\mathrm{np}}, \theta_{\mathrm{p}}, \theta_{\mathrm{m}}, \theta_{\mathrm{nm}}$ to their MAP estimate using the data gathered in the previous experiment. Action selection took less than one second. The action's resulting movement was used to infer the segmentation using our probabilistic approach.

Discussion: The results of the action selection experiments are shown in Fig. 13. After the first ten actions, three objects initially in the workspace had been explored using random actions. When two initially out-of-reach objects were introduced into the workspace, random actions were divided over all five objects, with a bias toward objects that were already in the robot's workspace. This bias might occur because those objects tended to be closer to the robot after the initial pushes, so more of their parts were reachable to be pushed. The action selection strategy employing the mutual information criterion focused explorative actions on the novel objects, overcoming the bias observed in the random strategy.

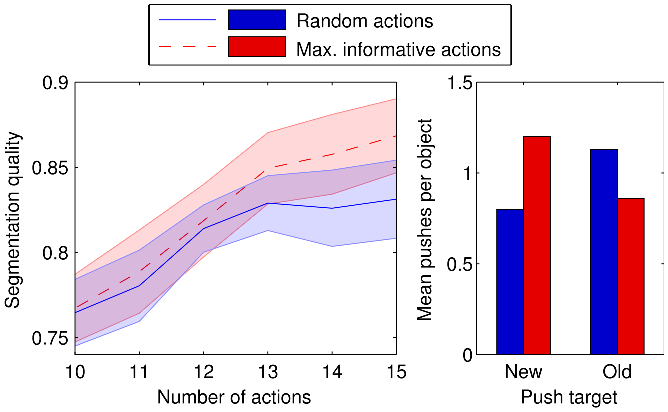


Figure 13: Action selection experiment. On the left, the segmentation quality is shown starting from the tenth action, when two objects are moved inside the robot's workspace. The right graph shows the mean number of pushes directed at existing and newly introduced objects during this period. The shaded areas show the standard error.

Timesteps lost, #	0	1-3	4-7	8-14
Occurence, %	65%	15%	9%	11%
Mean error contribution	-1.6	-2.1	-2.3	-4.8

Table I: The tracker sometimes fails to localize parts. The stated contribution is the number of pairs of parts including the lost part that do not contribute to $|P \cap Q|$ in the quality measure (Eq. 4). Shown are only parts that moved at least once during the experiment to avoid confounding movement and tracking success.

Analysis of segmentation errors: Our probabilistic interactive segmentation approach, whether employing actions deemed maximally informative or not, often does not attain the maximally possible segmentation quality of 1.0. We investigated a number of possible causes to understand where our set-up or algorithms can be improved.

First of all, we looked at our sampling method. Multimodal posterior distributions can cause Markov Chain samplers to get stuck in suboptimal local maxima of the likelihood function. In that case, we expect chains with different starting points to reach different local maxima. We compared chains starting with all parts in their own cluster, all parts in a single cluster, or with the ground truth (ensuring we are near a good local maximum). The mean segmentation quality after observing 15 actions was, respectively, 0.79, 0.80, and 0.80, with a standard deviation of 0.07 in each case. The likelihood values of the samples were comparable as well. Therefore, local maxima are unlikely to explain failures in our experiments.

Next, we looked at tracking errors. Sometimes, the tracker seems to recognize a part but returns the wrong location. Such false positives only occur in about 1% of the parts during each trial run, and can be handled as noise by our method. Losing track of parts happen more often, e.g. during occlusing or illumination changes (Tab. I). Occasionally failures can be handled by our method, but about 11% of the parts are lost for more than half of the 15 time steps, which decreases the system's performance. Tracker performance is therefore a target for future improvement (Sec. III-B).

