FP6-004381-MACS

MACS
Multi-sensory Autonomous Cognitive Systems Interacting with Dynamic Environments for Perceiving and Using Affordances

Instrument: Specifically Targeted Research Project (STReP)
Thematic Priority: 2.3.2.4 Cognitive Systems

D3.3.2 Sensorimotor decision making and affordance recognition

Due date of deliverable: November 30, 2007
Actual submission date: January 23, 2007

Start date of project: September 1, 2004               Duration: 39 months

Joanneum Research (JR_DIB)

Revision: Version 2

| Project co-funded by the European Commission within the Sixth Framework Programme (2002–2006) |
|---|---|
| Dissemination Level | |
| PU | Public |
| PP | Restricted to other programme participants (including the Commission Services) |
| RE | Restricted to a group specified by the consortium (including the Commission Services) |
| CO | Confidential, only for members of the consortium (including the Commission Services) |
Sensorimotor decision making and affordance recognition

Lucas Paletta, Gerald Fritz, Stefan May, Florian Kintzler, Jörg Irran, Georg Dorffner, Maria Klodt, Erich Rome, Ralph Breithaupt

Number: MACS/3/3.2
WP: 3.3
Status: version 2
Created at: Jan 17, 2007

FhG/AIS Fraunhofer Institut für Autonome Intelligente Systeme, Sankt Augustin, D
JR_DIB Joanneum Research Graz, A
LiU-IDA Linköpings Universitet, Linköping, S
METU-KOVAN Middle East Technical University, Ankara, T
OFAI Österreichische Studiengesellschaft für Kybernetik, Vienna, A
This research was partly funded by the European Commission’s 6th Framework Programme IST Project MACS under contract/grant number FP6-004381. The Commission’s support is gratefully acknowledged.

© JR/DIB 2007

Corresponding author’s address:
DI Dr. Lucas Paletta
Joanneum Research
Institute of Digital Image Processing
Computational Perception (CAPE)
Wastiangasse 6
A-8010 Graz, Austria

Fraunhofer Institut für Autonome Intelligente Systeme
Schloss Birlinghoven
D-53754 Sankt Augustin
Germany
Tel.: +49 (0) 2241 14-2683
(Co-ordinator)
Contact:
Dr.-Ing. Erich Rome

Joanneum Research
Institute of Digital Image Processing
Computational Perception (CAPE)
Wastiangasse 6
A-8010 Graz
Austria
Tel.: +43 (0) 316 876-1769
Contact:
Dr. Lucas Paletta

Linköpings Universitet
Dept. of Computer and Info. Science
Linköping 581 83
Sweden
Tel.: +46 13 24 26 28
Contact:
Prof. Dr. Patrick Doherty

Middle East Technical University
Dept. of Computer Engineering
Inonu Bulvari
TR-06531 Ankara
Turkey
Tel.: +90 312 210 5539
Contact:
Prof. Dr. Erol Şahin

Österreichische Studiengesellschaft für Kybernetik (ÖSGK)
Freyung 6
A-1010 Vienna
Austria
Tel.: +43 1 5336112 0
Contact:
Prof. Dr. Georg Dorffner
## Contents

1 Introduction  
2 Attention and Affordance Perception  
   2.1 Rapid Estimation of Salient Points  
   2.2 Towards Sequential Attention and Affordance Cueing  
3 Affordance Cueing in a Real-world Scenario  
   3.1 Affordance Cueing from Decision Tree based Learning  
   3.2 Affordance Cueing from Reinforcement Learning  
   3.3 Affordance Cueing with Information Fusion on 2D and 3D data  
4 Summary and Conclusions  
References
1 Introduction

This report is meant to be an addendum to the previously published report on ‘Sensorimotor decision making for affordance recognition’ [1] (D3.3.2, v1) which has already published the general methodology of reinforcement learning for affordance cueing. This report will therefore briefly present and discuss advances made in the third project year of MACS that complement the previous documents in order to prepare for the demonstrations at the final review meeting.

A major progress in the third project year was the full integration of the Perception Module (PM) into the MACS architecture, and the implementation of computer vision methodology for real world experiments and evaluation. Fig. 1 depicts the principal sketch of the architecture of the PM. In general, other modules can request the computation of specific entities the results and actual values of which will then be internally represented in terms of an entity trajectory structure over time. These results are outputs of computational units (CU) that perform information analysis (filtering, feature extraction, classification of perceptual categories) on the input data stream. In the third project year, the relevant computer vision methodology for affordance cue recognition has been implemented in the PM. Learning (decision tree based as well as reinforcement learning based) for affordance cueing in terms of feature extraction in time series of feature vectors is so far applied in an off-line, post-processing manner.

Sec. 2 describes the role of attention in affordance cueing and describes recent progress on rapid estimation of salient points. Sec. 3 outlines advances in affordance cueing in real world scenarios, the embedding in a developmental framework of affordance recognition, and recent results on reinforcement learning of affordance features in the MACS scenario. Finally, Sec. 4 presents a summary on the overall advances gained in Task 3.3 on Sensorimotor Decision Making in Affordance Recognition and briefly sketches promising directions of future research.

2 Attention and Affordance Perception

There are several aspects on attention in affordance cueing that require consideration. Firstly, attention is mandatory to focus processing on limited sources of information in order to render the interpretation in terms of affordance cueing feasible, in particular, when operating within a real-world scenario. As a consequence, in the MACS scenario we apply a curiosity drive in terms of the visual attention system VOCUS as a focus of attention on the input imagery (Fig. 2). Sec. 2.1 outlines details about a rapid estimation of salient points that determines regions of interest in the image that will then be processed by affordance cueing methodology thereafter. Secondly, we identify the potential to determine affordance cues from an estimation of sequential attention in the input images and over time when tracking various identities of focused regions of interest. Sec. 2.2 briefly sketches this idea and points to promising future work in this direction. As mentioned in previous work of the authors, there might be a close relation between attention and affordance cueing since cueing for opportunities for interaction is focussing on features in the input data that were extracted for the specific purpose to predict events in the future. However, attention on specific features and objects in the scene might be triggered from various sources of information. For example, the extraction of a particular context (task, location,
interaction, etc.) might require to focus on specific features of interest. As a conclusion, attention might be viewed as covering a much wider scope than affordance cueing, but a discussion in depth must be postponed to future work on the issue of affordance cueing, recognition on various levels of abstraction, and attention on salient features.

