Incremental Referent Grounding with NLP-Biased Visual Search

Rehj Cantrell
Indiana University
Bloomington, IN

Evan Krause and **Matthias Scheutz**
Tufts University
Boston, MA

Michael Zillich and **Ekaterina Potapova**
Technische Universität Wien
Vienna, Austria

Abstract

Human-robot interaction poses tight timing requirements on visual as well as natural language processing in order to allow for natural human-robot interaction. In particular, humans expect robots to incrementally resolve spoken references to visually perceivable objects as the referents are verbally described. In this paper, we present an integrated robotic architecture with novel incremental vision and natural language processing and demonstrate that incrementally refining attentional focus using linguistic constraints achieves significantly better performance of the vision system compared to non-incremental visual processing.

Introduction

Spoken natural language understanding (NLU) situated in a human-robot interaction (HRI) context is critically distinguished from other NLU applications by human expectations. In particular, human speakers expect colocated listeners to rapidly and incrementally integrate perceptual context (c.f. (Clark and Marshall 1981)). Such rapid integration can be used to solve difficult natural language processing (NLP) problems, such as resolving references and reducing parse tree ambiguity.

The success (or failure) of this integration in constraining the semantic interpretation of the utterance is communicated to the speaker though backchannel feedback such as gaze, verbal acknowledgments, and head nodding, all produced during the processing of the ongoing utterance (c.f. (Schiffrin 1988)). This feedback loop necessitates full bi-directional integration of incremental vision and NLP systems, each constraining the other. In particular, natural language descriptions can reduce the vision search space for particular objects, thus increasing the speed of visual reference resolution, while visually-acquired sensory information reduce parse-tree ambiguity by resolving attachment problems.

In this paper, we evaluate the effectiveness of an integrated incremental language and vision system by comparing the operation of two vision processing modes: in the first, a complete description of an object is first generated from natural language input, followed by a single visual search through all existing candidates (i.e., every object in the environment) for the referent; in the second, information gleaned incrementally from natural language input is used to constrain vision's search by progressively narrowing the field of possible candidates, in effect focusing the robot's attention on an increasingly restrictive set of criteria. We demonstrate that, by constraining vision in this way, the system is able to resolve references significantly faster.

The structure of the paper is as follows. In Section 2, we describe the problem in detail and review previous work in both NLP and vision processing. Then, in Section 3,we introduce our approach to accomplishing the integration of the two types of processing. In Section 4, we discuss an experiment that serves to evaluate our approach, closing in Section 5, with a summary of our accomplishments and proposals for future work.

Motivation

Imagine a scenario where a human instructs a robot to put objects in their proper places in a living room. The instructions will likely include object descriptions meant to uniquely describe, out of all possible candidate objects, one specific object or set of objects. It will also likely include location descriptions, often constructed from one or more prepositional phrases. The former presents a problem of visual search, while the latter presents a parsing problem.

The robot will typically be faced with several candidate objects for clean-up and various possible places to put them when trying to resolve referential expressions singling out objects such as "the red *book* on the floor", and spatial relations indicating goal locations such as "*on* the shelf *next to* the vase". When instructing a robot, humans will naturally look towards an intended object or point to it, gazing back at the robot to check whether it is attending to the object (Yu, Scheutz, and Schermerhorn 2010). If the robot is able to follow the human eye gaze to the target object, both human and robot will establish joint attention which will allow the human instructor to check quickly (and often subconsciously) that the robot understood the request correctly. In addition to looking at the object, humans will typically also expect a robot to verbally acknowl-

edge understanding by saying "OK" or "got it", or ask for clarification effectively such as "the one by the table?". Feedback is often already required for partial utterances, again through eye gaze, verbal acknowledgments, or through the immediate initiation of an action such as the robot reaching for a book after it heard "put the red book..." while the utterance is still going on.