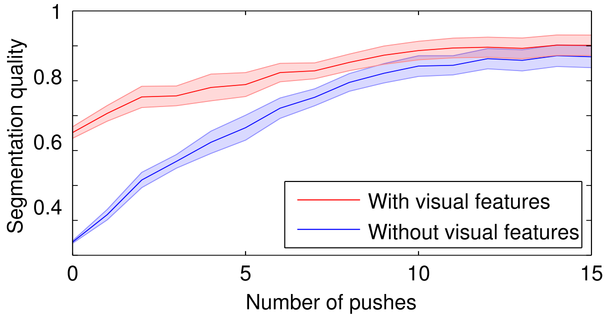


Figure 14: Experiment with random actions and set-ups with a cluster of four objects. Especially when little interaction data is available, knowledge about visual features (spatial proximity) helps to infer the correct segmentation. Shaded areas show the standard error.

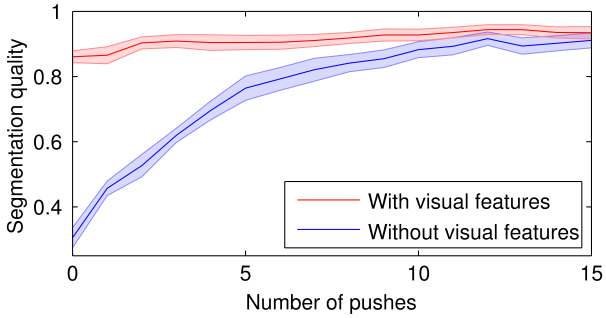


Figure 15: Experiment with mutual information action selection and set-ups with one cluster of three objects and one isolated object. On scenes where one object is initially separated, visual features are more powerful. Shaded areas show the standard error.

D. Integration of motion and visual clues experiment

As suggested in Sec. III-D, we combined the observed movement data with visually observed properties, in this case the spatial arrangements of objects in the beginning of the experiment. The parameters of the likelihood model are set by maximizing the marginal likelihood on a separate data set of 20 scenes. The likelihood parameters of the motion model were again set to their MAP values. We performed two experiments with different action selection criteria.

Random actions: We evaluated the model with and without the additional visual clues on 10 sequences from the data set used in Sec. IV-B for which the visual information was available (in each, 15 random actions were performed). The results of this experiment are shown in Fig. 14.

Informative actions: We performed twelve trials with the objects set up in two clusters: one containing three objects and the other with a single object (from the set in Fig. 8). In this set-up, we evaluate how action selection using the mutual information criterion can exploit information from static features. We hypothesize that exploration will focus on the larger cluster which is visually more ambiguous. Results from this experiment are shown in Figs. 15 and 16.

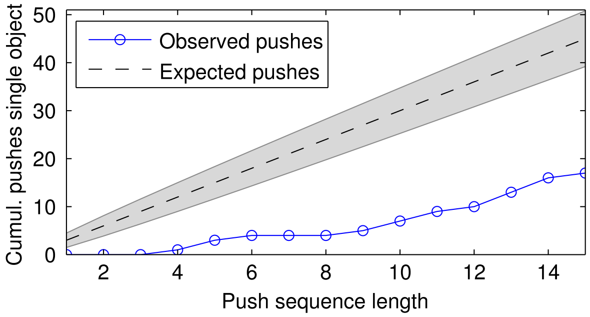


Figure 16: Experiment with mutual information action selection and set-ups with one cluster of three objects and one isolated object. We show the cumulative count of pushes targeted at the isolated object during the indicated sequence length. The shaded area represents the standard deviation of a binomial distribution.

Discussion: At the start of the experiments, no interaction data were available so the baseline approach guessed blindly. When using the static visual features, we obtained much better results. As interaction data became available, the gap between the approaches diminished but the method that employed additional static visual features stayed numerically better.

As shown in Figs. 15 and 16, in case we started with one object separated from the other objects, the robot avoided this object in subsequent exploration. Only after exploring the remaining cluster of objects, the robot targeted the single object more frequently.

