# Probabilistic Interactive Segmentation for Anthropomorphic Robots in Cluttered Environments

Herke van Hoof[1], Oliver Kroemer[1] and Jan Peters[1,2]

***Abstract*— Recognition and manipulation of novel objects in human environments are a prerequisite for many tasks of robots. Since objects often occur in clutter, such robots should be capable of segmenting their environment into individual objects before attempting to learn the objects' properties. In this paper, we propose a probabilistic part-based approach to interactive segmentation of cluttered scenes containing multiple novel objects. Our experiments show that our probabilistic approach outperforms commonly employed heuristics. Furthermore, the probability distribution over segmentations enables principled selection of informative actions.**

## I. Introduction

Human environments contain a wide range of different objects. Therefore, future humanoid robots performing tasks in such environments will require the ability to recognize and manipulate objects. Since novel objects may be encountered, relying on a fixed set of annotated training data is insufficient, as such a dataset cannot contain all objects the robot might encounter. Therefore, it is essential that robots learn about objects whenever they are encountered.

Many techniques for learning an object's appearance [1–4], require the object to be isolated from other objects or to be segmented a priori. In cluttered environments, however, object segmentations are not readily available. Some computer vision approaches, like [5], try to infer a segmentation from a single image using visual features such as contrast, texture, and color clues. When available, 3D-cues, such as surface normals, can also be integrated into such approaches [6]. The underlying assumption of such methods is that discontinuities in image features indicate object boundaries. This assumption does not necessarily hold in practice [7]. Hence, not all segmentation ambiguities can be resolved from a single image [8–12].

Another approach is to look at the differences between two or more images, or the movement in a video stream. Movement helps to resolve segmentation ambiguities [13–17]. Rather than waiting for movement in the environment, robots have the possibility to cause changes in the environment themselves [7–11, 18–21], as illustrated in Fig. 1. Such interactive strategies make robots more autonomous and allow them to ground the definition of objectness in terms of their own actions [9]. Additionally, knowing which actions was performed can help the robot to infer the correct segmentation [21].

Although co-movement provides strong clues for segmentation, there are multiple sources of uncertainty that need to be taken into account. Firstly, in cluttered scenes, adjacent objects might move together, so that co-movement might result even if parts belong to different objects. Additionally, only a part of a non-rigid object might move when pushed. Hence, absence of co-movement might occur even if parts belong to the same object. Furthermore, the movement of occluded parts is unknown, and noise and imperfections in the vision system introduce additional uncertainty.

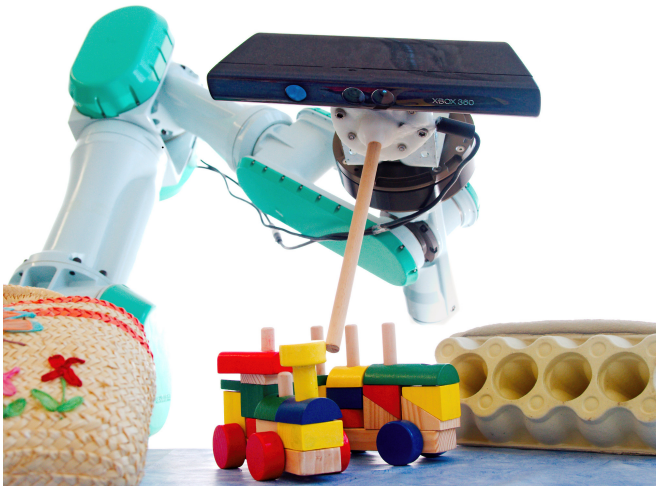

Fig. 1: Our robot arm autonomously interacts with its environment to segment a scene into objects.

Therefore, we require a principled way to represent *uncertainty in the segmentation*. Representing this uncertainty allows us to robustly plan the outcome of subsequent actions, and a measure of uncertainty can serve as a stopping criterion for the segmentation phase. The real world offers an abundance of interaction possibilities. Rather than trying all of these possibilities at random or exhaustively, it is desirable that our representation so far allows us to select actions targeted at *efficient exploration*.

### A. Contribution

In this paper, we propose a probabilistic approach to interactive segmentation that fulfills the aforementioned requirements. This approach enables us to deal with effects such as noisy observations and co-movement of objects in a principled way, using a minimal amount of assumptions and hand-tuning. Specifically, we do not assume objects are rigid bodies and avoid hand-tuned segmentation heuristics based on object appearance, compactness, or object shape.

