Generating Object Hypotheses in Natural Scenes through Human-Robot Interaction

Niklas Bergström
nbergst@kth.se

Mårten Björkman
celle@csc.kth.se

Danica Kragic
danik@csc.kth.se

Abstract—We propose a method for interactive modeling of objects and object relations based on real-time segmentation of video sequences. In interaction with a human, the robot can perform multi-object segmentation through principled modeling of physical constraints. The key contribution is an efficient multi-labeling framework, that allows object modeling and disambiguation in natural scenes. Object modeling and labeling is done in a real-time, to which hypotheses and constraints denoting relations between objects can be added incrementally. Through instructions such as key presses or spoken words, a scene can be segmented in regions corresponding to multiple physical objects. The approach solves some of the difficult problems related to disambiguation of objects merged due to their direct physical contact. Results show that even a limited set of simple interactions with a human operator can substantially improve segmentation results.

I. INTRODUCTION

How can robots learn about object concepts and their relations in natural scenes without complex prior models? Robots should be able to interact with humans in a natural manner, share and discuss concepts such as objects, places, activities. In terms of visual input, we need to develop methods that go beyond pure geometrical or appearance modeling and that provide some element of interpretation. Robots inhabiting the human world and interacting with people should also be able to learn from and in interaction with humans. The field of Human-Augmented Mapping (HAM) aims at building maps and labeling places by creating correspondences between how the robot and the human perceive the world, [1]. While in this case the large-scale concepts such as rooms and doorways are learnt, it is equally important for the robot to understand and model small-scale scenes: the existence, placement and relations between objects.

Thus, when faced with a new scene such as a table-top or a refrigerator, the robot needs to create hypotheses about objects. Whether the purpose is manipulation, learning a representation or describing the scene, a partitioning of the scene is necessary. Both inclusion of background or other objects, as well as only capturing a part of an object, might lead to wrong parts being grasped, or non relevant structures or only some structures being learnt. Without extensive experience and when faced with new instances or categories of objects, the robot is likely to fail its task. Thus, we want our robot to i) resolve mistakes and ambiguities

Fig. 1. The figure shows a scenario where a robot looks at a table, generates object hypotheses and displays the result to the user for confirmation. At www.csc.kth.se/~nbergst there are videos showing the system.

in generating object hypotheses through interaction with a human, ii) draw from the experience of a human interacting with objects. This encompasses a set of theoretical and technical requirements with scene segmentation methods in focus. While classical image segmentation aims to partition an image into meaningful regions, we aim at leveraging on 2D segmentation for performing 3D segmentation necessary for generating hypotheses of objects. Herein lies one of the fundamental challenges: *What constitutes an object?*

In this paper we take a step toward a *cognitive vision* framework: we present a segmentation method that creates a representation of a scene based on the objects contained in it, adapts this representation based on monitoring movements in the scene, and incorporates information provided by a human about the objects' geometric relations. Fig. 1 shows an abstraction of the system. The first step is to generate correct hypotheses of objects. In this case robot has grouped two objects in one segment, and asks a human for confirmation that, in its turn, helps the robot to resolve the problem. We build on our method for *object segmentation and tracking* system presented in [2], [3], and extend it to i) handle multiple hypotheses, and ii) leverage on the cues from a user regarding the number of objects and their relations.

After outlining the contributions and the related work, we first introduce the new multi-object segmentation framework in Sec. III and initialization strategies in Sec. IV. In Sec. V and Sec. VI, we introduce constraints modeling and evaluate the system. Finally, conclusions and future work are discussed in Sec. VII.

II. CONTRIBUTIONS AND RELATED WORK

At the heart of the proposed system, there is a theoretical model for 2D/3D segmentation performed in real time. The

segmentation literature presents a large variety of approaches to attack the problem of grouping pixels in a coherent way. Purely data driven approaches have in common that the concept of an object does not exist [4], [5], [6]. The resulting segmentation will be a partitioning of the image into pieces that are similar in color and position, and there is no guarantee that segments correspond to physical objects.