Note that in such an interactive setting, vision and NLP can mutually and incrementally constrain each other. For example, visually observing a scene that is being talked about can support understanding of ambiguous or underspecified utterances while they are being processed – "the red book on the floor" will most likely refer to a book visible to the instructor, *not* the one behind her back. Similarly, a syntactically ambiguous sentence like "put the book on the table on the shelf" will become clear as soon as the robot detects a book on the table, thus using visually observed spatial relations to constrain parsing and semantic analysis.

Conversely, incremental processing of a verbal description of a scene can direct visual processing to the relevant elements, e.g., "Put the red [[now prioritizing the processing of red image regions]] shoe on [[now prioritizing horizontal supporting surfaces on which an object can be placed]] the box", or "Take the shoe on your left [[now prioritizing the lower left field of view]] ...". In addition, non-linguistic cues such as pointing and gaze direction can be incrementally integrated with partial meanings to steer attention to those elements of the scene relevant to the current discourse situation.

While no current robotic NLU systems yet approach the ability to handle natural unrestricted spoken input, several efforts have advanced the state-of-the-art in natural language interactions with artificial entities by tackling different aspects of these challenges. For example, several robotic systems add genuine NLU components to the robotic architecture (c.f. Michalowski et al.'s robot GRACE combines speech with a touch screen (Michalowski et al. 2007); Müller et al's semi-autonomous wheelchair (Müller et al. 1998) responds to coarse route descriptions; Moratz et al. use goal-based or direction-based spoken commands to guide a robot through an environment (Moratz, Fischer, and Tenbrink 2001); Firby's Reactive Action Packages (RAPs) (Firby 1989) tightly integrate natural language and action execution; and Kruijff et al. (Lison and Kruijff 2009) are pursuing directions in incremental NLU for HRI very similar to ours (Brick and Scheutz 2007)).

However, only a few complete NLU systems operate in real-time. Allen et al. (Allen et al. 2007) use a manually-designed bottom-up chart parser with preferences and manually-defined weights rather than more standard probabilities. Syntactic analysis is complemented by semantic analysis that returns a logical form as a semantic network. One drawback of this architecture in an HRI setting is its standard pipeline architecture (i.e., syntactic analysis is completed before semantic analysis can begin) which prevents an embodied agent from timely backchanneling. Still more integrated is the system by Schuler et al. (Schuler, Wu, and Schwartz 2009) which processes phonological, syntactic, and referential semantic information incrementally; however, the system has not been used on a robot.

Several lines of research have addressed the problem of modulated object search and interactive or incremental visual processing. Unconstrained object segmenation is a notoriously hard and ill-defined problem. Mishra et al. (Mishra and Aloimonos 2009) use a seed point, obtained from user input or attention, together with a log-polar image representation to improve segmentation in 2D and depth images; Johnson-Roberson et al. (Johnson-Roberson et al. 2010) segment point clouds with a similar technique for robot grasping.

While bottom-up attentional processes are well known, more recent work addressed how top-down cues could bias visual search in a task-dependant manner. Choi et al. (Choi et al. 2004) train an adaptive resonance theory (ART) network from human labeling to inhibit bottom up saliency for non-relevant image regions. The VOCUS system by Frintrop et al. (Frintrop, Backer, and Rome 2005) employs bottom-up (scene-dependent) as well as top-down (target-specific) cues, which are learned from training images, leading to increased search performance. Navalpakkam et al. (Navalpakkam and Itti 2006) show how search speed can be maximized by incorporating prior statistical knowledge of target and distractor features to modulate the response gains of neurons encoding features.

The concept of incremental visual processing has not received much attention. Typically the aim is simply to make vision methods "as fast as possible". However often not all results are needed immediately or there is a trade-off between speed and accuracy. In one early attempt, Toyama et al. (Toyama and Hager 1996) layer so-called "selectors" and "trackers" such that selectors at lower (coarser) levels reduce the set of object candidates for higher levels, with trackers at the top generating output sets of size one. Failure at level i lets the system fall back on layer $i-1$, with a broader search space but smaller accuracy. The system can thus robustly maintain track, adjusting search space and accordingly tracking accuracy to changing conditions. Zillich (Zillich 2007) shows how an incremental approach in the perceptual grouping of edge segments removes the necessity of tuning parameters, which are often difficult to select and tend to lead to brittle systems.