2.1 Rapid Estimation of Salient Points

The visual attention system VOCUS is used for detecting salient cues in the image stream of both cameras. This system is primarily employed in the exploration phase, when no previous knowledge about the environment is available. In later phases it economises the computational effort for subsequent computational units by reducing the search space, i.e. by focusing on a region of interest in the optical array. Furthermore, we implemented a triangulation method to determine roughly the distance and direction of an attracting region. Only by the use of this information, the robot has to perform the approach towards the region until it is in the range of the crane arm. The precise localisation of related objects is done using data from the laser scanner. The employment of VOCUS as initial processor made it necessary to enhance the processing speed. The initial version performed one image in 130 ms, which is too slow to process every image captured by the cameras. The desired processing speed is below 30 ms since both cameras provide a framerate of 15 Hz. Therefore, we re-implemented VOCUS utilizing the parallel design of the graphics processing unit (GPU). The GPU-based implementation processes one frame in approximately 22 ms on a NVIDIA GeForce 8800 GTX device and enables the closed-loop approach and/or the monitoring of physical processes during interaction. The entire approach was presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’07) in San Diego [2] (paper is attached to this report).
2.2 Towards Sequential Attention and Affordance Cueing

Paletta and Fritz have recently published the work on a computational model on reinforcement learning for sequential attention estimation \[3\]. There is the potential to relate the extraction of relevant sequences of perceptions (focus on informative features) and actions (translations of foci of attention on the input image) to the framework of affordance recognition, i.e., recognising opportunities for interaction. However, due to the limitation of personnel resources in the third project year, there was not sufficient opportunity to develop a methodology and test it on the data in the real world scenario. However, the ideas about a concept to develop this methodology will be presented at the final review meeting of MACS.

3 Affordance Cueing in a Real-world Scenario

A key requirement in MACS is the application of affordance cueing to the MACS real world scenario. Hence in the third project year the focus of work was on the implementation of all required computer vision methodology within the Perception Module so as to make the associated functionalities accessible by the MACS system architecture, i.e., by other modules. Sec. 3.1 describes the details of the implementation of affordance cueing that was learned from the tracking of feature identities, visual feature extraction, and decision tree based classification of affordance cues. Sec. 3.2 briefly outlines the progress for the application of reinforcement learning to affordance cueing in the real world environment. Sec 3.3 points to the possibility of information fusion between affordance cueing both on 2D and 3D data, results will be published at the final review meeting.

3.1 Affordance Cueing from Decision Tree based Learning

Within this section we describe briefly the details of the implementation and integration into the overall MACS architecture. The Perception Module 1 incorporates its functionality into Computational Units (CU’s) (see Deliverable 4.2.1 \[4\] for an explanation regarding CU’s). These CU’s are managed by the Execution Control depending on the status of the
robot and the actual situation in the scenario. E.g., after approaching to an object the Execution Control will start the affordance cueing in order to determine whether this object is liftable or not. During the approach the module performing the curiosity drive (see 2.1) is activated from the Execution Control.

As depicted in Figure 3 the tracking of regions is based on color regions in combination with KLT features. The SIFT descriptors extracted within this regions are classified into two distinct classes (red denotes ‘circular’, blue denotes ‘rectangular’). The classification result is integrated into a histogram per region of interest and a majority voting generates one discrete label (i.e. ‘circular’ or ‘rectangular’) for each region. Together with features like color, size, top/bottom these attributes build up the query into the final decision tree.

The performance of the Perception Module heavily depends on the modules which are activated. The tracking of regions from one frame to the other takes on average 16ms, i.e. every second frame can be evaluated since both cameras provide a framerate of 15 Hz. The affordance cuing takes on average 2 seconds per frame because here the feature detection algorithm is costly and takes 90 percent of time. Sinha et.al. [5] has already shown that for both algorithms used here i) KLT features used in tracking and ii) SIFT feature extraction used for affordance cuing their counterpart GPU-based implementation perform 10 to 15 times faster compared to the CPU version. The timing experiments were performed on a laptop with 1.6 GHz Centrino processor and 500 MB RAM.

3.2 Affordance Cueing from Reinforcement Learning

Reinforcement learning is able to account learning from perception and action and therefore provides a natural framework for the learning of affordance cueing from sensorimotor input. In Deliverable D3.3.2 (version 1 [1]) we provided a detailed description about the methodology and the experiments in the MACS simulated robot environment that proved that reinforcement learning is capable to extract relevant cues to predict opportunities for interaction, and how to generalise in feature space about visual features. The complete approach was presented at the IEEE International Conference on Development and learning (ICDL’07) in London [6] (paper is attached to this report) in a biologically motivated framework of developmental learning of affordances.

For the application of the reinforcement learning methodology in the MACS real-world scenario, data about several test runs were captured in the robot playground at Sankt Augustin. In general, we applied the identical methodology as presented in [1; 6] but referred to visual features and meta-information associated with real world imagery described in Sec. 3.1). The classifier gained from the generalisation of features ([1]) gained results similar to the one received from decision tree based learning (Sec. 3.1)) and will be demonstrated in the final review meeting of MACS.

In the selected scenario of learning affordance cueing for the opportunity of lifting an object, actions are defined by discrete steps of gripper motion (up, down) and magnetization (on, off). Each perceptual state is defined by a discrete feature configuration $S = (c \in \{C\}, d \in \{D_{circ}, D_{rect}\}, e \in \{0, 1\}, h_r \in \{r_j\}, h_g \in \{r_j\}, m \in \{0, 1\})$, with region color class $c$, region shape $d$, configuration type $e$, region elevation level $h_r$, gripper elevation level $h_g$, and magnet state $m$. As outlined in [6], there is a reward of $1$ if the system is in a goal state and $0$ otherwise (with some negative reward of $-1$ issued for unrealistic action selections). The affordance based, purposive selection of features is here represented by relevance weighting of perceptual states in terms of associated expected rewards. Each
extracted image region is attributed by a perceptual state. Perceptual states that anticipate a complete trajectory of perception-action transition leading to the affordance outcome event with high certainty will be associated to high rewards and consequently constitute cue like states. Affordance based perceptual states are implicitly related to the reward function, an associated estimator on affordance hypotheses $P_{A_i}$, and an affordance classifier discriminating corresponding Q-values into affordance cues and irrelevant states.

Fig. 4 depicts the learning curve of the affordance cue classifier over multiple repetitive video test runs. The system was verified using video input and a simulation of actions (magnet on/off switching, actions to lift the crane, etc.) instead of applying it within the robot real world scenario. The mean of the expected accumulative reward in each perceptual state is continuously increasing with a more and more efficient greedy strategy on action selection. The most important fact here is that the positive accumulative reward (with higher rewards in a neighborhood of the ultimate goal, i.e., the lifting of the object above a certain threshold) proves that a meaningful sensorimotor behavior has been devel-
4 SUMMARY AND CONCLUSIONS

Figure 4: Learning curve for the application of reinforcement learning according to the methodology presented in D3.3.2v1 on real world images (Sec. 3.1). The results are well comparable to the results gained within the simulated robotic environment and demonstrate that reinforcement learning for affordance cueing is in general feasible in real world environments.

3.3 Affordance Cueing with Information Fusion on 2D and 3D data

Affordance cueing can be performed on either 2D and 3D data as presented in experiments on laser based scanning for affordance extraction [7] as well as using visual features for cueing [6]. However, there is the question about the specific contribution of each of these modalities for the benefit of affordance cueing in general. In order to perform a quantitative evaluation we performed a calibrated mapping from laser based scanning to the stereo cameras and received from this a correspondence between pixels in the stereo images and pixels in the depth image gained from the laser based scan.

The experiments are still on-going and will be presented at the final review meeting of MACS.