Object segmentation methods aim to separate one object in the foreground from the background. Contrary to the ones mentioned above, these methods require initialization to select *which* object that should be segmented. This is often done by letting a human mark directly in the image by drawing scribbles on the object and the background [7] or drawing a rectangle around the object [8]. If additional information, assumptions or cues, are provided, even a single point can be enough for initialization. This can be done by e.g. assuming self-similarity within objects [9] or using a log-polar transform of the image with respect to that point in combination with weighting with binocular disparities [10]. Similarly, our method only requires one point, which makes it suited for robots relying on attentional mechanisms, [11].

Interactive segmentation with user feedback is not a new fad. However, contrary to work where interaction is an integral part of the system, e.g. [7], [8], we are not restricted to a keyboard-mouse interface, which would not have been suitable for a robotic scenario. Instead, like in [12] we want the robot to handle spoken cues. In [13] the authors demonstrated a system where the segmentation could be improved with voice commands, an interface that is easily integratable with our method. In addition, in [7], [8] the interaction leads to hard assignments of pixels to either segment. We take a different approach where the aim is to provide a *minimum amount* of information by the user, and to provide more information if necessary. In our case, user commands affect how the data is *modeled*, rather than the data itself.

In recent work, contextual information has been used to divide a scene into semantic regions [14] or as a mean to segment out an unknown object from its context [15]. In these cases only outdoor scenes are considered, where structures like sky and buildings occur predictably in the images. In an indoor close up scene, like a table top scenario, similar structures do not occur. Therefore, we here take a fundamentally different approach. By acknowledging that, in order to produce reliable results, significant assumptions have to be made, we present an efficient and principled segmentation framework capable of exploiting and incorporating prior knowledge and assumptions, and to adapt the segmentation online as new knowledge becomes available.

This leads us to the two main contributions presented in this paper: (1) *a principled and efficient multi-labeled framework*, (2) *exploiting user feedback for object segmentation and disambiguation*. Since our model runs in real-time, can incorporate prior knowledge and assumptions of a general and flexible form, we exploit its computational benefits in order to incorporate a user feedback loop. Fig. 2 shows four steps of an interactive scenario, where a human gives

Fig. 2. A segmentation scenario. Without the requirement of a mouse, foreground segments are inserted through four sequential commands: *Add object*, *Split object*, *Add object* and *Add object*. Best viewed in color.

instructions to the robot.

The proposed method is based on an MRF formulation of segmentation. It adopts the idea of iterative segmentation and model optimization from [8], using an EM-framework. However, unlike many other methods for interactive image segmentation [7], [8], [16], the solution is found using belief propagation (BP), rather than graph cuts. The benefits for us of using BP is twofold: (1) While graph cuts does not scale well with the number of hypotheses, BP scales linearly in computational time. (2) One single EM iteration is very fast (around 15 ms), why the method can incorporate new from a moving camera or a changing scene in real time.

III. SEGMENTATION FRAMEWORK

In [2] we presented a method for segmenting one object from a background given a single point on that object, and in [3] we explored the tracking aspects of the same method. In this section, we describe the main parts of the algorithm in order to clarify the extensions made in this paper.

Using color, position and disparity information, the method proved to successfully segment single objects in scenarios with considerable clutter. The method benefits from scenes where objects are supported by a flat surface by including a planar hypothesis in addition to the hypotheses of figure and ground. This helps differentiating on object from the surface it is placed on, in cases when colors are similar.

While disparity information helps segmenting heterogeneously colored objects, it sometimes will lead to that more than one objects are included in the same foreground segment. This is not strange since, looking from a bottom-up perspective, there is no way of telling if two objects are standing next to each other, or if it is in fact only one object.

In this paper we extended the framework to handle multiple foreground hypotheses and integration of information provided by a human instructor. Such information can for instance be provided as spoken cues as demonstrated in [17], [13]. In addition, we allow for any combination of hypotheses. An important difference to our previous work is that the positional measurements are modeled in terms of

Fig. 3. A live sequence of an entering hand moving a can. The can is found with object search and hand with motion detection.

their projections in 2D image and disparity space. The reason is that since only the frontal side of objects can be seen, a 3D Gaussian would be a poor representation of object shape, while a combination of two projections will assume objects with a volume. Furthermore, if certain pixels lack disparities, due to the non-textured areas or occlusions, the disparity cue can simply be ignored for these pixels.