Most related to ours is work on interaction between vision and language by Bergström et al. (Bergstrom, Bjorkman, and Kragic 2011) and Johnson-Roberson et al. (Johnson-Roberson et al. 2011) who perform interactive segmentation of 2D images and 3D point clouds based on real-time MRF graph partitioning. Dialogue such as *robot*: "I think there are two objects" *human*: "No there are three objects" or *robot*: "So, should I split the green segment?" *human*: "No, the yellow one!" biases graph partitioning to form the most likely objects. However their work explicitly requires interaction in both ways to refine segmentation, rather than

just collecting attentional cues from the human.

While these and related research efforts tackle various aspects of NLU and vision, no existing framework allows for a deep integration of these different algorithms with a complex vision system into a unified integrated robotic architecture for natural HRI.

Integrating NLP and Vision

The context of situated natural language interactions between humans and robots provides several unique challenges for integrated robotic architectures, in particular, for *visual scene and natural language understanding*. We focus on two: **Challenge 1: Human timing.** All visual, natural language and action processing must be performed and completed within human-acceptable timing, ranging from fractions of a second for eye movements and other motor actions, to at most one second for verbal responses. **Challenge 2: Incremental multi-modal constraint integration.** All processing must be incremental for the robot to be able to determine the meanings of partial instructions, perform any required perception actions including the establishment of joint attention, and either acknowledge understanding or ask for clarification.

We address these challenges through incremental natural language and vision processing. The former gradually builds a hierarchical semantic representation, requesting a new vision search for each new discourse entity. The latter then allows for the continual refinement of the search by the addition of new filters as additional description is given by the speaker.

Incremental NL

Our incremental natural language system uses a shift-reduce dependency parser trained on approximately 2500 sentences comprising the training set (sections 02–21) of the Wall Street Journal (WSJ) corpus. The parser identifies labeled head/argument pairings (e.g., subject/predicate or object/predicate) and identifies, for each word, a manually-created dictionary definition. The argument structure is used to select a compatible definition from several possibilities. In this way, a semantic representation is produced for the utterance.

The semantic representation is produced incrementally: when a token is added or its argument structure augmented by the addition of a new argument, a new semantic definition is selected. Sensory information is requested incrementally as well: each time a new entity is referenced, a new vision search begins, and associated visual constraints (both adjectival and prepositional modifiers) are sent to vision as they are attached to their noun head. Semantics are produced in this way for a variety of types of utterances, including instructions, direct and indirect questions, and statements.

For each entity referenced in one of the above types of utterances, the robot consults its knowledge about the entity and determines what type of sensor should be used to identify and investigate the entity's properties.

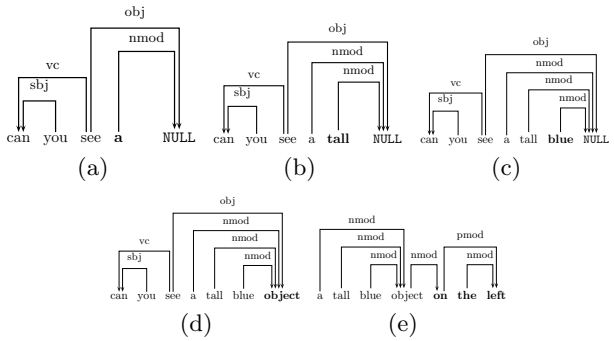


Figure 1: "Do you see the box on the left?"