In related work, visual clues were often used to provide a set of hypothesis to be confirmed by interaction [16, 25, 33]. In contrast, in our method we regard both visual and interactive clues as (noisy) information channels. Instead of maintaining a discrete set of hypothesis, our method assigns a, usually non-zero, probability to any possible segmentation. This view allows information fusion from both channels in a principled way. Parameters for the visual feature model were determined by training on previous (known) scene segmentations, allowing knowledge to be transferred between scenes. Hence, no manual tuning was needed for setting the hyperparameters of the visual feature model and for the integration of visual features and interaction features.

V. Conclusions and Future Work

In this paper, we have presented a part-based, probabilistic approach to interactive segmentation. Our approach aims at minimizing human intervention: the robot learns from the effects of its actions, rather than human-given labels, and the amount of tuning needed is limited by employing Bayesian methods (enabling important parameters to be marginalized given largely uninformative hyper-priors) and machine learning approaches to parameter setting (maximum marginal likelihood).

Our experiments showed that, firstly, our approach functions and is relatively robust to noise and co-movement, even in a complicated real-world environment including tracking failures and co-movement of different objects. As we employ a probabilistic representation, our robot has knowledge on the segmentation uncertainty. This knowledge can be exploited to select more informative actions. In a second experiment, we have shown that our information-theoretic scheme for action selection enables the robot to learn faster about new objects in its environment then a random baseline, by directing more explorative actions at those novel objects.

Another advantage of a probabilistic representation is that it offers a straightforward way to integrate clues for different sources, as conditionally independent clues can be integrated by multiplying their likelihood functions together. Specifically, we studied a spatial proximity feature. We avoided hand-tuning of the hyper-parameters of the spatial likelihood model by learning them from a set of scenes the robot has previously interacted with. The learned parameters allowed knowledge on typical spatial structures to be transferred to new scenes. We have shown that this transfer makes determination of the underlying segmentation substantially faster and can focus action selection to the most ambiguous parts of the scene.

Possible future research topics to improve performance include improving tracking performance, integrating more visual clues, such as color clues, and making the Markov Chain more efficient by a larger variety of moves.

References

[1] A. Pope and D. Lowe, "Probabilistic models of appearance for 3-d object recognition," *Int. J. Comput. Vision*, vol. 40, no. 2, pp. 149–167, 2000.

[2] A. Collet, D. Berenson, S. Srinivasa, and D. Ferguson, "Object recognition and full pose registration from a single image for robotic manipulation," in *Int. Conf. Robotics and Automation*, 2009, pp. 48–55.

[3] H. Murase and S. Nayar, "Visual learning and recognition of 3-D objects from appearance," *Int. J. Comput. Vision*, vol. 14, no. 1, pp. 5–24, 1995.

[4] A. Mian, M. Bennamoun, and R. Owens, "Three-dimensional model-based object recognition and segmentation in cluttered scenes," *Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1584–1601, 2006.

[5] R. Detry, N. Pugeault, and J. Piater, "A probabilistic framework for 3D visual object representation," *Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 10, pp. 1790–1803, 2009.

[6] W. Li and L. Kleeman, "Autonomous segmentation of near-symmetric objects through vision and robotic nudging," in *Int. Conf. Intelligent Robots and Syst.*, 2008, pp. 3604–3609.

[7] D. Beale, P. Iravani, and P. Hall, "Probabilistic models for robot-based object segmentation," *Robot. Auton. Syst.*, vol. 59, no. 12, pp. 1080–1089, 2011.

[8] H. van Hoof, O. Kroemer, and J. Peters, "Probabilistic interactive segmentation for anthropomorphic robots in cluttered environments," in *Proc. Int. Conf. Humanoid Robots*, 2013.

[9] J. Carreira and C. Sminchisescu, "CPMC: automatic object segmentation using constrained parametric mincuts," *Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1312–1328, 2012.