We first give an overview of the optimization framework and then describe extensions for hypotheses constraints.

A. Bottom-up segmentation with EM

The segmentation problem involves a set of models of foreground objects and background clutter. Given a parameter set θ , representing these models, the problem can be expressed as a Markov random field (MRF) where each pixel has a measurement $m_i \in \mathcal{M}$ and the task is to assign each pixel to a label $l_i \in \mathcal{L}$, denoting from which model the measurement was generated.

Since the labels are latent variables, maximizing $p(\theta, \mathbf{l}|\mathbf{m})$ would not take into account the uncertainties involved with the labeling. Therefore we instead maximize $p(\theta|\mathbf{m}) \propto p(\mathbf{m}|\theta)p(\theta)$, where $\mathbf{m}$ is the set of measurements and $\mathbf{l}$ their corresponding labels. With respect to the MAP solution, $p(\mathbf{m})$ is constant and can be omitted. The distribution can thus be expressed as a marginalization over the labels,

$$p(\mathbf{m}|\theta)p(\theta) = \sum_{\mathbf{l}} p(\mathbf{m}, \mathbf{l}|\theta)p(\theta). \quad (1)$$

Typically it is assumed that a measurement m_i only depends on its corresponding label l_i . If it is further assumed that dependencies between labels are only modeled between pairs of pixels and are independent on θ , the joint distribution can be expressed as

$$p(\mathbf{m}, \mathbf{l}|\theta) = p(\mathbf{m}|\mathbf{l}, \theta)p(\mathbf{l}) = \prod_i p(m_i|l_i, \theta)p(\mathbf{l}) \quad (2)$$

$$\text{where } p(\mathbf{l}) = \frac{1}{Z} \prod_i p(l_i) \prod_{(j,k) \in \mathcal{N}} \psi(l_j, l_k).$$

Here $\psi(l_j, l_k)$ represents the second order cliques, and the other factors the first order cliques. Potts model is used for $\psi(l_j, l_k)$, similar to [18], [19]. $\mathcal{N}$ denotes the set of all second order cliques in the MRF.

$$Z = \sum_{\mathbf{l}} \left[\prod_i p(l_i) \prod_{(j,k) \in \mathcal{N}} \psi(l_j, l_k) \right] \quad (3)$$

is a normalization constant that is needed as $\psi(l_j, l_k)$ is not a probability distribution. $p(l_i)$ are constant parameters.

A MAP solution to $p(\theta|\mathbf{m})$ can be found using Expectation Maximization (EM). This algorithm finds the parameter set $\hat{\theta}$ that maximizes $\log p(\theta|\mathbf{m})$ with respect to θ , which is equivalent to a maximization of $p(\theta|\mathbf{m})$. This is done iteratively by updating a lower bound $G(\theta) = \mathcal{Q}(\theta, \theta') + \log p(\theta) \leq \log p(\mathbf{m}|\theta) + \log p(\theta)$, where

$$\mathcal{Q}(\theta, \theta') = \sum_{\mathbf{l}} p(\mathbf{l}|\mathbf{m}, \theta') \log p(\mathbf{m}, \mathbf{l}|\theta) \quad (4)$$

depends on a current parameter estimate θ' . The algorithm alternates between two steps until convergence, given an initial parameter set θ^0 . In the *expectation* step, the distribution over labelings, $p(\mathbf{l}|\mathbf{m}, \theta')$, is evaluated given the measurements and current parameter estimate. Then, in the *maximization* step, a new set of parameters is sought:

$$\theta^{new} = \arg \max_{\theta} G(\theta) \quad (5)$$

The maximization step involves computing a sum over all possible labelings, see Eq. 4. Since labels are assumed to be dependent, the summation is done over $|\mathcal{L}|^N$ terms, where N is the number of measurements, which becomes intractable in practice. Fortunately, with the two assumptions made in Eq. 2, the problem is simplified. Since the factor $p(\mathbf{l})$ is independent on θ , it will not have any influence on the maximization step in Eq. 5. Thus only the factors $p(m_i|l_i, \theta)$ are of importance, and $\mathcal{Q}(\theta, \theta')$ can be replaced by