4 Summary and Conclusions

The work in Task 3.3 on 'Sensory-Motor Context and Attention' has in general focused on the aspects of visual attention (Sec. 2.1, reinforcement learning ([8; 1; 6], Sec. 3.2), and the implementation of the computer vision methodology into the MACS system architecture for real world experiments (Sec. 3.1). In addition, there is on-going work about affordance cueing from 2D and 3D information (Sec. 3.3).
The particular contributions in the frame of reinforcement learning for affordance cueing are as follows,

**Methodology for reinforcement learning of affordance cues:** Affordance cueing has been laid out in a framework of perception-action based recognition [8], using reinforcement learning to determine cues in terms of early perceptual states that would predict high rewards towards opportunities for interaction [1; 6]. In the experiments we were able to prove that the system actually selected affordance relevant visual features being representative for early perceptual states.

**Generalisation in affordance cueing:** The learning of the generalisation of affordance features was documented with respect to both decision tree based learning [9] as well as in the frame of reinforcement learning [6]. This framework of learning affordance features in a closed-loop processing seems to be highly promising and will nurture future research towards the recognition of more complex features - even to object recognition - in the future.

**Implementation into MACS system architecture:** all functional components that are necessary for the on-line extraction of affordance cueing were implemented in the Perception Module and are accessible by other modules of the MACS system architecture. In particular, this concerns tracking of regions of interest, extraction of visual features of low and intermediate level of abstraction, and the classification into affordance cues and irrelevant visual features, respectively.

**Experiments in real world environment:** The Perception Module has been implemented to perform real world experiments in the MACS scenario. The performance in the affordance based interpretation of the video frames is highly satisfying and enables an on-line extraction of affordance cues.

The work resulting from the engagement in MACS Task 3.3 is setting up a complete new framework for the extraction of visual features, relating perception to an action based interpretation of the dynamics in the environment. Learning of relevant features turned out to be crucial in order to determine the features that are particularly cueing to affordances for future interaction. This framework poses new questions not only in terms of providing viewpoints on human perception but merely about how to develop machine perception in a way to adjust the interpretation of the environment for the purpose of efficient control. Finally, these kind of affordance based categories - in the framework of reinforcement learning - are associated with a purpose of the acting system and therefore are naturally determined to be useful for the performance of the artificial system.

Promising directions for future research are most obviously towards the work on finding even more abstract feature representations, i.e., object parts and objects as perceived by human agents. There is also the requirement to find abstractions in the action representation by segmenting the sensorimotor input stream into reasonable chunks of information towards the autonomous definition of frequently occurring patterns of tasks and task goals. Finally, as outlined in the concept given in [6], one can research towards integrating increasingly complex abstractions of affordance triplets of (cue, behavior, outcome) in a developmental approach.
References


Tentative proposal for an affordance support architecture


GPU-accelerated Affordance Cueing based on Visual Attention

Stefan May, Maria Klodt, Erich Rome and Ralph Breithaupt
Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS)
Schloss Birlinghoven
D-53754 Sankt Augustin, Germany
{stefan.may, maria.klodt, erich.rome, ralph.breithaupt}@iais.fraunhofer.de

Abstract—This work focuses on the relevance of visual attention in affordance-inspired robotics. Among all approaches in robotics related to Gibson’s concept of affordances [1] the dealing with attention cues is only rudimentary. We are introducing this concept within the perception layer of our affordance-inspired robotic framework. In this context we present a high-performance visual attention system handling invariants in the optical array. This layer builds the base of higher-sophisticated tasks, like a “curiosity drive” that helps a robotic agent to explore its environment. Our attention system derived from VOCUS [2] utilizes the parallel design of the graphics processing unit (GPU) and reaches real-time performance for the processing of online video streams in VGA resolution on a single computer platform. GPU-VOCUS is currently the fastest known visual attention system running on standard personal computers.

I. INTRODUCTION

In the design of robotic agents coping with our real environment, as attempted in the domain of artificial intelligence, vision is a common approach to robotic perception. Typically, an appearance-based recognition stage is implemented that utilizes a model database. An alternative approach, which has its seeds in psychology, attempts to perceive the scene on a functional basis, namely by using so-called affordance cues.

The concept of affordances has been established by J.J. Gibson in 1979 [1]. It defines the set of possible actions accomplishable by an animal in the environment. The central idea of the affordance theory is that an animal is in a bidirectional relation to its environment. In analogy of Gibson’s original concept of affordances, an agent must be able to perceive what the environment affords and must have the capability to act upon these (agent) affordances. Gibson stated that affordances are perceived directly:

“An affordance is an invariant combination of variables, and one might guess that it is easier to perceive such an invariant unit than it is to perceive all the variables separately.”[1, p. 139]

Thus, perception of affordances is not a sequence of perceiving all the properties of an object, classifying these properties into abstract objects, and inferring how these objects could be employed in certain circumstances. Instead, the invariant combination of variables are perceived and utilized without use of any object recognition or labelling stage. In her book An ecological Approach to Perceptual Learning and Development [3] E.J. Gibson gives examples for invariants that are learned by infants, which range from perception of unity through motion to invariants for locomotion. She shows that the perception of space is directly coupled to the development of locomotion. This dependency indicates that an agent can only perceive affordances that are related to any of its possible actions. Another example is that an agent can only perceive whether an object affords lifting if it is capable to attach to the object and to lift it. This affordance inspiration is one of the fundamentals in our EU project MACS [4]. Within this context Paletta et al. presented a novel framework for cueing and hypothesis verification of affordances that could play an important role in future robot control architectures [5]. They also emphasized that it becomes important to consider visual attention mechanisms. The relevance of attention in affordance-inspired perception has first been mentioned by E.J. Gibson who recognized that attention strategies are learned by the early infant to purposively select relevant stimuli and processes in interaction with the environment [3]. In another work from psychology about wayfinding on foot in cluttered environments Cutting et al. described also the importance of fixating salient points [6]. Nonetheless, among all works in affordance-inspired robotics the dealing with attention cues is only rudimentary. The reason is likely to be the computational effort of calculating salient cues permanently. As for example, E.J. Gibson shows in [3] that for biological creatures, it is not enough to work on a snapshot of the environment. An approach in the domain of autonomous robotics that explicitly incorporates the temporal dimension of salient cues attracting attention is still a challenge. One development in computer graphics opens up new vistas for this problem: The programmability and performance increase through parallelism of graphics rendering devices has reached a high level. CPUs are designed for general purpose, whereas GPUs are designed for processing as much data as possible per instruction (SIMD architecture – single instruction multiple data). Especially the fact that typical models of visual attention are massively parallelizeable supports our effort. The focus in this paper lies on the real-time evaluation of attention for affordance-inspired robotics by the use of graphics rendering devices.

The outline of this paper is as follows: Section II elaborates the role of visual attention for perceiving the environment. Section III describes the current relevance of affordances in robot control architectures just as of visual attention. In section IV we put both ideas together and
focus on the role of attention for affordance perception and learning. Section V illustrates experimental results that support our accentuation of attention in affordance-inspired perception and learning tasks performed by GPU-VOCUS. Finally, section VI concludes with an outlook on future work.

