In: Institute of Electrical an Electronics Engineers: IEEE/RSJ International Conference on Intelligent Robots and Systems: IROS 2007, San Diego, CA, USA ; proceedings. Piscataway, NJ [u.a.]: IEEE, 2007, S. 3385-3390

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].

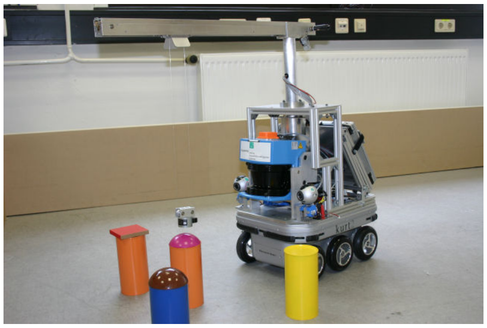


Fig. 1. Demonstrator scenario of the EU project MACS [4]. A goal of MACS is to explore affordance-inspired perception and learning for mobile robots related to J.J. Gibson's theory. The shown robotic agent KURT3D should perceive, learn and utilize its environment in a functional way.

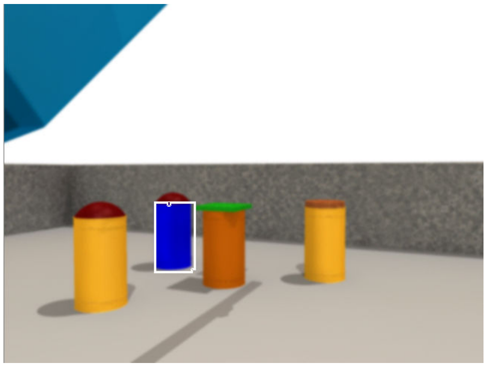
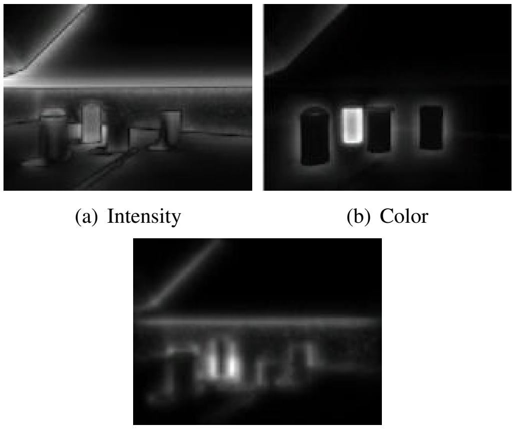


Fig. 2. Visual attention: The most salient region is selected by bottom-up attention. Regions are determined, which are highlighted from the remaining scene through their color, brightness or orientation, here the blue can.



(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 bit precision on VGA images.

III. Related Work

In the first part of this section practical works on implementing the affordance approach in the field of cognitive systems are described. In the second part we give an account on the most significant implementations of visual attention systems.

A. Affordance-inspired Robotics

Gibson once stated that invariants are directly perceived in the optical array. This is one of the most controversial aspects of the affordance theory. In this context cognitive approaches can be divided into two categories: those which are symbolic based and those which are not. Seminal approaches like those of Duchon et al. [10], Warren [11] and Mark [12] showed that also complex tasks can be solved using a non-symbolic base. But there is still no non-symbolic approach, which is capable of learning affordances related to a diverse set of action possibilities using a variety of sensorimotoric capabilities and a complex environment. Proposing the necessity of symbols, MacDorman stressed that Gibson underestimated the computational complexity of vision [13]. He argued that the complexity can only be handled on a higher degree of abstraction. The preprocessing stage in his approach handled the amount of data by transforming perceived data into a canonical form while reducing the data to a 64-by-64 grid. The use of a wavelet transform parametrized with two dimensional Gabor filters has its seeds in neurophysiology. It is obvious that the computational effort was also an important reason for the data reduction and filter design. Thus it supports our argumentation of combining biologically inspired algorithms with massive parallelism on graphics rendering devices.