To make this approach viable, we first reduce the problem

[1]TU Darmstadt, FB Informatik, FG IAS, Darmstadt, Germany
[2]Max Planck Institute for Intelligent Systems, Tübingen, Germany
{hoof, kroemer, peters}@ias.tu-darmstadt.de

by defining sub-object regions that can be tracked. How such regions are defined, modeled, and tracked is explained in Sec. II. In Sec. III, we define a probability distribution over segmentations obtained by merging these local models. We then show how, using the proposed representation, a robot can select explorative actions that maximally reduce its uncertainty. An overview of our approach is shown in Fig. 2. In Sec. IV, we evaluate our approach and compare it to two common heuristics. We present our conclusions in Sec. V.

### B. Related work

If an object occurs in different images, rather than segmenting using low-level visual features of a single image, the correspondence between images can be exploited in an approach known as co-segmentation [12, 14, 22] or co-recognition [15]. If the background is the same across the images, an alternative approach is to explicitly detect motion between them [17]. Instead of comparing a pair of images from before and after motion occurred, we might track objects during interaction to segment them [16]. However, occlusions caused by the robot arm limit the applicability of this approach to interactive robot set-ups.

In *interactive* segmentation approaches, the robot itself takes actions to cause movement in a scene, resulting in movement that provides information about the correct segmentation. If the workspace contains only a single object, it can be pushed by simply sweeping the arm across the workspace [11]. Alternatively, symmetry clues can be used to target more precise pushes [18]. Accumulating evidence over time leads to more precise segmentation [10].

Movement detection by image differencing, employed by the methods in the previous paragraph, requires a static background and textured objects. Some of these limitations can be addressed by estimating object membership per segment of the image, rather than per pixel [19, 21]. Alternatively, the iterative closest point algorithm can be used to find the number of objects that explains the movement in the scene [9]. Yet another possibility is using the movement of trackable features together with heuristic predictors based on appearance, compactness, and object shape to estimate the object membership of each of these features [7, 8, 20].

Another way to physically separate an object from background clutter is to grasp and subsequently lift it [23–27]. However, common reliable grasping methods usually require a model of the object. Alternatives such as learned grasping classifiers usually need the object to be segmented. Thus, these methods cannot easily be applied to cluttered scenes with novel objects.

Many of the discussed approaches deal with only one object of interest, e.g. [11, 18, 21, 22]. However, cluttered scenes contain multiple objects, that might move simultaneously in the same direction if they are adjacent. Such co-movement would lead approaches such as [10, 11] to conclude that just one object was present. Alternatively, multiple pushes could be performed, retaining only objects that consistently move as rigid bodies [8, 9, 15, 17]. These

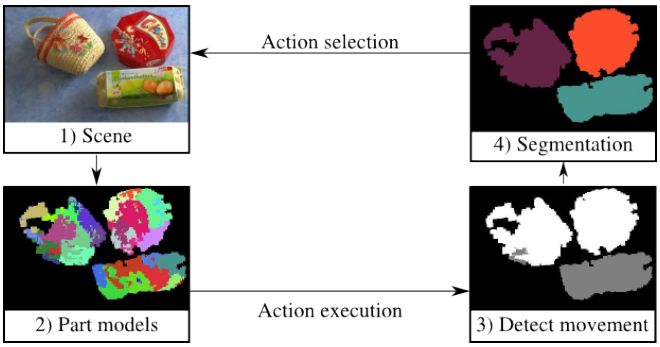


Fig. 2: Every scene is processed using our part-based approach: (1,2) Known parts are recognized using stored appearance characteristics and new parts are extracted from regions that have not been seen yet. (3) Movement of known parts (shown in grey) is detected based on the distance between their current and past locations. (4) From the detected movement, a distribution of segmentations is inferred (Sec. III). Targeted exploration can resolve ambiguities.

methods are inherently limited to rigid objects, and it is often unclear how to handle the uncertainty stemming from occlusions.

Several alternatives have been suggested: Multiple observations with possible conflicting information can be integrated by minimizing inconsistent movement [13]. However, this approach makes the common assumption that objects are connected components in the input image, which is not necessarily the case due to occlusions. The probabilistic approach in [21] quantifies the probability that moving segments belong to the target object based on the correlation between the object's movement and that of the robot arm, but considers only a single object of interest. To deal with non-rigid objects, Katz et al. [7] identify rigid segments and then identify rotational or prismatic joints between them. However, their approach cannot deal with non-rigid objects that are not explained by one of those two kind of joints, as for example, a bundle of clothing. Joulin et al. [12] take a probabilistic approach to multi-class segmentation using just visual features. Their approach requires that the number of classes is set by hand, and not every segment corresponds to a single object, i.e., visually similar instances of an object class, such as 'cars', are assigned to a single segment.