II. THE ROLE OF VISUAL ATTENTION

Among all human senses the visual sense provides the most environmental information. Evolution has developed mechanisms to handle the huge amount of information gathered by the visual sense, e.g. visual attention. Mostly we take no notice of the saccadic movement of our eye although we are using it permanently when we are not asleep. The intended purpose of visual attention is focusing on a region of interest for closer investigation. An analysis of the entire scene would be too time-consuming. This means that an efficient utilization of visual attention has been turned out to be advantageous in evolution. The biological inspiration of visual attention systems has a decisive advantage which is considered in the following architecture specification. Visual attention systems are based on many simple features that can be processed in parallel. The weighting of those features provides a highlighting mechanism to emphasize features which are more discriminative to the surrounding [7]. The high computational effort requires either high speed sequential or fast massively parallel computation. The latter can be well utilized on the parallel basis of the biological fundamentals, e.g. in specialized chip implementations [8]. Itti and Koch divided visual attention into two different categories [9]: bottom-up and top-down attention. The first one describes the aspect of salient regions attracting our attention automatically. This happens when an object is highlighted from the remaining scene through its conspicuity in color, intensity or orientation. The processing speed of bottom-up mechanisms for human beings is according to Itti and Koch in the order of 25 to 50 ms for each salient item. The second form of attention, top-down attention, includes selection criteria in the manner of searching for a specific cue. The processing speed of top-down attention is reported to be in the order of 200 ms [9].

Fig. 1. Demonstrator scenario of the EU project MACS [4]. A goal of MACS is to explore affordance-inspired perception and learning for mobile robots related to J.J. Gibson’s theory. The shown robotic agent KURT3D should perceive, learn and utilize its environment in a functional way.

Fig. 2. Visual attention: The most salient region is selected by bottom-up attention. Regions are determined, which are highlighted from the remaining scene through their color, brightness or orientation, here the blue can.

(a) Intensity  (b) Color  

(c) Orientation

Fig. 3. Computed conspicuity maps of Fig. 1. The blue can pops out dominantly from the color map.

To incorporate the temporal dimension of visual attention in realistic situations, it is necessary to fulfill these runtime constraints. Especially for small robotic platforms, it can be difficult to provide the needed computing power onboard. Networking resources are often used to this end, which is not a suitable solution for autonomous robots with the risk of a broken radio contact. So the difficult task of utilizing the onboard computing power as efficiently as possible remains. Up to now there was no visual attention system available, which could process video streams at VGA resolution on a standard single computer platform, while leaving enough computing power for the remaining control programs. In this context, we present an attention system, that runs completely on graphics rendering devices for personal computers. This system is able to process video streams online while keeping the computing power of the central processing unit nearly untouched. Graphics rendering devices are predestined for the computation of many simple features as typically occurring in the computation of the feature maps in visual attention systems. We will further show how visual attention can be used for affordance cueing of time-series in the manner of an action-perception cycle with our GPU (Graphics Processing Unit) version of VOCUS. GPU-VOCUS is currently the fastest known visual attention system running on a standard personal computer. It performs more than 30 Hz with a 32
of above-mentioned contexts implies that the system is fast enough to satisfy a certain performance constraint. These systems used either input images with less resolution than VGA or performed its computation with frame rates lower than 15 Hz. This performance requirement was the reason for the decision to redesign VOCUS in terms of utilizing the parallel computing capabilities of the graphics hardware to full capacity.

IV. ATTENDING AFFORDANCE CUES IN REAL-TIME

Since physical laws will sometimes limit the fundamentals for further performance boosts, e.g. processors cannot keep going up in clock speed forever, parallelism will gain more importance in future. This trend can already be observed for the newest dual-core or quad-core processor generations. Actually massive parallelism has been available in standard computers already for quite a while, namely on the GPU. Using the GPU for speeding up certain algorithms has recently gained more attention.

A. Potential of processing on GPU

Recent graphics hardware either for personal computers as well as for notebooks have been enormously enhanced in their parallel processing capability. The theoretical ratio of computing power between CPU and GPU for available PCs is in the order of some tens up to one hundred (e.g. Intel Pentium 4, 3 GHz: ≈ 3.6 GFlops vs. Pixelshader of NVidia GeForce 7800 GTX 256MB: ≈ 278.6 GFlops). This development has been pushed by the game industry for years and has already attracted attention by the computer vision community, e.g. [16] or [17]. The performance of graphics devices is rapidly increasing. During our evaluation of the capabilities of the GeForce 7 series, the next generation (GeForce 8 series) appeared with even twice as much transistors (278 bn vs. 681 bn).

B. GPU-VOCUS

GPU-VOCUS is a biologically inspired visual attention system based on the “Feature-Integration Theory of Attention” by Treisman [18]. It is derived from VOCUS [2] which was originally designed for computation on the CPU of a single computer platform. VOCUS detects regions of interest (ROI) that “pop up” from their surrounding, named salient regions. For comprehensibility reasons we summarize the computing cascade here (cf. Fig. 4; more details can be found in [2]). It can be divided into five steps:

1) From the input image six image pyramids are derived in order to provide scale-invariance: an intensity pyramid convolved by a gaussian blurring, an orientation pyramid convolved by a Laplacian filter and four color maps, one for each of the colors red, green, blue and yellow (LAB color space).

2) The image and the color pyramid then result in scale maps or scale pyramids, respectively, applying center-surround filters. For the orientation pyramid a Gabor filter with four different orientations (0°, 45°, 90° and
135°) is used. Summed up, there are now 48 generated scale maps as input for the next computation stage.

3) The different scale maps are then rescaled using a bilinear interpolation and summed up into feature maps, 10 in total.

4) Next, a weighted sum of the feature maps results in conspicuity maps.

5) The 3 remaining conspicuity maps are fused into 1 saliency map. The maximum value in this saliency map refers to the most salient region (MSR). The ROI is computed by a region growing algorithm determining the region around the MSR. In order to move the focus to the next salient region in the image, inhibition of return is used that inhibits previously attended salient regions.

Each stage in this sequence cascade has a data flow dependency to its previous stage which makes it necessary to process the cascade sequentially. Thus, multiple rendering passes are needed to produce the desired saliency map on a GPU. The whole “conversion” from colored input images to saliency maps is kept in charge of the graphics pipeline while using texture buffers as rendering targets. We have chosen to use the language GLSL (OpenGL Shading Language) to implement all necessary filters (shader programs). The execution model of such shader programs is fundamentally different from those on a CPU. Each shader program is executed on each pixel that passes a rendering pipeline. So, there is no need to use loops for the processing of each pixel, but there is also less flexibility (an example is given below). Unfortunately, the transmission of data from the host memory to the video memory and back as well as related format conversions constitute overhead. Thus, a speedup can only be achieved, if the runtime of a CPU program is longer than the transmission to and from video memory would take. Hence, there is a data flow dependency between the maps, but not for the operators themselves. The speedup results exclusively from this parallelization. It has also been turned out that the center-surround filters cause a lower computational effort on the GPU than the orientation filters. The potential in the use of integral images on the GPU is not high, wherefore we make no use of them in the first GPU implementation. Further speedup can also be achieved using multiple rendering devices, one for each map or even scale, but we leave that for future work. Even on graphics hardware the precision of data has to be balanced with performance. The first reason for that is the amount of data which has to be transmitted via the PCIe bus to the video memory. A doubling in precision also doubles the data volume. In computer graphics mostly the transmission can be reduced by reusing previously stored textures in the video memory. In computer vision this is different. Each image, captured by a camera or a different sensor, has to be transmitted. Mostly the result has also to be read back into host memory. The read back was a time consuming task on older graphics devices since they were primarily designed to communicate in one direction, i.e. from host to video memory. A test with an AGP version of an ATI Radeon 9800 XT device yielded no performance improvement due to the read back overhead. Second, the processing speed also depends on the data format. Current graphic cards already provide 16 bit and 32 bit floating point precision but nonetheless with the model we used for our experiments we noticed an influence on the transmission as well as on the processing time (cf. table II). All above-mentioned filters could be ported from the CPU to the GPU. The one and only difficulty was constituted by the region growing algorithm. Region growing is typically a serial process that produces irregularly shaped regions. This type of process is difficult to compute on a GPU due to its per-pixel execution model. Since the processing only consumes 1-2 ms on a CPU, we leave it there as post-processing module.