For the purposes of this paper, only visual entities (i.e., those requiring visual sensors) are discussed. In the case of a visual entity x, the robot calls the vision server in order to verify the existence of x. One of three cases then results: (1) The robot is able to identify one or more objects that meet the description, and it assents, "yes". (2) The robot is not able to identify any object meeting the description, and it announces, "I could not find any". or (3) The robot is expecting to find one and only one such object (e.g., "there is *the* [or *one*] blue object"), but it finds multiple such objects, and announces, "I was not able to identify a single referent. Please use a uniquely-identifying description."

This verification process is used in the case of all types of utterances. Given a situation in which two blue objects are before the robot, if the robot is asked, "Do you see the blue object?" or if it is directed "Pick up the blue object," the robot, being unable to find a single uniquely-identified object meeting the description in either case, will request additional constraints in order to narrow the reference down. If there are instead no blue objects, but the robot is still told "there is a blue object," the robot will respond that it cannot find any blue object. Determiners communicate how many objects that meet the description the robot is to expect. The robot distinguishes between three types of determiners: existentials (a, any, some) which requires at least one such object; referentials (the) which requires exactly one object; and universals (all, every, each) that allow any number of objects.

Figure 1 shows an example of incremental processing, beginning midsentence, just as we receive a determiner (our first sign of a coming noun phrase) in Figure 1(a). A dummy entity is created and a visual search begun. In Figures 1(b) and 1(c), adjectives are attached to the dummy entity; as this occurs, each adjective is interpreted as an additional constraints to the vision search. Figure 1(d) sees the appearance of the real noun, which now replaces the dummy. A last attachment is made in Figure 1(e) and the constraint sent to vision.

Incremental Vision

Given the goal of identifying the objects referred to by language, we tackle the segmentation of these ob-

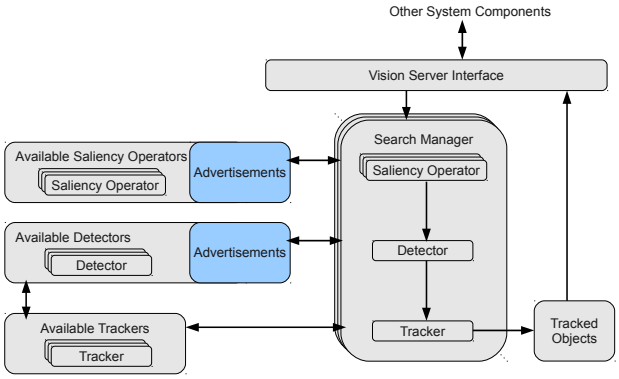


Figure 2: A high-level view of the vision framework.

jects from the overall visual scene by making use of an attention mechanism that relies on cues incrementally obtained from NLP. The input to visual processing are color images overlaid with 3D point clouds obtained with an RGB-D sensor (a Microsoft Kinect) and organised into a rectangular array (depth image).

The relevant types of vision processors are *saliency operators*, *object detectors* and *object trackers* (see Figure 2). In general, saliency operators detect the amount that a modifier such as a color or a location applies to a particular area of space, while object detectors typically search for specific nouns, such as "faces," "persons," "objects," and "clusters," which are then tracked by object trackers. An utterance such as "Do you see the blue object?" starts a visual search process composed of a number of saliency operators and one detector and its associated tracker. Several such visual search processes can run in parallel. Within a search, visual processors are registered to one another so that the completion of a processing iteration in one processor notifies the other processors that are related to the same search.

Saliency operators are computationally cheap bottom-up processes operating independently and in parallel. When created as part of a visual search they are configured using *processing descriptors* (derived from, e.g., the adjectives extracted from the utterance) that specify what quality is being sought in this specific search. Each saliency operator then outputs a 2D saliency map with values between 0 (not salient) and 1 (maximally salient) overlaid on the 3D point cloud. The output of different saliency operators is finally combined by multiplying the saliency maps.

Color is an object property often used to point out a specific object in a scene. The *color saliency* operator maintains a list of commonly used color words ("blue", "red", "black") associated with points in color space. These associations are currently hand-coded but could also be learned as in (Skocaj et al. 2010). Distances of pixel colors to the salient color selected by the processing descriptor are mapped to saliency values in $[0, 1]$. Several colors can be salient at the same time ("the red or blue object"), in which case the minimum distance to a salient color is used.