## II. Defining and Tracking Sub-Object Parts

The robot needs to represent the observed appearance characteristics of objects. These observed characteristics cannot be directly attributed to specific objects, because the segmentation of the scene into objects is (initially) unknown.

Instead, in our approach, appearance characteristics are attributed to local regions. Such regions (subsequently referred to as 'parts') are defined by a fixed radius (6 cm.) around the origin of a local coordinate frame. These local coordinate frames are initialized such that they cover all point cloud data. New parts can be added as necessary to cover regions of the scene that become visible during interaction. As points

can be within the radius of more than one part's center, parts can overlap.

### A. Part-based appearance model

In our part-based model, an object is recognized if the parts of that object are detected in the environment. To this end, all parts are described using *local key points* extracted from images of the scene from multiple viewpoints. Local key points are distinctive points with a local description that can reliably be detected in novel views. Considering only local key points, rather than all pixels, has the advantage of speeding up appearance matching between scenes. Different kinds of local key point detection and description algorithms exist. In this paper, we use the Scale Invariant Feature Transform (SIFT) algorithm [28] for this purpose. In case objects without texture are present, additional features such as the color regions in [20] could be used in addition.

The description and location of the key points, as well as the location and orientation of the parts' centers, are stored to recognize the identity and pose of the parts later on.

### B. Recognizing parts and detecting their movement

After the robot executes an action, it observes the effects of its action. To this end, parts are recognized in new scenes by matching their associated key points. Occasionally, false matches will occur, even with powerful descriptors. Although we do not assume *objects* are rigid bodies, we will assume the objects' *parts* are approximately rigid.

We therefore use the random sample consensus (RANSAC) algorithm [29] to robustly find the homogeneous transformation of the local coordinate frame that explains the majority of key point matches, even when false matches are present. In our experiments, 100 iterations of the RANSAC algorithm sufficed to find good matches.

If a homogeneous transformation with sufficient inliers is found, the local coordinate frame is transformed accordingly. If the corresponding translation is larger than a threshold of 3 cm, we conclude the part has moved. Looking for pure rotations in addition did not improve results. If no fitting transformation is found, we conclude that the part is not visible in the observation, for example due to occlusion of the object.

## III. Probabilistic segmentation

A segmentation of the scene is obtained by partitioning the set of extracted parts 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.

Seeing parts move together is more probable when those parts belong to the same objects [10, 11]. However, observing joint movement does not guarantee that the parts belong to the same object, as the robot's sensors are noisy and objects that push each other also result in joint movement. To represent the resulting uncertainty, we use a probabilistic approach. As data accumulates over time, the uncertainty reduces, resulting in distributions peaked around the true partition. An illustration of this process is provided in Fig. 3.

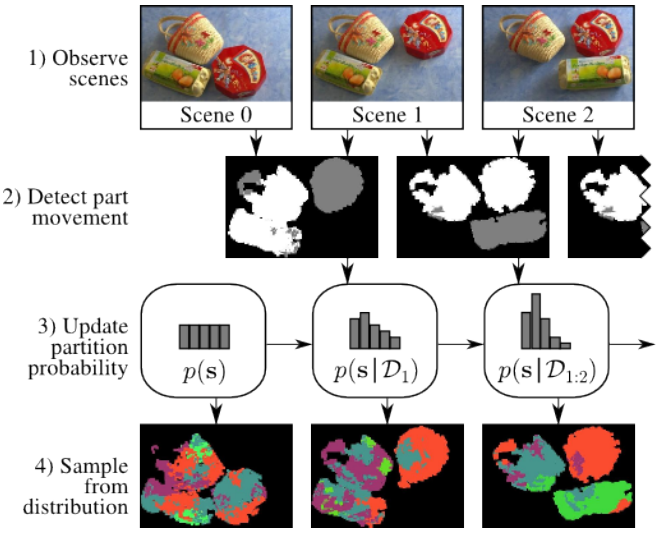


Fig. 3: Overview of our probabilistic segmentation approach. After every action, the resulting scene is observed (1) and moving (grey) and non-moving (white) parts are identified (2). This data $\mathcal{D}$ is then used to update the probability distribution over partitions $\mathbf{s}$ (3). This distribution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

### A. Probabilistic segmentation model and inference

We represent a partition of the set of $N$ parts into clusters (each corresponding to an object) by a vector $\mathbf{s} = [s_1, \ldots, s_N]^T$, with the variables $s_i$ indicating which object part $i$ belongs to. That is, the vectors $\mathbf{s} = [1, 1, 2]^T$ and $\mathbf{s} = [2, 2, 1]^T$ are equivalent as both indicate a partition where the first two parts constitute a single object, while the third part constitutes a different object.