V. EXPERIMENTS AND RESULTS

Our affordance-inspired framework couples the perception and learning with a behavior system. The used robot platform KURT3D is equipped with two cameras, a 3D laser scanner and a crane arm as manipulator (see fig. 1). The specification of the demonstrator scenario defines objects of different morphology, i.e. color, size and shape. Paletta et al. described how these morphologies can be used to find a proper handling categorization (liftable/non-liftable, stackable/non-
stackable, ...)[5]. Laser scanner and cameras are utilized to explore the environment and to detect conspicuous cues. One use case defined in our project starts with the activation of certain behaviors to accomplish the approach of the robot to the determined position of a salient cue. With trials of lifting the associated object, the robot should learn the trilateral relation between affordance cues, actions and outcomes, in this example for the affordance “liftable”. The crane of the robot enables only a few manipulation tasks, but also tasks can be combined to investigate affordance cues on a higher level, e.g. the “stackability” of cans. Paletta et al. also emphasized the importance of timeline series monitoring while applying an action, thus the attention system utilized by the robot has to be real-time capable. The attention system is integrated in the framework as sensor channel and provides important informations about the trigger of an action and the changes of saliency cues during the execution of an action. The first test showed that all objects specified in the demonstrator scenario were detected as salient without any model information.

A continuative experiment discloses the achieved speedups. All given time measurements are comprising needed computations and transmissions, starting at the time when an image is available in the main memory and ending at the time when the final result is located there. That additionally includes for the GPU implementation the time to transmit data from main memory to video memory and back. All measurements have been done on the same machine composed of the components specified in table I.

A. Monitoring feature time dimension

The comparison of the runtime of both CPU versions of VOCUS shows an inherent speedup achieved by the use of integral images [7] (cf. table II). Hence the additional speedup of the GPU version is even more impressive taking the transmission penalty into account. By the way, the GPU implementation the time to transmit data from main memory to video memory and back. All measurements have been done on the same machine composed of the components specified in table I.

<table>
<thead>
<tr>
<th>CPU</th>
<th>Pentium D 3.0 GHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphics device</td>
<td>NVIDIA GeForce 7800 GT / 8800 GTX</td>
</tr>
<tr>
<td>OS</td>
<td>SuSE Linux 9.3, Kernel 2.6</td>
</tr>
<tr>
<td>OpenGL version</td>
<td>2.0</td>
</tr>
<tr>
<td>Cameras</td>
<td>Logitech Quickcam Pro 4000 (15 Hz at VGA)</td>
</tr>
</tbody>
</table>

TABLE I
HARDWARE/SOFTWARE SPECIFICATION FOR THE EXPERIMENTAL SETUP

B. Monitoring the time dimension of a feature’s distance

Since the robot for our demonstrator scenario is equipped with two cameras, we aimed to determine the distance to each feature with a triangulation method. We have taken the bottom-up attention cues of both images and matched them according to their feature vector. Considering the distance of features is advantageous especially for the learning task where an action is involved that entails a chain of outcomes over the time in the direction of the robot, for instance when a can is pushed which then rolls away. The used camera system, which is simply build up of two webcams on a servo device (see fig. 1) does not allow very precise measurements. The absolute distance error to attention cues measured in our demonstrator scenario (4 m in length and 4 m in width) was smaller than 10 cm at any time (100 measurements varying 10 different objects in different distances). For the use case of approaching the affordance cue, this accuracy is adequate when used as estimation for a subsequent localization in a laser scan. The variation of the measured distance towards a “non-moving” cue was below a centimeter resolution which confirms that the desired monitoring of moving attention cues will work in principle. At the moment we can only give this qualitative statement on the variation and leave the precise analysis to future work.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have presented an application of visual attention in affordance-inspired robotics. The fundamentals for the incorporation of attention cues and their temporal dimension have been accomplished by the implementation of a real-time capable attention system. This system has been integrated in our affordance-inspired robotic framework, which couples the perception and learning with a behavior system. On the images provided by two onboard cameras real-time attention is used as “curiosity drive” with the ambition to explore visual cues in the environment. The distances to these cues are calculated through triangulation. The activation of certain behaviors should then accomplish the approach of the robot to the determined position. A trial of lifting the related object will then complete the task. The explore behavior can then be activated again. The implementation of a GPU version of the attention system VOCUS disclosed two important facts:

TABLE II
MEAN RUNTIME OF (GPU-_)VOCUS (20 RUNS/VGA RES.)
Fig. 5. Feature distance determination: In both images two cans are shown that pop out from the scene. Regions from the left and the right camera are matched according to their attention feature vector and triangulation is used to determine their distances to the robot.

1) The first porting of the non-optimized version of the attention system VOCUS results in the best case in a speedup of approximately 101 (65 for 32 bit precision). Compared to the VOCUS version using integral images a speedup of approximately 9 without orientation maps and 6 with orientation maps could be achieved. The GPU version is now able to calculate saliency cues from VGA video streams online even on a single computer platform. The important fact is that CPU resources are freed and can now be used for other tasks.

2) Second, the incorporation of the temporal dimension of salient cues attracting attention is accomplishable. The important fact is that a visual attention system is used in this context to tackle “interesting” objects that have not been stored in a model database. We showed that even the feature’s distance can be monitored over time. A triangulation method applied to salient features of both onboard cameras provided a sufficient accuracy and stability.

A. Future Work

Saliency cues from both top-down attention and bottom-up attention will further be dispatched to the learning module of our framework. Invariants in the saliency cue stream provide the information about the trigger of an action and changes of saliency cues during the execution of an action. We also aim to further decrease the runtime of GPU-VOCUS freeing more and more resources for robot control tasks. There is still much optimization potential for the first implementation presented in this paper, e.g. with the use of multiple graphic cards.

VII. ACKNOWLEDGMENTS

This research was founded by the European Commission’s 6th Framework Programme IST Project MACS under contract/grant number FP6-004381. The Commission’s support is gratefully acknowledged. Furthermore, we thank Simone Frintrop for the initial version of VOCUS and the ongoing collaboration with our institute.

REFERENCES


Abstract - Recently, the aspect of visual perception has been explored in the context of Gibson's concept of affordances [1] in various ways. We focus in this work on the importance of developmental learning and the perceptual cueing for an agent's anticipation of opportunities for interaction, in extension to functional views on visual feature representations. The concept for the incremental learning of complex from basic affordances is presented in relation to learning of specific affordance features. We demonstrate the learning of causal relations between visual cues and associated anticipated interactions by reinforcement learning of predictive perceptual states. The work purses a recently presented framework for cueing and recognition of affordance-based visual entities that plays an important role in robot control architectures, in analogy to human perception. We experimentally verify the concept within a real world robot scenario by learning predictive features from delayed rewards, and prove that features were selected for their relevance in predicting opportunities for interaction.

