Enhanced Visual Scene Understanding through Human-Robot Dialog

Matthew Johnson-Roberson, Jeannette Bohg, Gabriel Skantze,
Joakim Gustafson, Rolf Carlson, Babak Rasolzadeh and Danica Kragic

***Abstract*— We propose a novel human-robot-interaction framework for the purpose of robust visual scene understanding. Without *any* a-priori knowledge about the scene structure, the task of the robot is to correctly enumerate how many objects there are in the scene and segment them. Our approach builds on top of state-of-the-art computer vision, segmenting stereo reconstructed point clouds into object hypotheses. This process is combined with a natural dialog system. By putting a 'human in the loop' and exploiting the natural conversation of an advanced dialog system, the robot gains knowledge about ambiguous situations beyond its own resolution. Specifically, we are introducing an entropy-based system allowing the robot to detect the poorest object hypotheses and query the user for arbitration. Based on the information obtained from the human-to-robot dialog, the scene segmentation can be re-seeded and thereby improved. We present experimental results on real data that show an improved segmentation performance compared to segmentation without interaction.**

I. Introduction

Current robots are capable of autonomously completing many objectives that are challenging both in perception and manipulation. Recent examples are towel folding [1], unloading a dishwasher [2] or preparing the ingredients for a cheese'n'ham omelet [3]. However, autonomous behavior is still only possible under many assumptions and within a controlled environment. A key challenge in robotics is to relax previously made assumptions and thereby enable a robot to act in new situations and handle increased uncertainty.

One way to achieve this objective is to put a 'human in the loop' and to allow the robot to learn from him or her. The specific problem that we are considering in this paper is scene understanding in which the robot has to correctly enumerate how many objects there are in the scene and segment them accurately. This capability can ease several different subtasks like recognition [4], grasp planning [5] or learning models for previously unseen objects re-usable for later re-recognition. Furthermore, by introducing a 'human in the loop', new labels or symbols can be introduced online and grounded in the current percept.

The paper at hand moves towards this goal by combining state-of-the-art computer vision with a natural dialog system. In our previous work [6] an embodied robotic system aims to segment a scene into several objects by exploring it with an active vision system. Dependent on how close objects are to each other or how similar they are in appearance, some of them might be incorrectly grouped together or split into several parts. The dialog system allows a human to rapidly refine the model of a complex 3D scene. Through that approach, we harness the strengths of 'bottom up' automated processing with the 'top down' decision making power of a human operator. The resulting refined scene model forms the basis for a general symbol grounding problem [7].

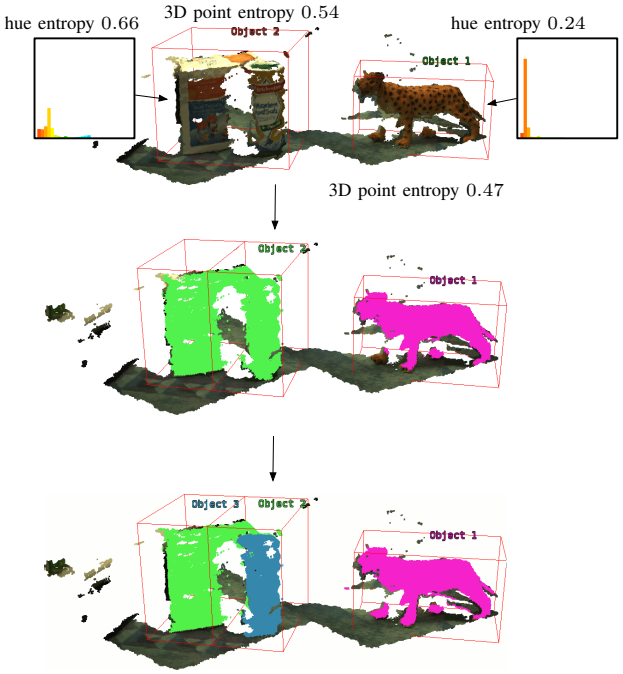


Fig. 1. Depiction of the different stages of the scene understanding. Top: Point cloud from stereo matching with bounding boxes (BBs) around segmented objects and their hue histograms. Left segment containing two objects has a higher entropy in hue and 3D point distribution and is therefore considered more uncertain. Middle: Initial labeling of the segments. Left segment is re-seeded by splitting BB in two parts based on human dialog input. Bottom: Re-segmented objects. Three objects are correctly detected.

An example of the scene refinement process is shown in Fig. 1. Given an initial scene segmentation provided by the vision system, we first identify the most uncertain object hypotheses through an entropy-based approach. The human operator is queried to provide information about whether this initial segment contains one or more objects and how those objects are positioned with respect to one another. If the initial segmentation is incorrect, the process is run again with new seed points respecting the user's input.