$$\begin{aligned} \mathcal{Q}'(\theta, \theta') &= \sum_{\mathbf{l}} \sum_i p(\mathbf{l}|\mathbf{m}, \theta') \log p(m_i|l_i, \theta) = \\ &= \sum_i \sum_{l_i \in \mathcal{L}} \left(\sum_{\mathbf{l} \setminus l_i} p(\mathbf{l}|\mathbf{m}, \theta') \right) \log p(m_i|l_i, \theta) = \\ &= \sum_i \sum_{l_i \in \mathcal{L}} p(l_i|\mathbf{m}, \theta') \log p(m_i|l_i, \theta). \end{aligned} \quad (6)$$

Here the summation is done over only $N|\mathcal{L}|$ terms and the marginals $p(l_i|\mathbf{m}, \theta')$ can be computed using loopy belief propagation [20]. Thus contrary to what is claimed in [2], no approximations have to be introduced in order to make the problem tractable.

B. Measurements and models

Like in our previous work [2], the scene consists of foreground $\mathbf{F}$, planar $\mathbf{P}$, and clutter $\mathbf{C}$ models with the addition that we now allow multiple instances of each model. The models are described by a set of parameters $\theta = \{\theta_f, \theta_c, \theta_p\}$. We refer the reader to [2] for their components and their usage in Eq. 2, and describe here the differences with respect to the foreground model used in this work.

$\theta_f = \{p_f, \Delta_f, c_f\}$, where p_f, Δ_f denote the mean 3D position and variance of a foreground hypothesis, while c_f describe its color distribution. As noted above we in this work describe a foreground model with image coordinates and disparities separated:

$$p(m_i|\theta_f) = \mathcal{N}(p'_i|p'_f, \Delta'_f) \mathcal{N}(p''_i|p''_f, \Delta''_f) H_f(h_i, s_i) \quad (7)$$

Here p'_i denotes a projection of p_i down to the 2D image space and p''_i a projection to 1D disparity space. We use projections instead of simply dividing p_i into (x_i, y_i) and d_i as we use other projections in Sec. V-B.

C. Model priors

For the maximization step of the EM-algorithm in Eq. 5, a prior distribution over the parameters $p(\theta)$, is needed. In case of no prior knowledge or constraints, a uniform

distribution is assumed and the MAP solution is reduced to a maximum likelihood (ML) solution. We seek this solution when segmenting single images with no prior information.

However, if information is available, such as a known object color, it can be introduced as priors. We exploit similar priors for image sequences, in order to get more consistent segmentations over time. Priors are given as a distribution $p(\theta|\theta^t)$. Here θ^t are the parameters estimated in the previous image of the sequence. The distribution is expressed as $p(\theta|\theta^t) = p(\theta_f|\theta_f^t) p(\theta_c|\theta_c^t) p(\theta_p|\theta_p^t)$, where

$$p(\theta_c|\theta_c^t) = \mathcal{N}(d_c|d_c^t, \Lambda_c) g(\Delta_c|\Delta_c^t, S_c) \mathcal{N}(c_c|c_c^t, \sigma_c^2 I)$$

is the prior for the clutter model. Priors for the other types can be expressed correspondingly. Here Λ_c and σ_c are fixed parameters indicating the expected variances over time for respective parameters, whereas S_c governs how the covariance Δ_c is updated. We adopt the definition of function $g()$ from [2]. Other priors can also be applied for inter-hypothesis constraints as shown in Sec. V-A.

IV. ONLINE HYPOTHESES INSERTION

In [3] we moved our Armar III active robot head to the image center and created a foreground segment from there. With multiple foreground segments, we here take a different approach (Sec IV-A).