Another common property when talking about a scene containing several objects is relative location, as in "Pick up the object on the left". The *location saliency* operator maps location in the image to saliency and can be configured for "left", "center", "right", "top", "middle", or "bottom". Saliency decreases linearly from the selected image border to the opposite or in form of a Gaussian located in the image center.

While operating on the raw input data prior to segmentation does not allow specification of object shape properties (as objects have not yet been segmented), *height saliency* (i.e., object height above the supporting surface) is a simple cue to support properties such as "tall" and "short". Height from ground to the highest point above ground is mapped to $[0, 1]$ for "tall" and $[1, 0]$ for "short". Similarly, *surface orientation saliency*, (i.e., the angle between local surface normal and normal of the supporting plane) is mapped to $[0, 1]$ for "horizontal" and $[1, 0]$ for "vertical".

Some of these operators may not be very distinctive or may be ambiguous (e.g., "short" could be the opposite of "tall" or refer to the small length of an elongated object), so we do not expect each of these operators to output very precise information. Rather, these operators need only to prioritize salient image regions (and thus corresponding parts of the point cloud) in order to render the following segmentation step computationally more tractable.

Object detection is performed by segmenting the 3D point cloud. For the experiments presented here we make the simplifying assumption often used in robotics scenarios (Johnson-Roberson et al. 2010; Wohlkinger and Vincze 2011) that objects are located on a dominant supporting plane. Segmentation then amounts to detecting the supporting plane, subtracting it from the point cloud, and clustering the remaining points into object candidates.

Clustering is based on the Euclidian clustering method provided by the Point Cloud Library (PCL) (Rusu and Cousins 2011) and is computationally the most expensive step. Given that the output of saliency operators cannot be considered very precise, we explicitly avoid thresholding based on the combined saliency map to yield distinctive regions of interest. Meaningful thresholds are difficult to define and will change from scene to scene. Our approach is to instead sort 3D points in order of decreasing saliency and use a modification of the PCL Euclidian clustering method to start with the most salient point, greedily collect neighbouring points and output the first most salient cluster, then repeat. So we make sure that the most salient objects pop out first and are immediately available as referents for language, while less salient objects follow later.

In order to bind detected objects as visual referents a final decision has to be made whether an object is, e.g., "blue and tall". This decision is based on a threshold value, performed on segmented objects rather than on saliency maps. Once a detector has successfully seg-

mented objects from a scene, **object tracking** is performed by a tracker tasked with consuming the resulting objects and tracking them from frame to frame. To this end, previously found objects are associated with new ones based on spatial consistency. Two objects are considered equal if they overlap by more than 50%, otherwise a new object is added to the tracker.

These processors work in tandem with each other and share information. A visual search for a "tall red object," for instance, might consist of an "object" detector using the results from a "red" saliency operator and a "tall" saliency operator. These implementation details are transparent to outside components such as natural language. Transparent interaction is provided by the interface described in the next subsection.

The Interface between Vision and NL

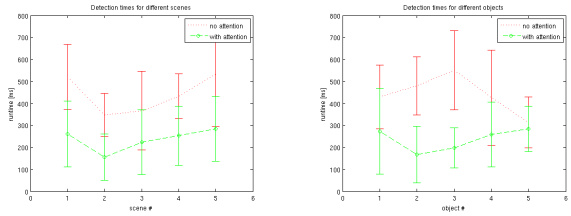
In order for a robotic system to perform naturally in the context of human-robot interactions, a robot vision system must quickly respond to incremental cues from natural language in order to dynamically instantiate and modify visual searches. To accomplish this, a vision system needs to expose an interface capable of naturally handling requests from natural language components, thereby freeing language components from requiring an intimate knowledge of visual components and their capabilities. A common currency must exist between language and vision components to enable this timely and natural interaction. Additionally, the vision framework must be able to rapidly convert requests (possibly incomplete) from natural language into meaningful visual searches in a robust and dynamic way.