This paper is organized as follows. The next section dis-

This work was supported by the EU through the project GRASP, IST-FP7-IP-215821, the Swedish Foundation for Strategic Research and the Swedish Research Council. M. Johnson-Roberson, J. Bohg, B. Rasolzadeh and D. Kragic are with the Centre for Autonomous Systems and Computational Vision and Active Perception Lab, mattjr,bohg,babak2,danik@csc.kth.se. G. Skantze, J. Gustafson and R. Carlson are with the Department for Speech Music and Hearing, gabriel,jocke,rolf@csc.kth.se. The institutes are part of the School of Computer Science, KTH in Stockholm, Sweden.

cusses related work and provides an overview of the system. Afterwards, the vision module is described followed by the dialog system. Section V describes the interaction logic and entropy based hypothesis system. Finally, Section VI shows several instances of interactions and the resulting improved scene labelings.

II. Related Work and Contributions

Interactive segmentation methods have gained a lot of attention recently. One approach addressing this problem, is to let the robot interact with the scene, e.g., through pushing movements and then use motion cues for verifying or refining object hypotheses [8], [9], [10]. These papers focus on the question of what knowledge can be gained from observing the outcomes of robotic actions. An open question is how to select the action that provides the greatest amount of information.

This paper presents a vision system where, through the use of dialog, the robot gains knowledge about ambiguous situations beyond its own resolution. Human input is not a new concept in object segmentation and scene understanding. Several previous systems have used it to guide automated processing or restrict the search space of algorithms both improving efficiency and decreasing false positives [11], [12], [13]. *GrabCut* [14] and *Lazy Snapping* [15] are probably the most related works on interactive segmentation. They require the user to operate a mouse either for selecting a region containing the object in the image or for drawing its rough outline. In this paper, we aim for human-robot interaction using only natural dialog and no additional tools. Thereby, we are moving closer to how humans naturally interact with one another in such situation.

Several unique challenges exist when designing a framework for human robot interaction using computer vision. One of them is to minimize the iteration time of the vision component of such a system. This processing must be rapid enough as to not induce a large lag in the system increasing the mental load on the user [16]. As most state-of-the-art vision algorithm for scene understanding run in tens of minutes or hours per scene [17], [18], [19] they would be unacceptable for any 'human-in-the-loop' system. This paper proposes the use of a novel graph-cuts based multi-object segmentation algorithm [6], which is capable of providing acceptable responsiveness in the proposed algorithm.

Another design challenge is to minimize the necessary interaction between the autonomous system and the human operator. This means maximizing the value of each interaction to obtain the greatest discriminatory information. A very common approach to this kind of problem is to make a decision that maximises the expected gain in information or, equivalently, minimises the uncertainty in the system [11], [20]. In this paper, we use the concept of entropy to characterize the quality of any object hypothesis. This helps in guiding the dialog system to resolve the most challenging and ambiguous segments first.

System Overview

Our robotic platform is the 7-joint Armar III robotic head [21], as depicted in Fig. 2. The stereo head carries four Point Grey Dragonfly cameras grouped in two pairs, a peripheral and a foveal one. These are parts of an existing vision system [22] that uses attention in the peripheral view to direct cameras towards nearby regions of interest. After gaze re-direction, such regions are placed in fixation in the foveal view. Binocular disparities are calculated and a 3D point cloud reconstructed. The resulting point cloud is then clustered by performing an initial segmentation that groups points with similar color traits [6].

The dialog system allows a human operator to provide responses to the robot's questions in a natural manner. The robot can be interrupted and corrected in mid-utterance allowing the operator to avoid the latencies associated with traditional dialog call and response conversation patterns.

The main contributions of this paper lie in the scene analysis module that seeks to bridge the vision and dialog module. It fulfills two tasks: (i) it determines areas of the scene that are the poorest object hypotheses and seek human arbitration, (ii) it translates the human input to a re-seeding of the segmentation process. An overview of the system and the interconnection between its modules is shown in Fig. 2.

III. Vision System

The initial segmentation of the point cloud into object hypotheses and background is performed using a Markov Random Field (MRF) graphical model framework. This paradigm allows for the identification of multiple object hypotheses simultaneously and is described in full detail in [6]. Here, we will only give a brief overview.

As described in the previous section, the active humanoid head uses saliency to direct gaze. The same salient fixation points serve as seed points that we project into the point cloud to create initial clusters for the generation of object hypotheses.

For full segmentation we perform energy minimization in a multi-label MRF. We use the multi-way cut framework as proposed in [23]. An MRF is a graph with two sets of costs for assigning a specific label to a node: unary costs and pairwise costs.

In our case, the unary cost describes the likelihood of membership to an object hypothesis' color distribution. This distribution is modelled by Gaussian Mixture Models (GMMs) as utilized in GrabCuts [14]. For each salient region one GMM is created to model the color properties of that object hypothesis.