I. INTRODUCTION

In the context of ecological perception proposed by J.J. Gibson [1], visual perception in terms of affordances would enable agents to experience in a direct way the opportunities for action. However, Gibson remained unclear about both how this concept could be used in a technical system and which representation to use. Neisser [2] replied to Gibson's concept of direct perception with the notion of a perception-action cycle that shows the reciprocal relationship of the knowledge (i.e., a schema) about the environment directing exploration of the environment (i.e., action), which samples the information available for pick up in the environment, which then modifies the knowledge, and so on. Our work on affordance-like perception is in the context of robotic systems based on a notion of affordances that 'fulfil the purpose of efficient prediction of interaction opportunities'. We extend Gibson's ecological approach under acknowledgment of Neisser's understanding that purposive visual feature representation on various hierarchies of abstraction are mandatory to appropriately respond to environmental stimuli. We take advantage of a refined concept of affordance perception by representing (i) an interaction component with affordance recognition (recognizing relevant events of interaction) and (ii) a predictive aspect (affordance cueing, i.e., predicting interaction via perceptual entities). This conceptual step enables lays a basis to identify the causal relation between predictive features and predicted event via machine learning technology.

The particular contribution of this work is to propose a framework for the developmental learning of affordances, and to frame the outline of basic affordances in terms of reinforcement learning. The work is in line with the concept to enable purposive - in particular, affordance based - perception which is consequently structured into cueing, behaviour, and outcome related components [22]. Learning is mandatory to enable agents to autonomously develop their characteristic embodied perception through interaction with the environment [3]. Reinforcements guide the development through exploration without external supervision, and a theory for estimating delayed reward is the appropriate framework to extract early cues.

II. COMPUTATIONAL MODELS ON AFFORDANCES

A. Related Work

Previous research on affordance-like perception focused on heuristic definitions of simple feature-function relations to facilitate sensor-motor associations in robotic agents [5,6]. The MIT humanoid robot Cog was involved in object poking and prodding experiments that investigate the emergence of affordance categories to choose actions with the aim to make objects roll in a specific way [7]. The research of Stoytchev [8] analyzed affordances on an object level and investigated new concepts of object-hood in a sense of how perceptions of objects are connected with visual events that arise from action consequences related to the object itself. In the biologically motivated cognitive framework of Cos-Aguilera et al. [15], object based affordances are set in the context of motivation driven behavior selection, however, visual feature extraction is not learnt in a purposive manner (Section II.B) but sensory input is rather matched with stored object features [16].

This work is funded by the European Commission’s projects MACS (FP6-004381) and by the FWF Austrian National Research Network ‘Cognitive Vision’ under sub-project S9104-N04.
Affordance based visual object representations are per se function based representations that aim at supporting control schemata for perception-action processing in the context of rapid and simplified access to agent-environment interactions [4,12]. In contrast to classical object representations, functional object representations (Stark & Bowyer [9], Rivlin et al. [10], Bogony & Bajcsy [11]) use a set of primitives (relative orientation, stability, proximity, etc.) that define specific functional properties. These primitives are subsumed to define, e.g., surfaces from the functional properties, such as ‘is sit-able’ or ‘provides stable support’. However, function based representations were so far basically pre-determined while it is particularly important to learn the structure and the features themselves from experience (Sec. III, IV).

B. Affordance based Perception and Learning of Affordances

Fig. 1 depicts the concept of feature based affordance perception as outlined in detail in [17]. We first identify affordance recognition, i.e., the recognition of the affordance related visual events that causally anticipate a relevant effect, e.g., the capability of lifting (lift-ability) an object using a robotic actuator. Recognizing this event means identifying a process of evaluating spatio-temporal information that leads to an outcome entity. This outcome entity should be characterized by the observation of specific feature attributes that are abstracted from the stream of sensory-motor information.

The second component of affordance cueing encompasses the key idea on affordance based perception, i.e., anticipating the opportunity for interaction from causally relevant features, i.e., predictive features, that are extracted from the incoming sensory processing stream. This component is embedded in the perception-action cycle of the robotic agent. In contrast to classical feature and object recognition, affordance based feature recognition is purposive in the sense of selecting exactly those features that efficiently support the evaluation of identifying an affordance, i.e., the perceptual entities that possess the capability to predict the events of affordance recognition. The outcome of affordance cueing is in general a probability distribution $P_A$ on all possible affordances (Section IV.A.), providing evidence for a most confident affordance cue by delivering a hypothesis that favors the future occurrence of a particular affordance recognition events. An overall consistent formal theory on affordances, describing agents with the capability to perceive functionalities for interaction, has been proposed by Doherty et al [24].

Purposive feature extraction must be performed in a machine learning process and therefore avoid heuristic engineering, a methodology which is necessarily both, firstly, error prone due to failing insight into statistical dependencies and, secondly, highly impractical for autonomously operating systems. Recent work on the learning of affordance features has focused on methodologies to estimate direct mappings between cues and actions from experience [17,23]. However, in the presented work we motivate from a developmental point of view (Section III) and outline a mathematical framework to extract affordance cues from arbitrary action sequences, i.e., from delayed rewards.

III. DEVELOPMENTAL LEARNING OF AffORDANCES

An agent embedded in its habitat is able to perceive the environment with its sensors and is able to move and manipulate this environment with its actuators. A structure enabling the robot to act on its perceptions by using its actuators is called control architecture. In case that the architecture causes actions, depending on the perceived state of the environment, a closed loop control emerges. The design of this control is essential for enabling the robot to use affordances; the proposed approach uses principles from the reactive control approach as well as from the subsumption based approach.

ABACUS We present ABACUS (Affordance Based Adaptive Control Using Self-Experience [22]), a multi layered conceptual framework, which enables the robot to use the concepts of affordances by taking it through several learning stages.

The fundamental functioning of the developmental learning is as follows (Fig. 2). In Phase 0, a control layer, implementing reactive behavior, is added to the structures sensor layer, filter layer, and actuator layer, and thus a basic reactive control is built. The reactive control of Phase 0 is then refined in Phase 1, where an adaptive structure learns to perceive and use basic affordances that are directly related to single action possibilities of the agent (e.g. gripping an object) and are thus mostly related to the object as a whole. The control developed in Phase 1 is refined by Phase 2 which is designed to learn to perceive and use more complex affordances that are related to sequences of actions (e.g. stacking, which consists of several interactions with 2 objects like gripping, lifting, driving and releasing) and are mostly related to object parts (e.g. a flat surface). The introduction of basic and complex affordances is our approach to enable an artificial agent to learn and deal with affordances of increasing complexity. The enhancement and refinement of each Phase $n$ through a succeeding Phase $n+1$ form a subsumption like affordance based control. The final scenario is realized by incrementally extending the Phase structures to gain an incrementally more complex affordance based architecture.