The interface between natural language and vision is handled by *search managers*, the highest level mechanism responsible for dynamically building searches from natural language cues, which are used to shield outside components from internal implementation details. When a new visual search is triggered by an outside component via a call to *startNewVisualSearch*, a new search manager is automatically instantiated, and a unique search ID is returned to the caller so that future visual constraint requests can be properly associated with that particular search. *addVisualConstraint* is then used to populate the search with the appropriate combination of saliency detector and tracker without the outside component being required to know any of the details of the different processor types.

Because each processor has a unique capability depending on its underlying implementation, processors that are used external to the vision system are responsible for advertising their capabilities to the search manager in order to allow it to populate the search with the appropriate vision processors. For example, a processor capable of generating saliency maps for various color values might advertise "red," "green," and "blue." These advertisements are specified at runtime via a series of xml configuration files. (In keeping with the responsibilities of different types of processors as described in the previous subsection, detector

(a) "Do you see the red object?" (b) "Do you see a green tall object on the right?"

Figure 3: Note the drastic changes in lighting in these scenes.



(a) Average time to detect target object across scenes (b) Average time to detect target object across objects

advertisements are generally nouns, as opposed to the description-based advertisements of saliency operators.) In this way the distinction between saliency operators and detectors is hidden within the vision framework, and outside callers are not required to have knowledge about their differences. Search managers automatically route incoming predicates to the most appropriate vision component.

Once objects have been detected and reach the tracking stage, outside components (e.g., NL) can query vision to retrieve results about the visual search (e.g., by calling *getTokensByTypeId*). Once a visual search is no longer needed, a request to vision to *endVisualSearch* can be made, which stops all vision components related to that particular search.

To summarize, as a search manager receives incremental constraints, the incoming predicate is mapped to a new instance of the appropriate vision component. An arbitrary number of constraints can be incrementally added to a search, and a fully functional visual search is composed of a detector, tracker, and zero or more saliency operators. Clients are relieved from details of the underlying vision framework, providing only a search ID and predicate constraints to build visual searches and query for results.

Results and Discussion

We evaluated the effectiveness of language-modulated attention by measuring the time needed to identify a specific discourse referent. We constructed five scenes, each composed of five objects, which were subsequently referred to in utterances such as "Do you see the red object?" or "Do you see a tall green object on the right?" (see Figure 3). Note the drastic differences in lighting,

which would make simple thresholding methods based on color very challenging.

Each scene (object configuration) was paired with a set of five utterances, each uniquely identifying a different target object within the configuration. Each scene/utterance run was repeated 10 times as the time to identify the target was measured. Without attention (i.e., performing a single vision search on all objects) the order in which objects were checked for compatibility with the description was random; on average the target object was found after checking half of the objects. When attention was used to incrementally filter the visual scene for saliency (in terms of the descriptive constraints) the target was often the first detected.

The average times and standard deviations are illustrated in Figures 4(a) and 4(b). Figure 4(a) shows the time until detection of the target object for each scene, averaged over all target objects. In most cases, we can see that the average detection time without attention is roughly twice the detection time with attention. This is what we would expect: with attention the target object is typically the first found; without attention the target object is on average the 2.5-th found. Figure 4(b) shows average detection times per object over all scenes. Note that object $O5$ (the tall green Pringles can) shows almost no improvement. In the absence of attention, pixels are clustered row by row beginning at the top. As a result, the tallest object often happens to be the first object detected. For smaller objects the difference is more pronounced.

These results clearly demonstrate that attentional cues obtained from dialogue efficiently steer visual processing to relevant parts of the scene, resulting in significantly reduced runtimes for detecting target objects. However, the system does have limitations.