The probability distribution over partitions $\mathbf{s}$ after $T$ actions is expressed as $p(\mathbf{s}|\mathcal{D})$, with $\mathcal{D}$ the data observed so far $\mathcal{D} = \{(a_t, \mathbf{o}_t)|t \in 0, 1, \ldots, T\}$. In this equation, $a_t$ is the index of the part targeted by the $t^{\text{th}}$ action and $\mathbf{o}_t$ is the resulting observation. Observation $\mathbf{o}_t[j]$ is equal to 1 if part $j$ was observed moving at time step $t$, or 0 if it was observed to be stationary.

For each hypothesized object $k$, we define latent variables $m_{k,t}$ that indicate whether the object moved at time $t$. The probability $p(m_{k,t} = 1)$, is assumed to be an unknown constant $\theta_{\text{p}}$ if that object was pushed, or $\theta_{\text{np}}$ if it was not pushed. Furthermore, part $i$ is assumed to be observed moving at time $t$ with an unknown, fixed probability $\theta_{\text{m}}$ if the object to which it belongs actually moved or $\theta_{\text{nm}}$ if it did not (to account for sensor noise). Generally, segmentations $\mathbf{s}$ that assign the same object to parts that move together

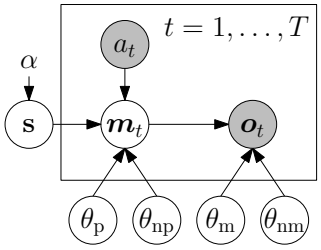


Fig. 4: Graphical model for segmentation. Shaded circles indicate observed variables.

are more probable, although these probabilities are learned from the data rather than pre-specified. Fig. 4 shows the structure of our graphical model. The free parameter $\alpha$ will be explained below.

The number of possible values for $\mathbf{s}$ and $\mathbf{m}$ grows exponentially with the number of parts. Hence, calculating the conditional probability for each segmentation quickly becomes computationally impracticable. Therefore, we approximate this distribution using samples drawn according to a Gibbs sampling scheme [30]. This approach produces samples from $p(\mathbf{s}, \mathbf{m}|\mathcal{D})$ by iteratively selecting one of the latent variables and reassigning that variable a value based on all other variables:

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

$$m_{k,t} \sim p(m_{k,t}|\mathcal{D}, \mathbf{s}, \mathbf{m}_{\setminus k,t}) \propto p(\mathbf{o}_t|\mathbf{m}_{1:T})p(\mathbf{m}_{1:T}|\mathbf{a}_t, \mathbf{s}),$$

with the notation $\mathbf{s}_{\setminus i}$ indicating the vector of all variables $s_j$ with $j \neq i$. If we keep only every $n^{\text{th}}$ sample we obtain independent samples for sufficiently large $n$.

We will first turn our attention to the prior $p(s_i, \mathbf{s}_{\setminus i}|\alpha)$. We do not assume to know how many objects there are in the scene. Hence, a suitable non-parametric prior distribution over partitionings $\mathbf{s}$ of $n$ parts is the Chinese restaurant process [31]. Given the assignment of the other parts $\mathbf{s}_{\setminus i}$, 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: $p(s_i = j|\mathbf{s}_{\setminus i}) = N_j/(\alpha + N - 1)$. However, the part can also be assigned to a new object $J$, to which no other parts have been assigned yet, with probability $p(s_i = J|\mathbf{s}_{\setminus i}) = \alpha/(\alpha + N - 1)$. The Chinese restaurant process is a non-parametric process that does not require the number of objects to be set in advance, allowing this number to be inferred from the data. The free parameter $\alpha$ controls how often new objects are created by the generative process. In our experiments, we used the (standard) setting of $\alpha = 1$. We found that inference is not very sensitive to $\alpha$, although when extreme values are used, more data is needed to infer the right number of objects.