A. Phase 0

The sensor layer consists of physical sensors and software modules that are interfaces between soft- and hardware to enable the agent to receive raw data about the state of the environment, and the state of the robot itself. The filter layer is designed to reduce computing complexity and fault sensitivity.
within the control layer. Instead of using the original time series from the cameras, the laser-scanner or the positions of the robotic arm, the control layer can use more complex data extracted from the sensory input space, e.g. by filters for detecting simple geometric forms like rectangles, or descriptor based filters (Sec. V). Other examples for modules within the filter layer are motion detectors, which are used to enable the artificial agent to interact with a dynamic environment. The filter modules can also be cascaded, so that a hierarchy of filters is formed. Hence careful design of these filters is imperative for achieving the desired reduction of computing complexity without losing essential information. Within the filter modules attention mechanisms can be used to reduce the amount of data transferred to the next layer. By cascading the filters, attention mechanisms can also be applied to sets of filters, e.g. to focus attention to one sensor modality or one special filter.

B. Phase 1

Phase 1 is a structure that is designed to learn to perceive and use basic affordances that are directly related to single action possibilities of the agent (e.g. gripping an object) and are thus mostly related to the object as a whole (Fig. 3). Phase 1 utilizes the described sensor layer, filter layer, and actuator layer and refines the perception and control of Phase 0. Phase 1 consists of an affordance recognizer and an affordance based control module. The Phase 1 affordance recognizer learns to recognize basic affordances on the basis of the output of the filter layer and sensor layer. Data from the sensor layer is included to ensure that essential information is preserved if the abstract data coming from the filter layer is insufficient for learning in certain cases. The module learns what the outcome of an action is and learns what the cues to detect affordances without interacting with its environment are. For a detailed description of an algorithm that can be used to realize the adaptive Phase 1 affordance recognizer see Sec. IV. The information, extracted by the affordance recognizer is used within the Phase 1 affordance based control to trigger Phase 1 actions, or trigger or inhibit the basic actions that are implemented within the modules of the Phase 0 control layer.

Figure 2. (a) Sketch of the basic control structure underlying ABACUS. (b) ABACUS is a multi layered framework, which enables the robot to extract concepts of affordances through several learning stages. Phase 2 is designed to learn to perceive and use complex affordances by using the information received from the filter layer and the Phase 1 affordance recognizer.

Figure 3. Example for developmental learning: Basic actions within Phase 0 are activated by environmental triggers, which result, e.g., in a grasping reflex. After a learning process, the Phase 1 affordance recognizer gained knowledge about liftable and non-liftable entities, which, e.g., can be used for Phase 1 control to inhibit grasping of slippery objects or lifting of too heavy objects.

B. Phase 2

Phase 2 is a structure that is designed to learn to perceive and use complex affordances that are related to sequences of actions (e.g. stacking, which consists of several interactions with two objects like gripping, lifting, driving and releasing) and are mostly related to object parts (e.g. a flat surface enables the robot to stack something on it). Phase 2 extends the affordance recognition capabilities of Phase 1 and refines the affordance based control of Phase 1 and Phase 0. Like Phase 1, Phase 2 is subdivided in an affordance recognizer and an affordance based control module. By using data from the filter layer and the Phase 1 affordance perception output as an abstract and highly complex sensor, the Phase 2 affordance recognizer is enabled to learn complex affordances that require basic affordances to be present. The main focus of phase 2 lies on action sequences, e.g. behaviours lifting, stacking, or turning objects etc. that can be composed out of simple basic behaviours or motion primitives. Like in the case of basic actions, the outcome of sequenced actions is categorized and perceptual cues are searched that indicate the presence of affordances.

IV. REINFORCEMENT LEARNING OF BASIC AFFORDANCES

A. Affordance based Cueing

Early awareness of opportunities for interaction from remote locations, such as, from visual features, is highly relevant for autonomous robotic systems. Although the necessity of affordance perception from 3D information recovery, such as optical flow, has been stressed in previous work, we intend to generalize here towards the use of arbitrary features (2D, 3D).

1) Scenario The scenario for the experiments consists of a mobile robotic system (Fraunhofer IAIS, Germany [17]), equipped with a stereo camera pair and a magnetizing effector, and some can-like objects with various top surfaces,
colors and shapes. The purpose of the magnetizing effector is to prove the nature of the individual objects by lowering its rope-end effector down to the top surface of the object, trying to magnetize the object (only can bodies are magnetizable) and then to lift the object. Only test objects with well magnetizable geometry (with slab like top surfaces, in contrast to those with spherical top surface) are subject to a lifting interaction. This interaction process is visualized for several test objects and sampled in a sequence of image frames which are referenced with multimodal sensor information, e.g., size of magnetizing and motor current of the robot.

2) Visual Features From the viewpoint of a technical system using computer vision for image interpretation, we selected local descriptors, such as the Scale Invariant Feature Transform (SIFT [13]), to support the generation of visual feature abstractions. We first segmented the colour information and then associated classified histograms of descriptor responses - sampled within the regions - to the region feature vector. The histograms integrated responses from SIFT descriptors that were trained to discriminate either rectangular or circular region shapes [17].

3) Affordance Hypotheses The outcome of the affordance cueing system is then expected to be - given a perceptual entity in the form of a multimodal feature vector - a probability distribution over affordance hypotheses,

$$P_{di} = P(A_i|F_i),$$ (1)

with affordance hypothesis $A_i$ and feature vector $F_i$ at time $t$. It is then appropriate to select an affordance hypothesis $A_{max} = \text{arg max}(P(A))$, with Maximum A Posteriori (MAP) confidence support for further processing.

4) Cue-Features In the experiments, we extracted a set of features, such as, colour regions (colour G=green, R=red, M=magenta, etc.), descriptor (SIFT) category (R=rectangular, C=circular, etc.), and geometric information (TD/ top=T region, i.e., representing a region that is located on top of another region; B=bottom). Only SIFT category information together with a geometric feature provide the discriminative features that would allow to predict the future outcome (e.g., lift-able or non lift-able) of the affordance recognizer.

B. Reinforcement Learning

In the following, we describe an implementation of developing control Phase 1 (Sec. III) of affordance learning, by encapsulating Phase 0 in terms of a reactive behaviour that is modelled within the reinforcement learning of the overall affordance liftability. Reinforcement learning [21,19,15] is the appropriate methodology to learn early cues, i.e., affordances, that are capable to predict opportunities for future interactions and their associated consequences in terms of rewards.

Markov decision processes [18] (MDPs) are the mathematical framework underlying reinforcement learning and have already applied in perception-action based visual recognition [19,20]. The MDP will provide the general framework to outline a multi-step behavioural task under the viewpoint of state based prediction / cueing of future outcomes of that task.