Pairwise costs enforce smoothness between adjacent labels. The pairwise structure of the graph is derived from a KD-tree neighborhood search directly on the point cloud. The 3D structure provides the links between points and enforces neighbor consistency. Once the pairwise and unary costs are computed, the energy minimization can be performed using standard methods. The α-expansion algorithm with available implementation [24], [25], [26] efficiently

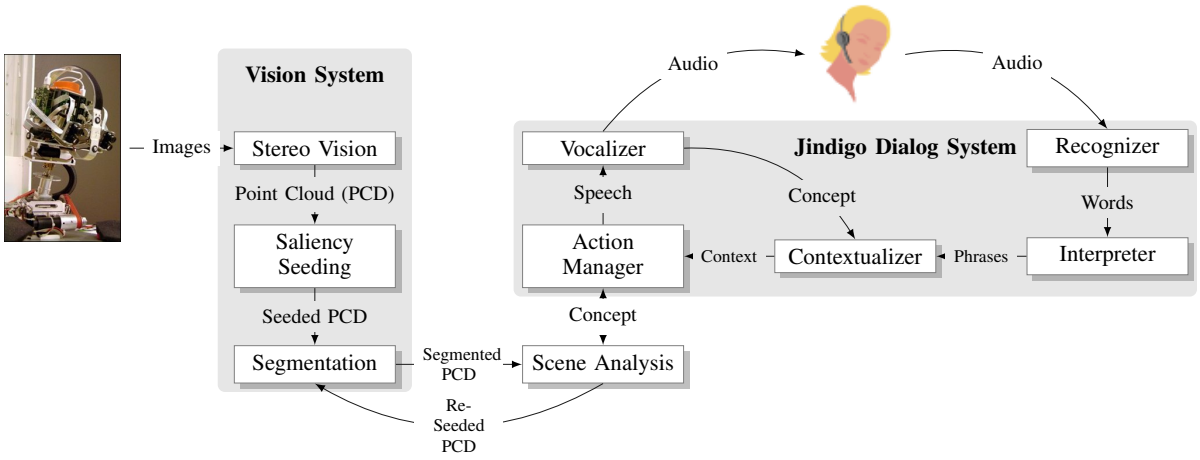


Fig. 2. Overview of the System. Left: Armar III Active Stereo Head [21]. Middle: Vision System that obtains stereo images from the cameras and outputs a segmented point cloud. Right: Architecture of the Jindigo Dialog System. Vision and Dialog System are communicating through a Scene Analysis Module.

computes an approximate solution that approaches the NP-hard global solution.

For each resulting segment, we compute several different attributes that will be used by the dialog system to refer to these object hypotheses. As attributes we choose the most dominant color, the position of the segment's centroid in the world coordinate system and the volume of the aligned bounding box.

IV. Dialog System

To facilitate interaction in the proposed system, we implement a dialog system as shown in Fig. 2. It is based on Jindigo – a Java-based open source dialog framework [27]. A typical limitation for dialog systems is that the dialog proceeds with strict turn-taking between user and system. The minimal unit of processing in such systems is the "utterance", which is processed in whole by each module of the system before it is handed on to the next. Contrary to this, Jindigo processes the dialog incrementally, word by word, allowing the system to respond more quickly and to give and receive feedback in the middle of utterances [28].

For speech recognition (Recognizer), we use CMU Sphinx 4 [29] which has been adapted to act as a Jindigo module. The words are parsed by an Interpreter [30] which tries to find an optimal sequence of matching phrases from a predefined grammar. These understood phrases are then re-interpreted by a Contextualizer [31], based on the history of the conversation. For example, fragmentary utterances like "red" and "yeah", and referring expression like "it", can be interpreted given the previous utterances from both the robot and the user. The contextualized interpretation of the user's utterance is then sent to the Action Manager (AM). The task of the AM is to decide which step to take next. To make these decisions, the AM communicates directly with the Scene Analysis (SA) component. If the AM or SA decides that the robot should say something, the semantic representation of this utterance is sent to the Vocalizer. The Vocalizer transforms this semantic structure to a textual representation, using the same domain specific grammars that are used by the Interpreter. The textual representation is then synthesized using MaryTTS [32]. As the utterance is spoken, the semantic representation is sent back to the Contextualizer for system self-monitoring. This way, the next user utterance may be interpreted in light of the last robot utterance.

Jindigo provides a model for semantic representation using typed concepts and arguments. For example, there may be a concept `Red` which inherits from the concept `Color` which in turn inherits from the base class `Concept`. Each concept may have a number of typed arguments, which are pointers to other concepts.

V. Refining the Scene Model through Human-Robot Interaction

From the visual scene segmentation as described in Section III, we obtain an initial bottom-up analysis of the scene resulting in several object hypotheses.