While the system is quite general due to the training grammar and handling of different types of utterances, it is negatively affected by the lack of explicit directionality inherent in the dependency grammar. The lack of directionality extending into the natural language definitions, the system has some difficulty distinguishing questions from statements. For example, *is* requires a subject and a predicate (e.g., "[there]$_{SBJ}$ is [a blue box]$_{PRD}$, represented by the definition $\lambda x_{SBJ}.\lambda x_{PRD}.exists(y_{PRD})$. In both questions ("is there y") and statements ("there is y"), the same argument structure, and thus, problematically, the same definition is used, and the identical semantics produced (e.g., $exists(y)$). In the case of the question, however, the semantics produced should instead be $report(self, exists(y))$, indicating that the robot should report the truth of the proposition $exists(y)$.

Currently, this is handled by treating statements as questions: the robot must report on the truth of a statement as if it were answering a question. This results in a positive benefit, that the report is constantly verifying everything it is told rather than assuming it to be true; however, it results from a general limitation: in all cases where multiple word senses share the same POS tag/valency pair, but different semantic definitions, it is not currently possible to distinguish between the two senses and use the different definitions.

A further limitation is that, while the syntactic system is trained on a large corpus and is thus generally applicable to a large variety of sentences, currently the only part of the definitions that is learnable from annotated data is the argument structure or valency. The semantic form itself is learnable only from a set of manually-written rules. While the existence of groups of words forming semantic forms in exactly the same way does render the writing of such rules feasible, relying on rules is less than ideal. This limitation is currently being addressed in ongoing research.

In the experiments presented here, the visual processing required was fairly simple; one may be tempted to argue that with some optimisation, possibly including GPU implementation of the 3D point clustering based segmentation step (the computational bottleneck in our case), the system could be made "fast enough" without requiring this integration. But visual processing does not stop here. Once we add object categorization and recognition, and begin to eliminate the initial simplifying assumptions, we are bound to yet again hit performance bottlenecks. Biological vision systems have developed attentional mechanisms to be able to quickly react to the relevant parts of the visual scene. Accordingly the focus in our work lies in developing principled attentional mechanisms for the case of human robot interaction to support visual processing at time frames compatible with human language, rather than in optimising specific vision methods for certain scenarios.

Conclusions and Future Work

In this paper, we argued for integrated incremental versions of vision and NLP in order for robots to meet the requirements posed by natural interactions with humans. We demonstrated experimentally that constraining vision with incrementally-acquired natural language descriptions can significantly speed up vision processing, and thus also reference grounding. The reverse direction, constraining natural language interpretation with visually-acquired information about objects, will be the next problem to tackle.

Another extension will address the fact that the decision whether a detected object is considered to meet a verbal description is based on a threshold. Future work will employ probabilistic models of object properties (such as the incrementally learned KDE based representations of Skocaj et al. (Skocaj et al. 2010)) and on how these probabilities can be dealt with by NLP. This will require a substantive extension of the NLU system as well in order to fuse existing confidence measures (e.g., of the parsers) with those coming from the vision system. Finally, we will also populate the vision framework with more processors (such as object categorisation as in Wohlkinger et al. (Wohlkinger and Vincze 2011)), object recognition algorithms as well as a variety of further saliency operators.

References

Allen, J.; Dzikovska, M.; Manshadi, M.; and Swift, M. 2007. Deep linguistic processing for spoken dialogue systems. In *Proceedings of the ACL Workshop on Deep Linguistic Processing*.

Bergstrom, N.; Bjorkman, M.; and Kragic, D. 2011. Generating Object Hypotheses in Natural Scenes through Human-Robot Interaction. In *IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, 827–833.

Brick, T., and Scheutz, M. 2007. Incremental natural language processing for HRI. In *HRI*, 263–270.

Choi, S.-B.; Ban, S.-W.; Lee, M.; Shin, J.-K.; Seo, D.-W.; and Yang, H.-S. 2004. Biologically motivated trainable selective attention model using adaptive resonance theory network. In Ijspeert, A.; Murata, M.; and Wakamiya, N., eds., *Biologically inspired approaches to advanced information technology*. Springer Berlin / Heidelberg. 456–471.