1) Markov Decision Processes An MDP is defined by a tuple $(S,A;\delta;\mathcal{R})$ with state recognition set $S$, action set $A$, probabilistic transition function $\delta$, and reward function $\mathcal{R}: S \times A \rightarrow \mathcal{R}(S)$ describes a probability distribution over subsequent states, given action $a \in A$ executable in state $s \in S$. In each transition, the agent receives reward according to $\mathcal{R}: S \times A \rightarrow \mathcal{R}, \mathcal{R}_b \in \mathcal{R}$. In our experimental scenario, the agent must act to maximize the utility $Q(s,a)$, i.e., the expected discounted reward

$$Q(s,a) = U(S,a) = E\left[\sum_{\tau=t}^{\infty} \gamma^\tau \mathcal{R}_{t+\tau}(s_{t+\tau},a_{t+\tau})\right]$$

(2)

where $\gamma \in [0,1]$ is a constant controlling contributions of delayed reward. We formalize a sequence of action selections $a_t, a_{t+1}, ..., a_n$ as an MDP and are searching for optimal solutions with respect to finding action selections so as to maximizing future reward with respect to the affordance task. With each action, an estimate on the cumulative reward gives feedback about the direction towards the goal of the task. With each action, the reward is received per action by $R(s,a) := \Omega$, with $\Omega=1$ if the goal event is reached (object lifted into goal image zone), and $\Omega=0$ if not. Since the probabilistic transition function $\Pi(\cdot)$ cannot be known beforehand, the probabilistic model of the task is estimated, e.g., by Q-learning [21] which guarantees convergence to an optimal policy.

The Q-function update rule is

$$Q(s,a) = (1-\alpha)Q(s,a) + \alpha \left[ R + \gamma \max_{a'} Q(s',a') \right]$$

(3)

where $\alpha$ is the learning rate, $\gamma$ controls the impact of an action on future policy returns. The decision process is determined by the sequence of actions. The agent selects then the action with largest $Q(s,a)$ at time instant $T$, i.e.,

$$a_t = \text{arg max}_{a'} Q(s_t,a')$$

(4)

so as to maximize the cumulative expected reward $Q(s,a)$.

2) Actions and States In the selected scenario, actions are defined by discrete steps of gripper motion (up, down) and magnetization (on, off). Each perceptual state is defined by a discrete feature configuration

$$S = \{c \in \mathcal{C}, d \in \mathcal{D}_c, e \in \mathcal{C}_c, h_i \in \mathcal{R}, h_g \in \mathcal{R}, m \in \mathcal{M}\},$$

with region color class $c$, region shape $d$, configuration type $e$, region elevation level $h_i$, gripper elevation level $h_g$, and magnet state $m$. The affordance based, purposive selection of features is here represented by relevance weighting of perceptual states in terms of associated expected rewards. Each extracted image region is attributed by a perceptual state. Perceptual states that anticipate a complete trajectory of perception-action transition leading to the affordance outcome event with high certainty will be associated to high rewards and consequently constitute cue like states. Affordance based perceptual states are implicitly related to the reward function, an associated estimator on affordance hypotheses $P_{\mathcal{A}}$ and an affordance classifier discriminating corresponding $Q$-values into affordance cues and irrelevant states.
V. PROOF OF CONCEPT

The experiments were performed in a real world robot environment with the purpose of providing a proof of concept on the successful learning of predictive 2D features, i.e., affordance based cues, and on characterizing affordance recognition processes.

A. Scenario
Robot operations are discriminated into two phases, (a) the cueing phase where the robot is moving to the object, and (b) the recognition phase, where the robot tries to lift an object (Fig. 1). In both phases, parts of objects are described by their regions attributed by features like colour, centre of mass, top/bottom configuration and shape description (rectangular, circular); features are extracted from the robot camera imagery. Additional information, such as, effector position, are provided by the robot. Regions are the entities used in the experiments, no explicit object model is generated for the can-like objects.

B. Affordance Recognition and Cueing
The recognition of an affordance is crucial for verifying a hypothesis about an affordance $A$ associated with an entity feature $F$. These entities are specifically extracted out of the images as follows. Firstly, a watershed algorithm is used to segment regions of similar colour together. After merging of smaller parts, every entity is represented by the average colour value, the position in the image and the relation to adjacent regions (top/bottom). This information is also used for tracking entities over time. To verify whether or not an entity becomes ‘lift-able’, the magnetizable effector of the robot is lowered until the top region of the object under investigation is reached, the magnet is switched on and the effector is lifted up. If the entity is lift-able, a common motion between effector and region can be recognized, and both can and gripper regions are undergoing a vertical transition (direction up) in the field of view. Additionally the magnet has to be switched on and the effector has to be placed in the centre of the top region. These rules in the recognition phase of the experiment (Fig. 4, right) build up the affordance recognizer looking for lift-able entities.

C. Reinforcement Learning of Predictive Features
Affordance cueing and recognition may require different kinds of feature extraction. For cueing, some structural description of the top region is required to separate the unequal shape of the top regions. In order to get structural information about an entity a histogram over prototypical SIFT descriptors is used to discriminate between circular and rectangular regions.

D. Structural Classification.
Local SIFT descriptors extracted in the entity region are clustered using unsupervised clustering ($k$-means, $k=100$). For each specific entity, we generate a histogram over cluster prototypes, using a nearest neighbour (NN) approach to get the cluster label for each SIFT descriptor in that region. In a supervised learning step, every histogram is labelled whether it is or not associated with a rectangular or circular entity. A C4.5 decision tree [14] of size 27 is then able to distinguish between these two classes. The error rate on a test set with 353 samples is $\approx 1.4\%$.

D. Q-learning, decisive states, and affordance based cueing
Images about the objects that were tested for the affordance ‘lift-able’ in the recognition phase collect positive rewards that trace back to early perceptual states due to the Q-learning update rule. As mentioned earlier, there exists no object model yet, therefore only entities exist for the system, and the learning of cueing states is with respect to the region extraction determining the perceptual states (time span of the experiment is $\approx 2.5$ seconds). The entity representation for the cueing phase contains the following features: (a) average...
colour value of the region in the image, (b) top/bottom information, (c) the result of the structure classification, (d) the size of the segmented region. Fig. 6a depicts the learning curve resulting from the reinforcement learning phase, with respect to the predicted cumulative reward associated to an early perceptual state that thereby is verified to represent a ‘cueing’ state. Fig. 5d depicts results for predicted cumulative rewards regarding observed regions (top, bottom; Fig. 5a-c) reflecting different evaluation of perceptual states towards ‘cueing’ (green bar) and ‘non-cueing’ (red bar) states. Fig. 6b shows the dynamic view on step-wise classification of regions under predictive cueing, e.g., predictive features (green) were detected all through the sequence of 45 successive frames.

E. Extraction of Cue Features

The experimental results that prove the reinforcement learner identified the actually relevant features was finally received from a statistical analysis of perceptual states. A C4.5 decision tree [14] was learned to estimate the decisive attributes that enable classification of perceptual states into affordance cues \( Q(s,a)>0 \) and irrelevant states \( Q(s,a) \leq 0 \). Features found were CR<0.5 (‘rectangular’) and TD<0.5 (‘top region’) which exactly defines the decisive features as outlined in Sec. IV.A (cue-feature value matrix).

VI. CONCLUSION

This work presents the framework of reinforcement learning for perceptual cueing to opportunities for interaction of robotic agents. The framework for cueing and recognition of affordance-like visual entities is verified with a concrete implementation using state-of-the-art visual descriptors on a real world robot scenario and proved that features are successfully selected that are relevant for prediction towards affordance-like control in interaction. The real world robot environment was chosen to enable a proof of concept in terms of learning to select exactly those features that are relevant for the prediction of the interaction outcome. In this line of research, we think that the presented reinforcement learning provides the appropriate methodology to motivate the learning of functional object recognition, grounding thereby the object notion in a concept of predictive feature abstractions.

REFERENCES