The quality of the initial segmentation can vary due to occlusions, objects that are very close to each other or when the position and number of seed points does not correspond to the actual objects. The demonstrated system was configured to handle the most common configurations and given a corresponding vocabulary for verbal disambiguation. However, we will show later how our system can easily be extended to deal with more complex cases.

In the following sections, we will show how user utterances are interpreted and how natural language is generated using the example dialog shown in Fig. 4. This dialog is an instantiation of the general dialog loop shown in Fig. 3. Furthermore in Section V-B and V-C, we propose a method to (i) rank the object hypotheses in terms of their uncertainty and (ii) re-segment the incorrect hypotheses.

A. Contextual Interpretation and Generation of Natural Language

In a natural dialog about the structure of a scene, objects may be referred to in different ways. One way is to use their properties (e.g, *the red object* or *the leftmost object*). To be able to ground these descriptions into the actual objects that are being talked about, the dialog system must have a

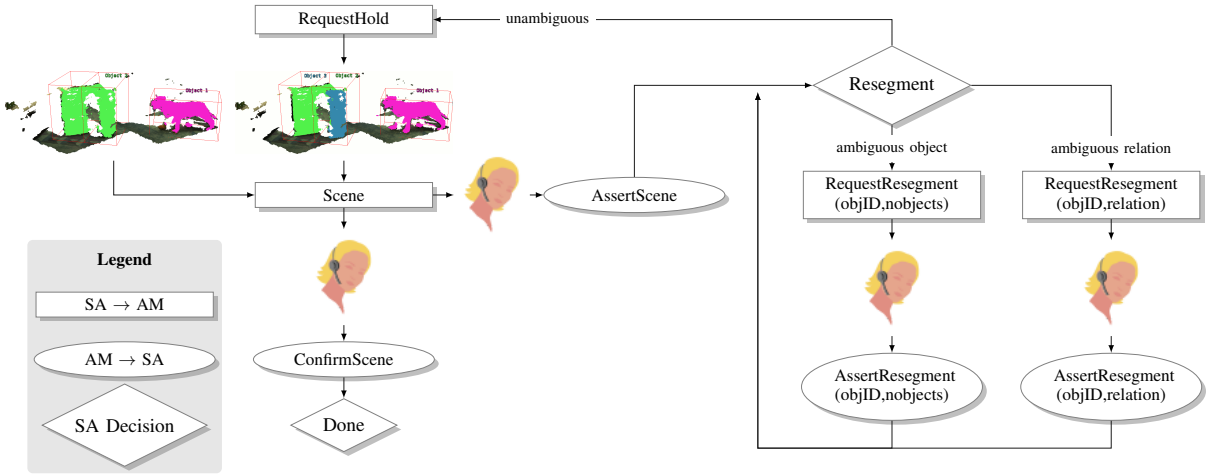


Fig. 3. Graph of the dialog loop showing the messages being passed between the action manager (AM) and the scene analysis (SA) module.

representation of the robot's hypothesis of the set of objects. Therefore, the initial segmentation of the scene is sent from the SA component to the AM, including the properties of objects and their IDs (see Fig. 4, Row 1).

Another way of referring to objects in dialog is by anaphoric expressions. For example a speaker may say *it* or *the segment* to refer to an object that has just been mentioned. A dialog system must be able to both understand and generate such expressions with the help of the dialog context. Another phenomena where the dialog context needs to be taken into account is elliptical (or fragmentary) expressions. These are expressions that lack propositional content, such as *the red one* or *to the left*. The dialog system needs to resolve these into full propositions, so called ellipsis resolution (e.g., *the red one* should perhaps be understood as *the red segment should be split into two objects* given the dialog context).

When the AM receives the initial scene, it starts to describe the scene, allowing the user to correct anything that is incorrect. It starts by stating the number of objects using the `CheckScene` action (Fig. 4, Row 2). This is transformed into text and speech by the Vocalizer (Row 3). Since the robot's hypothesis is incorrect, the user now corrects the robot. This utterance is unambiguous and therefore sent as it is all the way to the SA (Row 5). Notice how the semantic representation in Row 2 and 5 and the textual representation in Row 3 and 4 reflect each other. This illustrates how large parts of the domain grammars may be used for both interpretation and generation.

The SA requests help with refining the segment that is most likely to be incorrect (see the next section for how this choice is made). As can be seen in the example, the SA only refers to objects by their IDs. The AM then chooses the best way of referring to the object. It may for example be by the color (as in Row 7: *the green segment*) or by using an anaphoric expression (as in Row 14: *the objects*) depending on the current dialog history. The user now replies with an elliptical (fragmentary) expression (Row 9). Note that this utterance cannot be fully understood unless the dialog context is taken into account. This is precisely what the Contextualizer does - it re-interprets the ellipsis into a full proposition (Row 11). Thus, the `RequestResegment` from the robot combined with the `Object` from the user is transformed into an `AssertResegment` with this `Object` as argument. The same principles are applied in the ellipsis resolution in Row 18 (but where the spatial relation is replaced instead of the object).