The uncertainty in the parameters $\theta_\text{p}, \theta_\text{np}, \theta_\text{m}$ and $\theta_\text{nm}$ can be marginalized in closed form if we use the conjugate factorizing prior $p(\theta_\text{p}, \theta_\text{np}|\mathbf{s}) = \text{Beta}(\theta_\text{p}|\alpha_\text{p}, \beta_\text{p})\text{Beta}(\theta_\text{np}|\alpha_\text{np}, \beta_\text{np})$. We then obtain

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

with $\alpha'_\text{p}$, $\alpha'_\text{np}$ corresponding to $\alpha_\text{p}$, $\alpha_\text{np}$ plus the number of times an object moved given that it was pushed ($\alpha'_\text{p}$) or not pushed ($\alpha'_\text{np}$). Similarly, $\beta'_\text{p}$, $\beta'_\text{np}$ correspond to $\beta_\text{p}$, $\beta_\text{np}$ plus the number of times an object did not move given that it was pushed ($\beta'_\text{p}$) or not pushed ($\beta'_\text{np}$). We set uniform priors $\alpha_\text{p} = \beta_\text{np} = \alpha_\text{np} = \beta_\text{p} = 1$. Analogously, we define the observation model

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

with the parameters $\alpha'_\text{m}$, $\alpha'_\text{nm}$, $\beta'_\text{m}$, $\beta'_\text{nm}$ expressing the the number of times parts were observed moving or not moving given movement of the corresponding object, added to the respective uniform prior $\alpha'_\text{m} = \beta'_\text{nm} = \alpha'_\text{nm} = \beta'_\text{m} = 1$. Not all parts are observed at every time step, e.g. because of occlusions or because the part was added at a later time step. Such instances are ignored in these calculations.

The hyper parameters can be interpreted as 'pseudo-observations' for each event. Samples of prior and posterior segmentation of a test scene are shown in Fig. 5. Inference in our probabilistic model takes between one and seven seconds depending on the amount of data. Compared to the time needed to observe the scene and execute an action, this is a modest duration.

(a) Test scene to be segmented.

(b) True partitioning.

Prior samples (0 actions) | Posterior samples (5 actions) | Posterior samples (15 actions)

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

Fig. 5: The same test scene is used to show the belief state after exploring this set of objects in a similar training scene (not shown). A priori, the number of objects as well as the segmentation are unknown: the samples are blind guesses. Over time, the number of objects and the correct segmentation are inferred. The uncertainty decreases, which is visible from the growing consistency of the samples.

### B. Selecting maximally informative actions

Our approach approximates a probability distribution over segmentations regardless of the action selection strategy. However, if the robot deliberately performs pushes it expects to be most informative, segmentations might be obtained faster [19]. Hence, our robot chooses actions to maximize the mutual information $I(\mathbf{s};\mathbf{o}|a,\mathcal{D})$, where $\mathbf{s}$ is the partition of the parts into objects and $\mathbf{o}$ is the observed outcome of an action targeted at part $a$. The mutual information corresponds to the expected information gain of observing the result of pushing part $a$, and is given by

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

where $D_{\mathrm{KL}}$ is the Kullback-Leibler divergence. The argument of the logarithm is computed as

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

assuming $p(\mathbf{s}|a,\mathcal{D}) = p(\mathbf{s}|\mathcal{D})$, as shown in Fig. 4.

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

$$I(\mathbf{s};\mathbf{o}|a,\mathcal{D}) \approx \frac{1}{J}\sum_j \log\left(\frac{p(\mathbf{o}_j|\mathbf{s}_j,a,\mathcal{D})}{\frac{1}{K}\sum_k p(\mathbf{o}_j|\mathbf{s}_k,a,\mathcal{D})}\right). \quad (1)$$

Samples from $p(\mathbf{s}|\mathcal{D})$ can be obtained using the Gibbs sampling procedure described in Section III-A. To obtain samples from the joint $p(\mathbf{o},\mathbf{s}|a,\mathcal{D})$, we again sample $\mathbf{s}_j \sim p(\mathbf{s}|\mathcal{D})$ and separately sample $\mathbf{o}_j \sim p(\mathbf{o}|\mathbf{s}_j,a,\mathcal{D})$, where

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

We assume $p(\mathcal{D}|\mathbf{s}_j,a) = p(\mathcal{D}|\mathbf{s}_j)$, as shown in Fig. 4. In this equation, $p(\mathcal{D}|\mathbf{s}_k)$ is calculated using

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

using the conditional independence of $\mathcal{D}$ and approximating the expectation with samples $\mathbf{m}_j \sim p(\mathbf{m}|\mathbf{s}_k)$. To calculate $p(\mathbf{o},\mathcal{D}|\mathbf{s}_j,a)$, we simply treat the potential observation $\mathbf{o}$ as additional actual observations and use the same computation.