Foreground hypotheses are sequentially added to the segmentation framework, that initially contains only planar hypotheses and a single clutter one. Planar hypotheses are introduced through random sampling [21] of triplets of points in (x, y, d) -space. A sampled planar hypothesis is accepted and inserted, if the support among not yet assigned points is large enough and the absolute majority of the remaining points is on the side of the plane facing the camera. All remaining unassigned points are then assigned to the clutter hypothesis.

A. Foreground insertion

Foreground hypotheses are created from the clutter points, using either object search, splitting or motion detection, procedures that are all triggered by a human operator through the commands *Add object*, *Split object* and *Detect motion*. Object search is performed by first finding the densest clutter of points in 3D metric space, using a mean-shift operation [5]. Since points closer to the camera are more densely distributed, there is a natural bias towards nearby objects, such that these are found first. For image sequences or live streams, motion detection by background subtraction is applied for the same purpose. The sequence in Fig. 3 shows a few frames from a live motion detection scenario.

For both object search and motion detection a foreground hypothesis is inserted, if the supported number of points is larger than a threshold, defined by a given expected object size. To prevent points from neighboring regions to disturb the initial foreground model, only points within a small 3D ball (one fourth of the object size) are initially assigned to the hypothesis. These serve as seeds that will grow through connectivity in 3D space, once the EM loop is restarted.

In the beginning of the loop the disparity cue tends to be dominating, while the color cue is able to fill the gaps for those points that lack disparities.

B. Hypotheses splitting

If two objects are in close proximity, they are likely to be encapsulated by the same segment. By issuing the command *Split object*, an operator may allow the segmentation to recover by introducing another foreground hypothesis. Given only the information that a segment covers two objects, a likely split of that segment must be found. This is done by clustering the pixels within the segment using k-means based on position, disparity, hue and saturation, and fit a Gaussian in (x, y, d) space to each of the clusters. Then the object modeling loop is resumed with the new hypotheses, using either the mean of the Gaussians as seeds, or with all the pixels from respective cluster. We refer to the two reinitialization types as *point* and *cluster* initialization, and compare the two types in Sec. VI.

Since k-means clustering is initiated with random centers, it could happen that it fails to correctly divide the segment into two different objects. In this case the user needs to provide additional information. In the following section we will use relational priors to deal with this problem.

V. ONLINE MODEL ADAPTATION

It might not be enough to just add an extra foreground model. As discussed in the previous section, k-means might fail to separate the two objects properly, resulting in faulty segmentations. One way to solve this is to guide the splitting process, using the relative positions between objects. We do this through three commands: (1) *Split object <relation>*, (2) *Split object <relation> with prior* and (3) *Split object <relation> with constraints*. Here *<relation>* is either (A) *sideways*, (B) *height wise* or (C) *depth wise*.

The splitting directions have to be done relative to some frame of reference. The x-axis of will always be given by the camera. However, if a gravitation vector is known, or if the normal $(\alpha, \beta, -1)$ of a large supporting planar hypothesis is available, then these will define the y-axis. If not, the y-axis will be assumed parallel to that of the camera's. With the first command (1) k-means clustering, as described in Sec. IV-B, will be overridden and splitting done in the given direction. The two other commands will be discussed in following sections, (2) in Sec. V-A and (3) in Sec. V-B.

A. Relational priors

In Fig. 4(a) the situation is not resolved by issuing the command *Split object sideways*, as seen in Fig. 4(b). However, if the command *Split object sideways with prior* is used, the model is changed to affect the foreground parameters' priors, resulting in the segmentation shown in Fig. 4(c).

We model the relative position of the objects as the difference between their two means, $p_\Delta = p_{f_1} - p_{f_2}$, where $p_{f_1} \in \theta_{f_1}$ and $p_{f_2} \in \theta_{f_2}$ are members of their corresponding parameter sets. Depending on how the objects are related

(a) One foreground (b) Two foregrounds (c) FGs with prior

(d) One foreground (e) Two foregrounds (f) FGs with constraint

Fig. 4. Two examples of scenes using relational priors, (a)-(c), and model constraints, (d)-(f).

to each other, we constrain p_{Δ} using a prior $p(p_{\Delta}|\Lambda_r)$ and change the prior $p(\theta)$ discussed in Sec. III-C accordingly:

$$p(\theta_{f_1}, \theta_{f_2}) = p(\theta_{f_1})p(\theta_{f_2})p(p_{\Delta}|\Lambda_r) \quad (8)$$