Clark, H., and Marshall, C. 1981. Definite reference and mutual knowledge. In Joshi, A. K.; Webber, B. L.; and Sag, I. A., eds., *Elements of discourse understanding*. Cambridge: Cambridge University Press. 10–63.

Firby, R. J. 1989. *Adaptive Execution in Complex Dynamic Worlds*. Ph.D. Dissertation, Yale University.

Frintrop, S.; Backer, G.; and Rome, E. 2005. Selecting what is important: Training visual attention. In *Proc. of the 28th Annual German Conference on AI (KI'05)*.

Johnson-Roberson, M.; Bohg, J.; Bjorkman, M.; and Kragic, D. 2010. Attention based active 3D point cloud segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Johnson-Roberson, M.; Bohg, J.; Skantze, G.; Gustavson, J.; Carlsson, R.; and Kragic, D. 2011. Enhanced Visual Scene Understanding through Human-Robot Dialog. In *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3342–3348.

Lison, P., and Kruijff, G.-J. M. 2009. Efficient parsing of spoken inputs for human-robot interaction. In *Proceedings of the 18th IEEE International Symposium on Robot and Human Interactive Communication*.

Michalowski, M. P.; Sabanovic, S.; DiSalvo, C.; Font, D. B.; Hiatt, L.; Melchoir, N.; and Simmons, R. 2007. Socially distributed perception: GRACE plays social tag at AAAI 2005. *Autonomous Robots* 22(4):385–397.

Mishra, A., and Aloimonos, Y. 2009. Active Segmentation for Robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Moratz, R.; Fischer, K.; and Tenbrink, T. 2001. Cognitive modeling of spatial refernce for human-robot interaction. *International Journal on Artificial Intelligence Tools* 10(4):589–611.

Müller, R.; Rofer, T.; Landkenau, A.; Musto, A.; Stein, K.; and Eisenkolb, A. 1998. Coarse qualitative description in robot navigation. In Freksa, C.; Braner, W.; Habel, C.; and Wender, K., eds., *Spatial Cognition II*. Berlin: Spinger-Verlag. 265–276.

Navalpakkam, V., and Itti, L. 2006. A theory of optimal feature selection during visual search. In *Proc. Computational and Systems Neuroscience (COSYNE)*.

Rusu, R. B., and Cousins, S. 2011. 3D is here: Point Cloud Library (PCL). In *International Conference on Robotics and Automation*.

Schiffrin, D. 1988. *Discourse Markers*. Cambridge University Press.

Schuler, W.; Wu, S.; and Schwartz, L. 2009. A framework for fast incremental interpretation during speech decoding. *Computational Linguistics* 35(3).

Skocaj, D.; Janicek, M.; Kristan, M.; Kruijff, G.-J. M.; Leonardis, A.; Lison, P.; Vrecko, A.; and Zillich, M. 2010. A basic cognitive system for interactive continuous learning of visual concepts. In *ICRA 2010 Workshop ICAIR - Interactive Communication for Autonomous Intelligent Robots*, 30–36.

Toyama, K., and Hager, G. 1996. Incremental focus of attention for robust visual tracking. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, 189–195.

Wohlkinger, W., and Vincze, M. 2011. Shape-Based Depth Image to 3D Model Matching and Classification with Inter-View Similarity. In *IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, 4865–4870.

Yu, C.; Scheutz, M.; and Schermerhorn, P. 2010. Investigating multimodal real-time patterns of joint attention in an hri word learning task. In *5th ACM/IEEE International Conference on HRI*, 309–316.

Zillich, M. 2007. Incremental Indexing for Parameter-Free Perceptual Grouping. In *Proc. of the 31st Workshop of the Austrian Association for Pattern Recognition (OAGM/AAPR)*, 25–32.