## IV. Experiments

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. Then, in Sec. IV-C we compare our probabilistic segmentation method to alternative methods on data gathered by a real robot. Finally, in Sec. IV-E, 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.

### A. Experimental set-up

We evaluated our approach using a 7 degrees of freedom Mitsubishi PA-10 robot arm. Objects relevant for the experiment were set up on a table next to the robot. 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 so that observations taken from different points of view were aligned in the robot's coordinate frame and observed parts not belonging to the scene on the table could automatically be removed.

Fig. 6: 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).

To learn object models, the robot was presented with a cluttered scene of novel objects taken from the set shown in Fig. 6. The robot interacted with the scene, pushing selected parts in a direction corresponding to the estimated surface normal. After every action, the scene was observed from three different view points in order to update both the individual part models (as described in Section II) and the distribution over partitions of the parts into objects (as described in Section III-A). The entire procedure is illustrated in Fig. 7.

### B. Comparisons

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. Our baselines are:

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

*2) Pairwise:* Two parts belong together if co-movement is observed more often than separate movement, independent of any other parts.

### C. Evaluation of interactive segmentation

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 probability that parts move given a certain push.

(a) Observation (b) Integration (c) Push (d) Observation

Fig. 7: 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 resultant scene is observed and the patch centers are registered as moving (green) or non-moving (red).

We evaluated the interactive segmentation algorithms on a scene with four objects. The robot explored these objects using fifteen random actions. The set of twelve different objects (Fig. 6) was used to create fifteen initial set-ups.

After every action, the robot updated its posterior probability distribution over segmentations. Parts that belong to the same object according to the ground truth (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 [32], we evaluated the partitions using the correspondence

$$B = \frac{|P \cap Q|}{\sqrt{|P||Q|}}, \tag{2}$$

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. The results are shown in Fig. 8 and Fig. 9. The same dataset was used for all methods.

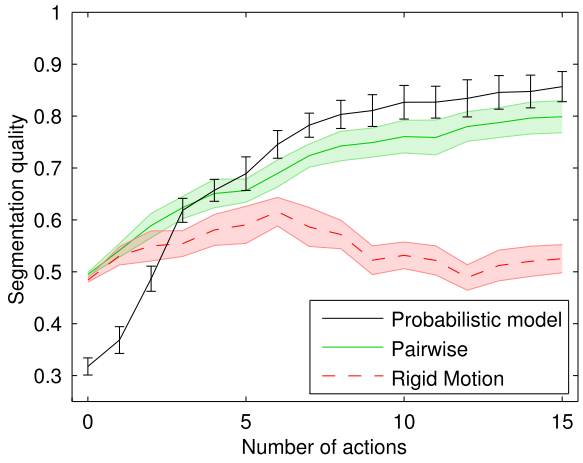


Fig. 8: Inference of scene segmentations using difference inference methods. Parameter learning increases initial uncertainty in the probabilistic model, which reduces segmentation quality. Error bars show the standard error.

### *D. Discussion of the segmentation task results*

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 needing an annotated training set or external feedback signal.

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 Fig. 8). Our approach attained an average quality of 0.86 in contrast to 0.80 for the 'pairwise' method. This difference is a significant step towards 'perfect' segmentations (1.00). After 15 actions all methods except for the 'rigid motion' method seem to have converged. Our probabilistic approach needed only 9 actions for half of the trials to attain a segmentation quality of at least 0.85, while this quality is not reached within 15 actions for the 'pairwise' and 'rigid motion' methods (see Fig. 9).