As the scene is refined, the SA sends new scene hypotheses to the AM (Row 22). Finally, it will receive a `ConfirmScene` message from the AM (Row 27).

B. Identifying Poor Object Hypotheses through Entropy

To bootstrap the process of scene model refinement, the robot should be able to identify the most ambiguous object hypotheses and query the user about these first. We propose an entropy-based system to rank the object hypotheses according to their uncertainty. In general, the entropy $H[P(X)]$ of a distribution over a discrete random variable X with x_j as all M possible values is defined as follows:

$$H[P(X)] = -\sum_{j=0}^{M} P(x_j) \ln P(x_j). \quad (1)$$

and equivalently for a continuous variable with the integral. Intuitively, this means that sharply peaked distributions will have a low entropy while rather flat distribution will have a relatively high entropy. The maximum entropy is reached with a uniform distribution and is computed as $H_{max}[P(X)] = \ln 1/M$.

Our observation is that single objects tend to be relatively homogeneous in an appropriate attribute space. Consider for example a correctly segmented single object and a segment in which two objects are contained as in Fig. 1. The hue histogram of the correct hypothesis will show a narrower distribution than for the segment containing two objects. The same holds for the distribution of 3D points which will vary more in the second segment.

To test the hypothesis of entropy being a good indicator for the quality of object segmentation, we formalise it as binary classification problem. In this paper, we compute the entropy for each initial object hypothesis over a number of different attribute histograms. Let N be the set of points in

1	SA → AM	`Scene(objects:[ Object(id:32 color:Green ...) Object(id:47 color:Yellow ...)])`
2	AM→Voc	`CheckScene(Scene(nobjects:2))`
3	Voc→User	"I think there are two objects"
4	User→Rec	"No there are three objects"
5	Int→Con→AM→SA	`AssertScene(Scene(nobjects:3))`
6	SA→AM	`RequestResegment( object:Object(id:32) nobjects:2)`
7	AM→Voc	`RequestResegment( object:Object(color:Green) nobjects:2)`
8	Voc→User	"So, should I split the green segment?"
9	User→Rec	"No, the yellow one!"
10	Int→Cont	`NegFeedback Object(color:Yellow)`
11	Cont→AM	`AssertResegment( object:Object(color:Yellow) nobjects:2))`
12	AM→SA	`AssertResegment( object:Object(id:47) nobjects:2)`
13	SA→AM	`RequestResegment( object:Object(id:47) relation:TopOf)`
14	AM→Voc	`RequestResegment( object:Object(ref:Anaphora) relation:TopOf))`
15	Voc→User	"Are the objects on top of each other?"
16	User→Rec	"No, next to each other"
17	Int→Cont	`NegFeedback NextTo`
18	Cont→AM	`AssertResegment( object:Object(ref:Anaphora) relation:NextTo)`
19	AM→SA	`AssertResegment( object:Object(id:47) relation:NextTo)`
20	SA→AM→Voc	`RequestHold`
21	Voc→User	"Okay, just wait a moment"
22	SA→AM	`Scene( ... )`
23	AM→Voc	`CheckScene`
24	Voc→User	"Is this correct?"
25	Voc→User	"Yes"
26	Int→Con	`PosFeedback`
27	Con→AM→SA	`ConfirmScene`

Fig. 4. Example dialog between human and robot and the information flow in the system. The second column indicates the currently active system modules. The third column shows the messages being send between the modules. SA = Scene Analysis. AM = Action Manager. Voc = Vocalizer. Rec = Recognizer. Cont=Contextualizer.

a segment and let us define a set of normalized histograms as follows:

1) 1D histogram $P(C)$ with 30 bins over color hue C.
2) 3D histogram $P(N)$ over the number of 3D object points N in each voxel (10mm side length) in a grid over the whole scene.
3) 3D histogram $P_{BB}(N)$ over the number of 3D object points N within the oriented object bounding box BB.
4) One normalized 1D histogram $P_x(N), P_y(N), P_z(N)$ for each axis x, y and z of BB counting the number of 3D points falling into 10mm wide BB slices along the respective axis.

Both 3D histograms, $P(N)$ and $P_{BB}(N)$, will be relatively sparse compared to the 1D histograms. While $P(N)$ reflects on the size of an object segment relative to the whole scene, $P_{BB}(N)$ is normalized to the dimensions of the oriented bounding box and instead measures the complexity of the 3D distribution of points. $P_x(N), P_y(N), P_z(N)$ break down this complexity to the single bounding box axes.