[10] J. Strom, A. Richardson, and E. Olson, "Graph-based segmentation for colored 3D laser point clouds," in *Int. Conf. Intelligent Robots and Syst.*, 2010, pp. 2131–2136.

[11] C. Erdogan, M. Paluri, and F. Dellaert, "Planar segmentation of RGBD images using fast linear fitting and Markov chain Monte Carlo," in *Conf. Comput. and Robot Vision*, 2012, pp. 32–39.

[12] T. Hermans, J. Rehg, and A. Bobick, "Guided pushing for object singulation," in *Int. Conf. Intelligent Robots and Syst.*, 2012, pp. 4783–4790.

[13] D. Katz, A. Orthey, and O. Brock, "Interactive perception of articulated objects," in *Int. Symp. Experimental Robotics*, 2010.

[14] J. Kenney, T. Buckley, and O. Brock, "Interactive segmentation for manipulation in unstructured environments," in *Int. Conf. Robotics and Automation*, 2009, pp. 1377–1382.

[15] P. Fitzpatrick and G. Metta, "Grounding vision through experimental manipulation," *Philos. T. Roy. Soc. A*, vol. 361, no. 1811, pp. 2165–2185, 2003.

[16] N. Bergström, C. Ek, M. Björkman, and D. Kragić, "Scene understanding through autonomous interactive perception," in *Int. Conf. Comput. Vision Syst.*, 2011, pp. 153–162.

[17] S. J. Pundlik and S. T. Birchfield, "Real-time motion segmentation of sparse feature points at any speed," *Trans. Syst., Man, and Cybern.*, vol. 38, no. 3, pp. 731–742, 2008.

[18] E. Herbst, X. Ren, and D. Fox, "RGB-D object discovery via multi-scene analysis," in *Int. Conf. Intelligent Robots and Syst.*, 2011, pp. 4850–4856.

[19] M. Cho, Y. M. Shin, and K. M. Lee, "Co-recognition of image pairs by data-driven monte carlo image exploration," in *European Conf. Comput. Vision*, 2008, pp. 144–157.

[20] C. Rother, V. Kolmogorov, T. Minka, and A. Blake, "Cosegmentation of image pairs by histogram matching – incorporating a global constraint into MRFs," in *Conf. Comput. Vision and Pattern Recognition*, 2006, pp. 993–1000.

[21] J. Modayil and B. Kuipers, "The initial development of object knowledge by a learning robot," *Robot. and Auton. Syst.*, vol. 56, no. 11, pp. 879–890, 2008.

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

[23] J. Sinapov, T. Bergquist, C. Schenck, U. Ohiri, S. Griffith, and A. Stoytchev, "Interactive object recognition using proprioceptive and auditory feedback," *Int. J. Robot. Res.*, vol. 30, no. 10, pp. 1250–1262, 2011.

[24] S. Griffith, J. Sinapov, M. Miller, and A. Stoytchev, "Toward interactive learning of object categories by a robot: A case study with container and non-container objects," in *Int. Conf. Develop. and Learning*, 2009, pp. 1–6.

[25] D. Schiebener, J. Morimoto, T. Asfour, and A. Ude, "Integrating visual perception and manipulation for au-

tonomous learning of object representations," *Adapt. Behav.*, vol. 21, no. 5, pp. 328–345, 2013.

[26] M. Krainin, B. Curless, and D. Fox, "Autonomous generation of complete 3D object models using next best view manipulation planning," in *Int. Conf. Robotics and Automation*, 2011, pp. 5031–5037.

[27] W. Li and L. Kleeman, "Interactive learning of visually symmetric objects," in *Int. Conf. Intelligent Robots and Syst.*, 2009, pp. 4751–4756.

[28] A. Ude, D. Omrčen, and G. Cheng, "Making object learning and recognition an active process," *Int. J. Hum. Robot.*, vol. 5, no. 2, pp. 267–286, 2008.