Here $p(p_{\Delta}|\Lambda_r) = \mathcal{N}(\theta_f^r|0, \Lambda_r)$ is a Gaussian distribution with zero mean and the covariance matrix Λ_r controls how much p_{f_1} and p_{f_2} can move with respect to each other. If the two objects are standing next to each other, like in Fig. 4(a), we allow p_{Δ} to have a large variation in the x -direction, while it should be constrained in the y -, and d -directions. The prior will act as a resistance on the evolution of the objects in the EM-algorithm, making it harder for one segment to grow under or above the other, which happened in Fig. 4(c).

B. Constraining the model

Another problem that might occur is that objects close to each other are separated in the proper direction, but that one segment encapsulates parts of both objects. By issuing a *Split object <relation> with constraint* command, we can enforce a constraint on the foreground model. Fig. 4(d) gives an example where giving the *Split object height wise* command results in parts of the bottom object is encapsulated by the top segment, Fig. 4(e).

We assume that objects are separated by a virtual plane. Thus if we observe the objects from an angle parallel to that plane, the objects will be separated. Fig. 5(a) displays the problem with the unmodified method in the example above. Since only the frontal sides of the objects can be seen, the Gaussians are not aligned with the objects. However, if we change the coordinate system according to Fig. 5(b), we restrict the variances to better capture the individual objects, resulting in a better segmentation, Fig. 4(f).

(a) Original model

(b) Rotated coordinate frame

Fig. 5. Side view of the scene in Fig. 4(d). Visible parts are marked in blue and Gaussians in $y-d$ -directions are marked in red. In 5(b) the coordinate frame is parallel to the surface and the Gaussians are forced to evolve in the objects' directions.

A change in coordinate system is done by modifying the projections of Eq. 7. We now set $p'_i = M'p_i$ and $p''_i = M''p_i$. M' is a 2×3 , and M'' a 1×3 projection matrix. The projection axes are found by first defining a coordinate system in metric space and then transforming these back to (x, y, d)

space. The metric axes are given by $n_y = (\alpha, \beta, \delta/f)$ and two axes on the plane $n_z = (0, -\delta/f, \beta)$ and $n_x = n_y \times n_z$, where f is the focal length. α, β, δ are parameters describing the surface of the planar model $\mathbf{P}$. The transformations are done relative to the mean position of all foreground points. Note that different foreground models can simultaneously be observed in different projections and that pixels with no disparities are always measured in the original frame in terms of their image coordinates.

VI. EXPERIMENTAL EVALUATION

A hand-segmented database of 100 images containing pairs of objects of different complexity and backgrounds was created, in order to evaluate the presented method with its modifications and compare it to alternative approaches. Each scene contains two objects placed either *next-to*, *behind*, or *on-top* of each other. They are placed close enough for at least some parts to touch. Examples of scenes can be found in Fig. 6. To test in a structured way how the method handles two objects in close proximity, we focus our testing on two foreground hypotheses.

The testing procedure is as follows: For each of the ground truth examples, a point inside one of the two objects is found using object search as explain in Sec. IV-A. Segmentation is then done using only one foreground hypothesis (method *1FG*). Note that the ground truth images were created such that this stage almost always fails. For cases in which a segment covers both objects, a new object hypothesis is added using splitting (method *2FG*). This knowledge assumes a human operator that points out the fact that a segment needs to be divided. Splitting is further tested using relational constraints (method *2FG+dir*), and by adding relational priors and model constraints (method *2FG+constr.*), as explained in Sec. V. Splitting directions and choice of method are information also provided by an operator, through the commands mentioned above. Table I shows the precision, recall and F1 scores for respective test.

1FG - One foreground hypothesis: The precision score is rather low, due to fact that the segment often spreads to both objects. However, the recall score is high, indicating that the method usually manages to fully capture both objects with the segment. This is important when attempting to split the segment, since the splitting can then be limited to an area covering both objects, but without background.