The entropy values of these histograms are normalized with their respective maximum entropy. We use these values to form a feature vector

$$f_i = (H[P(C)], H[P(N)], H[P_{BB}(N)], H[P_x(N)], \ldots \ldots H[P_y(N)], H[P_z(N)]) \quad (2)$$

for each object hypothesis i. From a set of example scenes that are initially segmented, we can extract labeled data to train an SVM with an RBF kernel for classifying an object hypothesis as either being correct or not. We use the probability estimates provided by [33] to rank the object hypotheses according to their uncertainty.

C. Re-Segmenting Based on User Input

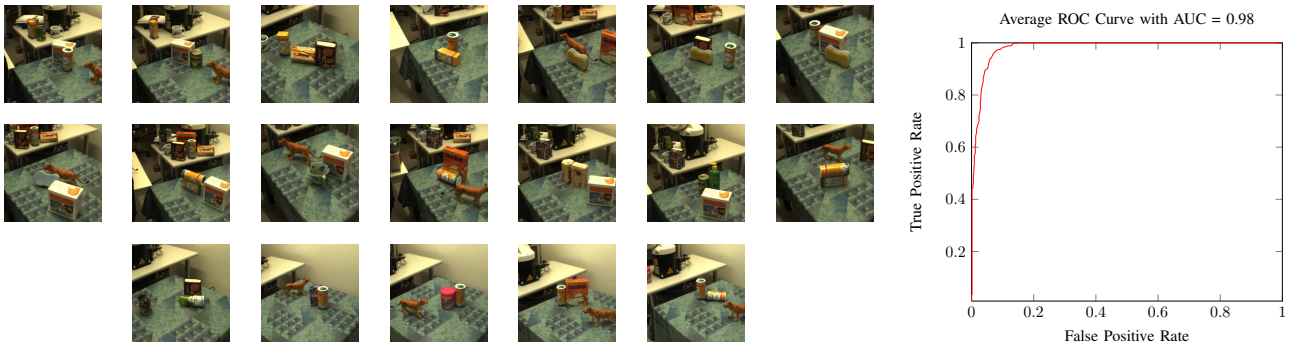
Once a possible candidate for refinement has been identified, the algorithm proceeds to query the user to determine two things: (i) Is the current segmentation correct? And (ii) if not, what is the relative relationships of the objects in the current incorrect segment. In this paper, we limit the user's options for spatial relationships to two objects being either next to, in front of or on top of one another. We propose this limited set of configuration to simplify the interaction. Furthermore, the robustness of the MRF segmentation algorithm eliminates the need for tight boundaries. Despite the simplicity of these three rules, quite challenging scenes can be resolved in practice.

To begin the re-segmentation, the extents of the original segment are calculated to produce an object aligned bounded box. This bounding box is then divided in half along one of the three axes based upon the human operator's input. Once the bounding box is divided the initially segmented points are relabeled and new Gaussian Mixture Models for the region are iteratively calculated as in [14]. A new graph is constructed based upon the probability of membership to these new models. Energy minimization is performed for the new regions and repeated until convergence. This process is effectively attempting to find the most stable set of n mixture models for these points, where n is the number of user specified objects in the region.

In a similar manner, the more rare situations in which two object hypotheses have to be merged can be dealt with. The 3D points initially labeled with two different labels would be re-labeled to carry just one. The implementation of this case is considered as future work.

VI. Experiments

In this section, we will show that through the interaction of a robot with a human operator, the resulting segmentation is improved and that this interaction is made more efficient through the use of an entropy based selection mechanism.



(a) (b)

Fig. 5. (a) Pictures of the 19 scenes in the data set on which the experiments are performed. Ordering from left to right and then top to bottom. (b) The receiver operating characteristic curve which displays true positives vs. false positives for the binary classifier trained to recognize mislabeled object segments. This curve represents 5 random shuffles of the data into 50 training examples and 214 testing examples. The ROC parameters are over all these runs. To summarize the plot the area under the curve (AUC) is also shown.

There exist only a few databases containing RGB-D data, e.g. in [34]. However, they are targeted at object recognition and pose estimation instead of segmentation and therefore usually contain scenes with single objects. Therefore, we recorded 19 scenes containing two to four objects in specifically challenging scenes, see Fig. 5(a). Reconstructing the point cloud and obtaining the seeding for the initial segmentation is done through the active vision system described in Sec. II. Instead of exhaustively reconstructing the whole scene, it uses attention to fixate on salient points in the scene. For these particular examples, we weighted those attention points higher that are closer to the green table. An example point cloud for the first scene is shown in Fig. 1.

In the recorded scenes, objects are hard to separate from each other due to similar appearance or close positioning. There is one incorrectly segmented pair of objects in each of the 19 scenes. Each scene is labeled with human ground truth, thereby separating each object perfectly.

A. Identification of Incorrect Object Hypotheses

We proposed an entropy-based system to select most uncertain object hypotheses to be confirmed by the user. For evaluation, we captured and labeled 264 examples 127 incorrect (positive) and 137 correct (negative) hypotheses i and computed the feature vector f_i as described in Eq. 2.