Parameters of our probabilistic model ($\theta_{\mathrm{p}}, \theta_{\mathrm{np}}, \theta_{\mathrm{m}}$ and $\theta_{\mathrm{nm}}$) are learned rather than tuned which increases uncertainty in the beginning of the experiment. This uncertainty reduces performance of our learning method relative to manually specified methods initially. The baseline methods, on the other hand, do not manage to take optimal advantage of

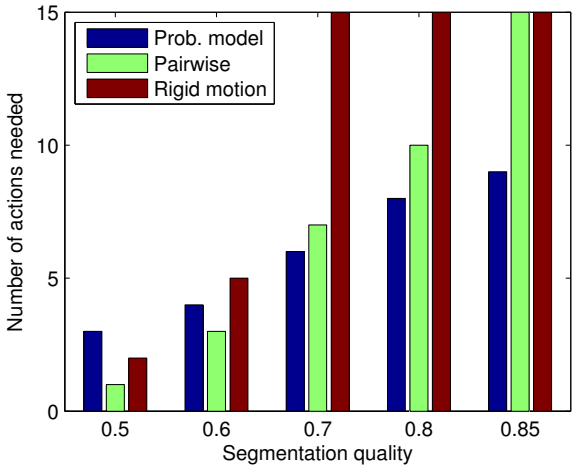


Fig. 9: Inference of scene segmentation using different methods: number of actions needed for the majority of trials to reach a given accuracy. Since the experiments lasted for 15 actions, 15 is the maximum number shown even if the segmentation quality was not reached within this number.

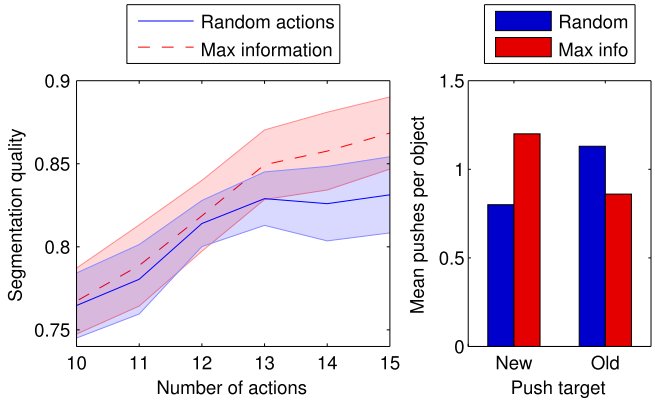


Fig. 10: Action selection experiment. On the left, the segmentation quality is shown starting from the tenth action, when two objects are moved inside the robot's workspace. The right graph shows the mean number of pushes directed at existing and newly introduced objects during this period.

a larger dataset. One cause for occasional failure in all of the methods was that the tracker system occasionally loses certain parts.

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.

### *E. 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 at a later point in time.

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. 6, 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 so that they became reachable to the robot.

Then, five more actions were executed using one of two action selection strategies: selecting actions at random or according to the maximal mutual information criterion explained in section III-B. To speed up computation, we set the parameters $\theta_{\text{np}}, \theta_{\text{p}}, \theta_{\text{m}}, \theta_{\text{nm}}$ 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. We expect the mutual information criterion to focus exploration at the objects that it has not yet explored when they are moved inside the workspace, thereby improving the robot's knowledge faster. The results are shown in Fig. 10.

### *F. Evaluation of the action selection experiment*

After the first 10 actions, the three objects initially in the workspace have been explored using random actions. When the two initially out-of-reach objects are introduced into the workspace, the action selection strategy employing the mutual information criterion focuses explorative actions on these objects, quickly improving the segmentation quality. Random actions are still divided over all five objects in the robot's workspace, leading to slower improvement for this baseline.

## V. Conclusion

In this paper, we have introduced our approach for probabilistic segmentation of cluttered scenes. This approach allows the robot to learn about novel objects in cluttered environments, even in the presence of noise or uncertainty. It retains a probability distribution over segmentations. This distribution could be helpful for subsequent tasks such as planning for different possible outcomes and the selection of robust actions. Furthermore, by representing the remaining uncertainty, the robot can direct exploration to reduce this uncertainty. Reaching a low uncertainty could also serve as a stopping criterion for the segmentation phase, allowing the robot to start exploring other properties.

We avoid hand-tuning important parameters, such as the number of objects and the probability of movement given a push, by learning them from the observed effects of interacting with the objects. Furthermore, we do not hand-code assumptions about object shapes, appearance, or compactness. In fact, in our experiments, only the movement resulting

from actions was used to infer segmentations, and not the visual similarity between parts. Using learning techniques to incorporate such visual clues is subject of ongoing work.

We evaluated the quality of the segmentation found by a real robot as it obtained experience through interaction with its environment. Two heuristics for motion segmentation were outperformed by our method. We suggested a maximum mutual information criterion as a principled way of directing exploration. In a separate experiment, we showed that when objects are not consistently reachable, this criterion improves the robot's learning speed.

## Acknowledgment

The project receives funding from the European Community's Seventh Framework Programme under grant agreements no. ICT-248273 GeRT and no. ICT-270327 Complacs.