2FG - Two foreground hypotheses: Given that *1FG* fails to segment only one object, we try to automatically resolve the situation by issuing a *Split object* command. Tab. I shows the score for point and cluster initialization (explained in Sec. V). The slightly lower performance of the

Fig. 6. Six examples with segmentations. Column 1: Results with a single foreground hypothesis. Column 2: Segmentation from k-means clustering, with two foreground hypotheses (column 3). Column 4: Split using positional relations and corresponding segmentation, with two foreground hypotheses (column 5). Column 6: Two foreground hypotheses with constraints. In the first example, k-means fails to provide good enough clustering, and the segmentation splits the objects in the wrong direction. However, it is enough to provide relational information to the guide splitting. In the remaining examples this is not enough. However, examples 2-3 are helped with relational priors (Sec. V-A), while examples 4-6 are helped by constraining the model (Sec. V-B).

	Init	Precision	Recall	F1
(1) <i>1FG</i>		0.494	0.884	0.625
(2) <i>2FG</i>	point	0.908	0.847	0.869
(2) <i>2FG</i>	clust.	0.869	0.814	0.834
(3) <i>2FG+dir</i>	point	0.942	0.877	0.901
(3) <i>2FG+dir</i>	clust.	0.922	0.865	0.885
(4) <i>2FG+constr.</i>	point	0.952	0.892	0.914
(4) <i>2FG+constr.</i>	clust.	0.914	0.863	0.881

TABLE I

EVALUATION OF THE DIFFERENT METHODS FOR BOTH CLUSTER AND POINT INITIALIZATION.

	Precision	Recall	F1
Mishra - best	0.931	0.634	0.631
Mishra - mean	0.631	0.323	0.338

TABLE II

EVALUATION OF THE METHOD OF MISHRA *et al.* [10].

latter indicates that it might be better to let the segments evolve on their own, rather than giving a large bias from the start and having the risk of getting stuck in local minima. Since the resulting foreground segments are not ordered, we

compare with the ground truth segment that gives the best score. Fig. 6 shows some segmentations and results from the k-means clustering.

2FG+dir - Two hypotheses with direction: Here we identify each example as being of one of three kinds: *next-to*, *behind* and *on-top*. Given this information, splitting is guided to its corresponding axis. The results are significantly better compared to method *2FG*. This is due to examples where completely wrong segmentation using unguided k-means splitting, like the first example in Fig. 6, gets corrected given a proper direction. Again we conclude that point initialization outperforms cluster initialization.

2FG+const. - Two hypotheses with constraints: If splitting is constrained with relational priors and model constraints as described in Sec. IV-B, we observe additional improvements for point initialization, while cluster initialization gave slightly worse results. It should be noted that the examples that benefit from constraints were only 25% of the complete set, but for these examples such as the last five in Fig. 6, improvements can be significant.

Alternatives: In addition we evaluate how the method in [10] compares to our method. This method is similar to the method used in this paper as it also utilizes disparity to improve segmentation and takes a single points as initialization. It is more restricted though, as it builds the segmentation around a polar representation of the image generated from the initialization point, and can thus not easily handle multiple foreground hypotheses. The method relies heavily on an edge map computed with [22], and results are largely dependent on where the initialization point is placed. We evaluate the method as follows: For each ground truth segment for each image, we generate ten segmentations by randomly picking initialization points. We then take the best and mean scores from these ten segments, and average over all images and ground truth segments. The results can be found in Table II. Noticeable is that the method gives a high precision, indicating that the segments only to a small extent spills over on the background or other objects. On the other hand, even in the best case the recall is low, indicating that the method often get stuck in local structures.

VII. CONCLUSIONS AND FUTURE WORK

Robots capable of performing cognitive tasks are on a research agenda for many of us. An important step toward achieving such systems, is to equip them with a capability to generate and share human-like concepts - where objects are one natural example of these. In this paper we presented a system that is able to generate hypotheses of previously unseen objects in realistic scenarios. We have shown how simple interaction with humans can help to resolve problems that the current state-of-the-art approaches are not able to give meaningful results. At the heart of the approach, there is a method for semi-supervised figure-ground segmentation capable of generating and monitoring multiple foreground hypotheses.