The data is randomly divided into a training and a test set. The training set contains 25 positive and 25 negative examples, the test set 214 examples. The complete set was randomly divided five times and the aggregated results appears as a receiver operating characteristic (ROC) curve in Fig. 5(b). As can be seen from the plot and the area under the curve of AUC=0.98, the classifier is easily able to distinguish between correctly and incorrectly labeled examples in this dataset.

B. Improvement of Segmentation Quality

To validate the technique, we present experimental results designed to display its performance on the 19 scenes in the data set. We compared the resulting total segmentation accuracy before and after interaction with the human operator on the mislabeled segments.

The graph in Fig. 6(a) displays two bars for each segment. Both display the total classification rate for all object points in the segment as compared to the hand labeled ground truth. The first bar in each set is with the aid of human input in the proposed framework, while the second is without.

The increase in performance of approx. 33.25% on average is attributable to the correct labeling of an object initially missing from the segmentation or incorrectly split. A confusion matrix for a typical under-segmentation can be seen in Fig. 6(b) and the resulting correction achieved through interaction in Fig. 6(c). This exact process is highlighted in the flow of Fig. 1 where the correct segmentation of an initially undetected object is shown.

VII. Conclusions

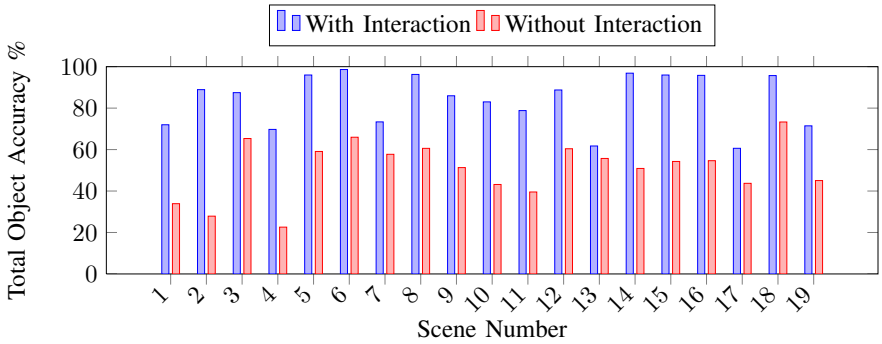
We proposed a novel human-robot-interaction framework for the purpose of robust visual scene understanding. A state-of-the-art computer vision method for performing scene segmentation was combined with a natural dialog system. Experiments showed that by putting a human in the loop, the robot could gain improved models of challenging 3D scenes when compared with pure bottom-up segmentation. Furthermore, through an entropy-based approach the interaction between human and robot could be made more efficient.

As future work, we would like to address the problem of how to represent objects efficiently in the working memory and label them with symbols that are convenient for a human operator. Furthermore, we want to conduct user studies with non-expert user to gain more insight into interaction patterns for this specific task. Additionally, it would be interesting to let the robot iteratively learn a classifier to identify poor object hypothesis during human-robot interaction.

References

[1] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel, "Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, Alaska, USA, May 2010.

[2] A. Saxena, J. Driemeyer, and A. Ng, "Robotic Grasping of Novel Objects using Vision," *International Journal of Robotics Research*, 2008.



(a) Average of approx. 33.25% increase in overall performance when misclassification is attributable to under segmentation. Note that in some cases when there is quite a poor initial segmentation even with the human input the total accuracy remains low. This suggests speech input is most useful when the automated technique has at least some understanding of the scene.

Predicted \ Actual	Bkrd.	Obj 1	Obj 2
Bkrd.	**42939**	456	410
Obj 1	797	**9667**	7054
Obj 2	1367	419	**11750**
Acc:	95.2%	91.7%	61.2%

(b) Interaction Confusion Matrix

Predicted \ Actual	Bkrd.	Obj 1	Obj 2
Bkrd.	**42131**	456	571
Obj 1	2972	**10086**	18643
Obj 2	0	0	**0**
Acc:	93.4%	95.7%	0.0%

(c) Without Interaction Confusion Matrix

Fig. 6. The results of a comparison to hand labeled ground truth across an aggregate set (a) and the confusion matrix for scene 1 with 6(b) and without human input 6(c). Note the small decrease in performance on object one. This effect is created when the re-segmentation attempts to shift some of the points from object one to object two. In this process misalignment of the bounding box and the lack of fine control over the split before optimization can result in a slight decrease in single object performance.

[3] S. Calinon, *Robot Programming by Demonstration: A Probabilistic Approach*. EPFL/CRC Press, 2009.

[4] M. Björkman and J.-O. Eklundh, "Foveated Figure-Ground Segmentation and Its Role in Recognition," *Proc. of British Machine Vision Conference*, 2005.