[29] M. Krainin, P. Henry, X. Ren, and D. Fox, "Manipulator and object tracking for in-hand 3D object modeling," *Int. J. Robot. Res.*, vol. 30, no. 11, pp. 1311–1327, 2011.

[30] L. Natale, F. Orabona, G. Metta, and G. Sandini, "Exploring the world through grasping: a developmental approach," in *Int. Symp. Computational Intell. in Robotics and Automation*, 2005, pp. 559–565.

[31] L. Chang, J. Smith, and D. Fox, "Interactive singulation of objects from a pile," in *Int. Conf. Robotics and Automation*, 2012.

[32] O. O. Sushkov and C. Sammut, "Feature segmentation for object recognition using robot manipulation," in *Australian Conf. Robotics and Automation*, 2011.

[33] K. Hausman, F. Balint-Benczedi, D. Pangercic, Z.-C. Marton, R. Ueda, K. Okada, and M. Beetz, "Tracking-based interactive segmentation of textureless objects," in *Int. Conf. Robotics and Automation*, 2013, pp. 1122–1129.

[34] M. Huber, T. Dencker, M. Roschani, and J. Beyerer, "Bayesian active object recognition via gaussian process regression," in *Int. Conf. Inform. Fusion*, 2012, pp. 1718–1725.

[35] J. Denzler and C. Brown, "Information theoretic sensor data selection for active object recognition and state estimation," *Trans. Pattern Anal. Mach. Intell.*, pp. 145–157, 2002.

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

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

[38] O. O. Sushkov and C. Sammut, "Active robot learning of object properties," in *Int. Conf. Intelligent Robots and Syst.*, 2012, pp. 2621–2628.

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

[40] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[41] P.-E. Forssén, "Maximally stable colour regions for recognition and matching," in *Conf. Comput. Vision and Pattern Recognition*, 2007, pp. 1–8.

[42] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[43] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, p. 13, 2006.

[44] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 6, pp. 721 –741, 1984.

[45] D. Aldous, "Exchangeability and related topics," in *École d'Été de Probabilités de Saint-Flour XIII - 1983*. Springer Berlin / Heidelberg, 1985, vol. 1117, pp. 1–198.

[46] H. van Hoof, O. Kroemer, H. Ben Amor, and J. Peters, "Maximally informative interaction learning for scene exploration," in *Int. Conf. Intelligent Robots and Syst.*, 2012, pp. 5152–5158.

[47] E. B. Fowlkes and C. Mallows, "A method for comparing two hierarchical clusterings," *J. Am. Stat. Assoc.*, vol. 78, no. 383, pp. 553–569, 1983.

Herke van Hoof obtained his Bachelor's and Master's degrees in Artificial Intelligence from the University of Groningen in 2008 and 2011, respectively.

He is a Ph.D. student at the institute for Intelligent Autonomous Systems at the Technische Universität Darmstadt, Germany, since 2011. He is interested machine-learning for robots in non-standardized environments with minimal human supervision. Among others, he works on sensory-motor coordination and autonomous exploration.

Oliver Kroemer obtained his Master's and Bachelor's degree in engineering from Cambridge University in 2008.

He joined the Intelligent Autonomous Systems institute at the Technische Universität Darmstadt, Germany, as a Ph.D. student in 2011. At TU Darmstadt, he continued the research he had been doing at the Max Planck Institute for Biological Cybernetics. He specializes in robots learning to grasp and manipulate objects.

Jan Peters is a full professor (W3) for Intelligent Autonomous Systems at the Computer Science Department of the Technische Universitaet Darmstadt and at the same time a senior research scientist and group leader at the Max-Planck Institute for Intelligent Systems, where he heads the interdepartmental Robot Learning Group. Jan Peters has received the Dick Volz Best 2007 US PhD Thesis Runner Up Award, the 2012 Robotics: Science & Systems - Early Career Spotlight, the 2013 Young Investigator Award of the International Neural Network Societys, and the 2013 IEEE Robotics & Automation Society's Early Career Award.