By introducing and modeling additional constraints, such as number of objects and objects' physical relations, the method shows an excellent performance and clear contribution with respect to the other existing methods. These constraints help in difficult situations where the basic theoretical method fails, e.g. mixing parts of one object with the other. We have evaluated the method in a setting where it is likely to cover two nearby objects when using one foreground hypothesis. Our experiments show that while the method copes well with this setting by just adding one foreground hypothesis, there were significant improvements by introducing information on the objects' positional relations.

So, what is next? The question arises as to what challenges in the area of cognitive vision and cognitive robotics in general would be most empowering to robotics if they were solved. What and where is the need for significant advancement that will fundamentally increase the competence of human-interacting robots? The challenges presented in this paper have proven to be rather difficult, perhaps because the context of situated robots increases the complexity of the task and the variability of human behavior changes the output goal. This why it is necessary to continue to study

systems like this in the context of realistic tasks and realistic scenarios.

REFERENCES

- [1] H. Zender, Óscar Martínez Mozos, P. Jensfelt, G.-J. M. Kruijff, and W. Burgard, "Conceptual spatial representations for indoor mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 493–502, June 2008.
- [2] M. Björkman and D. Kragic, "Active 3d scene segmentation and detection of unknown objects," in *ICRA*, 2010.
- [3] —, "Active 3d segmentation through fixation of previously unseen objects," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2010, pp. 119.1–119.11.
- [4] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, August 2000.
- [5] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, 2002.
- [6] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, 2004.
- [7] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, 2001.
- [8] C. Rother, V. Kolmogorov, and A. Blake, "'grabcut': interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.
- [9] S. Bagon, O. Boiman, and M. Irani, "What is a good image segment? a unified approach to segment extraction," in *ECCV*, 2008.
- [10] A. Mishra and Y. Aloimonos, "Active segmentation with fixation," in *ICCV*, 2009.
- [11] G. Kootstra, N. Bergström, and D. Kragic, "Using symmetry to select fixation points for segmentation," in *International Conference on Pattern Recognition (ICPR)*, 2010.
- [12] G.-J. M. Kruijff, H. Zender, P. Jensfelt, and H. I. Christensen, "Clarification dialogues in human-augmented mapping," in *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, ser. HRI '06. New York, NY, USA: ACM, 2006, pp. 282–289. [Online]. Available: <http://doi.acm.org/10.1145/1121241.1121290>
- [13] M. Johnson-Roberson, G. Skantze, J. Bohg, J. Gustafson, C. R., and D. Kragic, "Enhanced visual scene understanding through human-robot dialog," in *2010 AAAI Fall Symposium on Dialog with Robots*, November 2010.
- [14] A. Vezhnevets and J. M. Buhmann, "Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning," in *CVPR*, 2010.
- [15] Y. J. Lee and K. Grauman, "Collect-cut: Segmentation with top-down cues discovered in multi-object images," in *CVPR*, 2010.
- [16] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, "Lazy snapping," *ACM Trans. Graph.*, vol. 23, pp. 303–308, August 2004.
- [17] D. Skocaj, M. Janicek, M. Kristan, G. Kruijff, A. Leonardis, P. Lison, A. Vrecko, and M. Zillich, "A basic cognitive system for interactive continuous learning of visual concepts," in *ICAIR workshop*, 2010.
- [18] R. Potts, "Some generalized order-disorder transformation," *Proc. Cambridge Philosophical Society*, vol. 48, pp. 106–109, 1952.
- [19] D. Geman, S. Geman, C. Graffigne, and P. Dong, "Boundary detection by constrained optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 609–628, July 1990.
- [20] Y. Weiss, "Correctness of local probability propagation in graphical models with loops," *Neural Comput.*, vol. 12, no. 1, pp. 1–41, 2000.
- [21] M. A. Fischler and R. C. 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, pp. 381–395, 1981.
- [22] J. M. D. Martin, C. Fowlkes, "Learning to detect natural image boundaries using local brightness, color and texture cues," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 5, pp. 530–549, 2004.