[5] J. Bohg and D. Kragic, "Learning grasping points with shape context," *Robotics and Autonomous Systems*, vol. 58, no. 4, pp. 362 – 377, 2010.

[6] M. Johnson-Roberson, J. Bohg, M. Björkman, and D. Kragic, "Attention Based Active 3D Point Cloud Segmentation," in *IROS 2010*, Taipeh, Taiwan, October 2010, accepted.

[7] S. Harnad, "The symbol grounding problem," in *PhysicaD: Nonlinear phenomena*, vol. 42, 1990, pp. 335–346.

[8] J. Kenney, T. Buckley, and O. Brock, "Interactive Segmentation for Manipulation in Unstructured Environments," in *IEEE Int. Conf. on Robotics and Automation*, 2009, pp. 1377–1382.

[9] G. Metta and P. Fitzpatrick, "Better Vision through Manipulation," *Adaptive Behavior*, vol. 11, no. 2, pp. 109–128, 2003.

[10] D. Katz and O. Brock, "Manipulating articulated objects with interactive perception," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, May 2008, pp. 272–277.

[11] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, "Gaussian Processes for Object Categorization," *International Journal of Computer Vision*, vol. 88, pp. 169–188, 2010.

[12] A. Vrečko, D. Skočaj, N. Hawes, and A. Leonardis, "A computer vision integration model for a multi-modal cognitive system," in *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, October 2009, pp. 3140–3147.

[13] J. R. Hedvig Kjellström and D. Kragic, "Visual Object-Action Recognition: Inferring Object Affordances from Human Demonstration," *Computer Vision and Image Understanding*, 2010, (to appear).

[14] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, August 2004.

[15] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, "Lazy snapping," in *ACM SIGGRAPH 2004 Papers*, ser. SIGGRAPH '04. New York, NY, USA: ACM, 2004, pp. 303–308.

[16] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz, and M. Goodrich, "Common metrics for human-robot interaction," in *HRI '06: Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*. New York, NY, USA: ACM, 2006, pp. 33–40.

[17] P. Sturgess, K. Alahari, L. Ladicky, and P. H. S. Torr, "Combining appearance and structure from motion features for road scene understanding," in *Proceedings of British Machine Vision Conference*, 2009.

[18] T. Malisiewicz and A. A. Efros, "Beyond categories: The visual memex model for reasoning about object relationships," in *NIPS*, December 2009.

[19] R. S. Li-Jia Li and L. Fei-Fei, "Towards total scene understanding: Classification, annotation and segmentation in an automatic framework." in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2009.

[20] J. Bohg, M. Johnson-Roberson, M. Björkman, and D. Kragic, "Strategies for multi-modal scene exploration." in *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[21] T. Asfour, K. Welke, P. Azad, A. Ude, and R. Dillmann, "The Karlsruhe Humanoid Head," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, Daejeon, Korea, December 2008.

[22] B. Rasolzadeh, M. Björkman, K. Huebner, and D. Kragic, "An Active Vision System for Detecting, Fixating and Manipulating Objects in the Real World," *The International Journal of Robotics Research*, vol. 29, no. 2-3, pp. 133–154, 2010.

[23] Y. Boykov, O. Veksler, and R. Zabih, "Markov random fields with efficient approximations," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1998, pp. 648–655.

[24] ——, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1222–1239, 2001.

[25] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147–159, 2004.

[26] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1124–1137, 2004.

[27] G. Skantze, "Jindigo: A Javabased Framework for Incremental Dialogue Systems," 2010, www.jidingo.net.

[28] D. Schlangen and G. Skantze, "A General, Abstract Model of Incremental Dialogue Processing," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL09)*, Athens, Greece, 2009.

[29] P. Lamere, P. Kwok, E. Gouvea, B. Raj, R. Singh, W. Walker, M. Warmuth, and P. Wolf, "The cmu sphinx4 speech recognition system," in *Proceedings of the IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, HongKong, 2003.

[30] G. Skantze and J. Edlund, "Robust interpretation in the higgins spoken dialogue system," in *Proceedings of ISCA Tutorial and Research Workshop (ITRW) on Robustness Issues in Conversational Interaction*, Norwich, UK, 2004.

[31] G. Skantze, "Galatea: A discourse modeller supporting concept-level error handling in spoken dialogue systems," in *Recent Trends in Discourse and Dialogue*, L. Dybkjær and W. Minker, Eds. Springer, Aug. 2008.

[32] M. Schröder and J. Trouvain, "The german text-to-speech synthesis system mary: A tool for research, development and teaching." *International Journal of Speech Technology*, pp. 365–377, 2003.

[33] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[34] K. Lai, L. Bo, X. Ren, and D. Fox, "A Large-Scale Hierarchical Multi-View RGB-D Object Dataset," in *IEEE International Conference on on Robotics and Automation*, 2011.