B. Attention in Robot Perception

The use of visual attention in robotics is reported to be advantageous for many purposes, like automated target detection, human machine interaction and so on [14][15][2]. It is inspired by the morphology of the human visual system. Most approaches report a high computational demand. Based on a popular visual attention system, the Neuromorphic Vision Toolkit (NVT), Itti et al. presented a parallel implementation performing real-time attention cue computation on video streams [14]. To reach a sufficient speedup the system needed to run on a 16-CPU Beowulf cluster. The attention system of the robot Kismet [15], designed for social interaction with humans, was also processed on a parallel computer. It was attached to a DSP (Digital Signal Processor) network, which computes the saliency maps for color and motion on different DSP nodes. Comparatively new to above-named approaches is the system VOCUS (Visual Object detection with a CompUtational attention System) presented in [2]. It processes foci of attention (FOA) sequentially in contrast to the others, but it has been optimized by measures like the use of integral images. This system was also reported to be real-time capable [7]. The term "real-time" in each 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: $\approx$ 3.6 GFlops vs. Pixelshader of NVidia GeForce 7800 GTX 256MB: $\approx$ 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

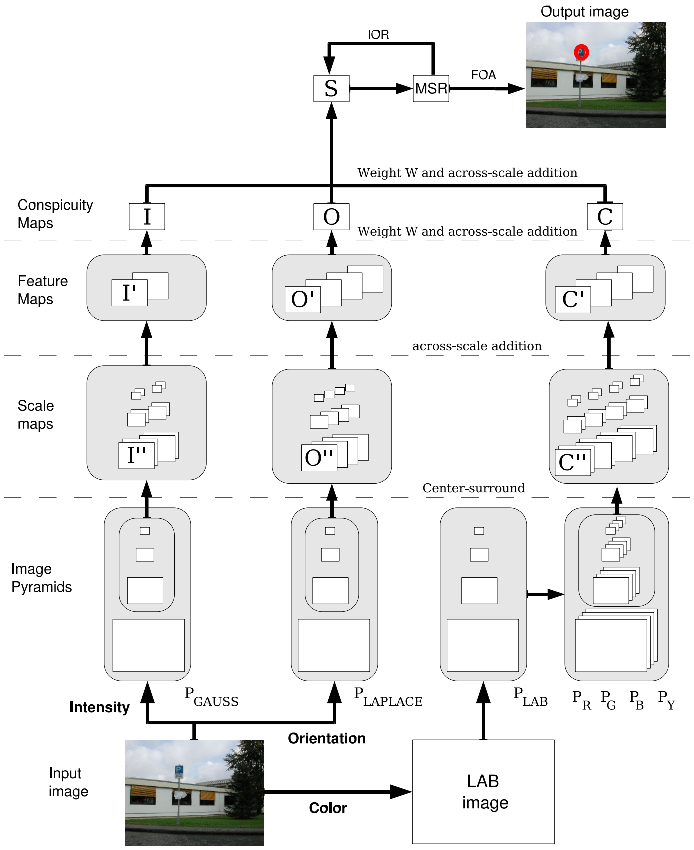


Fig. 4. Overview of the attention system GPU-VOCUS. The diagram is reprinted from [2]

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 themself. 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/nonstackable, ...) [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 “liftability”. The crane of the robot enables only a few manipulation tasks, but these 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.

CPU	Pentium D 3.0 GHz
Main memory	1024 MB DDR2 PC533
Graphics device	NVidia GeForce 7800 GT / 8800 GTX
OS	SuSE Linux 9.3, Kernel 2.6
OpenGL version	2.0
Cameras	Logitech Quickcam Pro 4000 (15 Hz at VGA)

TABLE I

HARDWARE/SOFTWARE SPECIFICATION FOR THE EXPERIMENTAL SETUP

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 does not deal with integral images. Orientation maps are calculated only when needed. This depends on the use case. If the emphasis will be on color and intensity features, the disabling of orientation features results in a shorter runtime. It is also worth mentioning that only the orientation maps need a computation with the precision of 32 bit. The results of all other maps show only negligible differences between a computation with 16 bit and 32 bit precision. Using 16 bit precision was necessary to fulfill the needed performance constraint of 30 Hz (15 Hz for each of both cameras) on the tested GeForce 7 device. The GeForce 8 device fulfilled this constraint even with 32 bit precision and achieved a speedup of 6 to 9 compared to the optimized VOCUS version using integral images (cf. table II). Supportingly the computational load is removed from the CPU, which only acts as data provider for the GPU and is now free for remaining recognition and control tasks.