## References

[1] A. Mian, M. Bennamoun, and R. Owens, "Three-dimensional model-based object recognition and segmentation in cluttered scenes," *PAMI*, vol. 28, 2006.

[2] H. Murase and S. Nayar, "Visual learning and recognition of 3-D objects from appearance," *IJCV*, vol. 14, 1995.

[3] A. Collet, D. Berenson, S. Srinivasa, and D. Ferguson, "Object recognition and full pose registration from a single image for robotic manipulation," in *ICRA*, 2009.

[4] R. Detry, N. Pugeault, and J. Piater, "A probabilistic framework for 3D visual object representation," *PAMI*, vol. 31, no. 10, 2009.

[5] J. Carreira and C. Sminchisescu, "CPMC: automatic object segmentation using constrained parameteric mincuts," *PAMI*, vol. 34, no. 7, 2012.

[6] J. Strom, A. Richardson, and E. Olson, "Graph-based segmentation for colored 3D laser point clouds," in *IROS*, 2010.

[7] D. Katz, A. Orthey, and O. Brock, "Interactive perception of articulated objects," in *ISER*, 2010.

[8] N. Bergström, C. Ek, M. Björkman, and D. Kragić, "Scene understanding through autonomous interactive perception," in *ICVS*, 2011.

[9] T. Hermans, J. Rehg, and A. Bobick, "Guided pushing for object singulation," in *IROS*, 2012.

[10] J. Kenney, T. Buckley, and O. Brock, "Interactive segmentation for manipulation in unstructured environments," in *ICRA*, 2009.

[11] P. Fitzpatrick and G. Metta, "Grounding vision through experimental manipulation," *Philosophical Transactions of the Royal Society of London, Series A*, 2003.

[12] A. Joulin, F. Bach, and J. Ponce, "Multi-class cosegmentation," in *CVPR*, 2012.

[13] E. Herbst, X. Ren, and D. Fox, "RGB-D object discovery via multi-scene analysis," in *IROS*, 2011.

[14] C. Rother, V. Kolmogorov, T. Minka, and A. Blake, "Cosegmentation of image pairs by histogram matching – incorporating a global constraint into MRFs," in *CVPR*, 2006.

[15] M. Cho, Y. M. Shin, and K. M. Lee, "Co-recognition of image pairs by data-driven Monte Carlo image exploration," in *ECCV*, 2008.

[16] S. J. Pundlik and S. T. Birchfield, "Real-time motion segmentation of sparse feature points at any speed," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 3, 2008.

[17] E. Herbst, X. Ren, and D. Fox, "Object segmentation from motion with dense feature matching," in *ICRA Workshop on Semantic Perception, Mapping and Exploration*, 2012.

[18] W. Li and L. Kleeman, "Autonomous segmentation of near-symmetric objects through vision and robotic nudging," in *IROS*, 2008.

[19] H. van Hoof, O. Kroemer, H. Ben Amor, and J. Peters, "Maximally informative interaction learning for scene exploration," in *IROS*, 2012.

[20] A. Ude, D. Schiebener, N. Sugimoto, and J. Morimoto, "Integrating surface–based hypotheses and manipulation for autonomous segmentation and learning of object representations," in *ICRA*, 2012.

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

[22] D. Hochbaum and V. Sing, "An efficient algorithm for co-segmentation," in *ICCV*, 2009.

[23] A. Ude, D. Omrčen, and G. Cheng, "Making object learning and recognition an active process," *Int. Journal of Humanoid Robotics*, vol. 5, no. 2, 2008.

[24] D. Kraft, R. Detry, N. Pugealt, E. Başeski, F. Guerin, J. Piater, and N. Krüger, "Development of object and grasping knowledge by robot exploration," *IEEE Trans. on Autonomous Mental Development*, vol. 2, 2010.

[25] M. Krainin, P. Henry, X. Ren, and D. Fox, "Manipulator and object tracking for in-hand 3D object modeling," *Int. Journal of Robotics Research*, vol. 30, no. 11, 2011.

[26] W. Li and L. Kleeman, "Interactive learning of visually symmetric objects," in *IROS*, 2009.

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

[28] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, 2004.

[29] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, 1981.

[30] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *PAMI*, vol. 6, no. 6, 1984.

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

[32] E. B. Fowlkes and C. Mallows, "A method for comparing two hierarchical clusterings," *Journal of the American Statistical Association*, vol. 78, no. 383, 1983.