	Mean Runtime / ms	
Feature orientation	yes	no
VOCUS (non-integral)	1407.6	969.7
VOCUS (integral)	129.2	89.1
GPU-VOCUS (NV GF 7800 GT / 32-bit)	77.5	34.3
GPU-VOCUS (NV GF 7800 GT / 16-bit)	57.7	25.0
GPU-VOCUS (NV GF 8800 GTX / 32-bit)	21.8	9.6

TABLE II

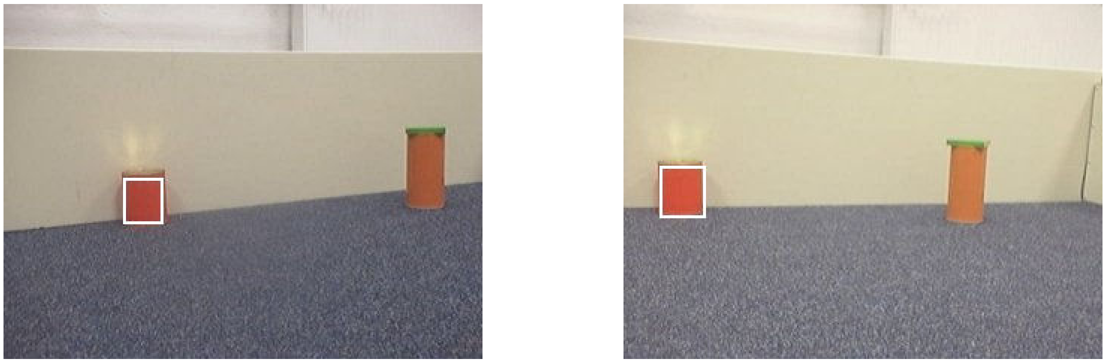
MEAN RUNTIME OF (GPU-)VOCUS (20 RUNS/VGA RES.)

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:



(a) Left cam

(b) Right cam

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

[1] J. J. Gibson, *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, 1979.

[2] S. Frintrop, *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search*, ser. Lecture Notes in Artificial Intelligence (LNAI). Springer Berlin/Heidelberg, 2006, vol. 3899 / 2006.

[3] E. J. Gibson and A. D. Pick, *An Ecological Approach to Perceptual Learning and Development*. Oxford University Press Inc, USA, 2000.

[4] Fraunhofer IAIS. (2007) EU Project MACS. [Online]. Available: http://www.macs-eu.org

[5] G. Fritz, L. Paletta, R. Breithaupt, E. Rome, and G. Dorffner, "Learning Predictive Features in Affordance-based Robotic Perception Systems," in *Proeedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2006.

[6] J. E. Cutting, K. Springer, P. A. Braren, and S. H. Johnson, "Wayfinding on foot from information in retinal, not optical, flow." *J Exp Psychol Gen*, vol. 121, no. 1, pp. 41–72, Mar 1992.

[7] S. Frintrop, M. Klodt, and E. Rome, "A Real-time Visual Attention System Using Integral Images," in *Proceedings of the 5th International Conference on Computer Vision (ICVS 2007), Bielefeld, Germany*, March 2007.

[8] C. Bartolozzi and G. Indiveri, "A selective attention multi–chip system with dynamic synapses and spiking neurons," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. Cambridge, MA: MIT Press, 2007.

[9] L. Itti and C. Koch, "Computational Modelling of Visual Attention," *Nature Reviews Neuroscience*, vol. 2(3), pp. 194 – 203, 2001.

[10] A. P. Duchon, W. H. Warren, and L. P. Kaelbling, "Ecological robotics," *Adaptive Behavior, Special Issue on Biologically Inspired Models of Spatial Navigation*, vol. 6, no. 3-4, pp. 473–507, 1998.

[11] W. Warren and S. Whang, "Visual guidance of walking through apertures: Body scaled information for affordances," 1987, vol. 13, pp. 371–383.

[12] L. Mark, "Eyeheight-scaled information about affordances: Lerning and projecting a sersori-motor mapping," 1987, vol. 13, pp. 361–370.

[13] K. MacDorman, "Grounding symbols through sensorimotor integration," 1999.

[14] University of Southern California, iLab and Prof. Laurent Itti. (2001) Visual Attention - Ongoing Projects. [Online]. Available: http://ilab.usc.edu/bu/ongoing/

[15] C. Breazeal and B. Scassellati, "A Context-Dependent Attention System for a Social Robot," in *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 1146–1153.

[16] S. N. Sinha, J.-M. Frahm, M. Pollefeys, and Y. Genc, "GPU-Based Video Feature Tracking and Matching," in *EDGE 2006, workshop on Edge Computing Using New Commodity Architectures, Chapel Hill*, 2006.

[17] M. C. L. Naga K. Govindaraju, Stephane Redon and D. Manocha, "CULLIDE: Interactive Collision Detection Between Complex Models in Large Environments using Graphics Hardware," in *ACM SIGGRAPH/Eurographics Graphics Hardware, San Diego, CA*, 2003.

[18] A. M. Treisman and G. Gelade, "A feature-integration theory of attention." *Cognitive Psychology*, vol. 12, no. 1, pp. 97–136, January 1980.