Demand-Driven Visual Information Acquisition

Sven Rebhan, Andreas Richter, Julian Eggert

2009

Demand-Driven Visual Information Acquisition

Sven Rebhan, Andreas Richter, and Julian Eggert

Honda Research Institute Europe GmbH,
Carl-Legien-Strasse 30,
63073 Offenbach am Main, Germany

Abstract. Fast, reliable and demand-driven acquisition of visual information is the key to represent visual scenes efficiently. To achieve this efficiency, a cognitive vision system must plan the utilization of its processing resources to acquire only information relevant for the task. Here, the incorporation of long-term knowledge plays a major role on deciding which information to gather. In this paper, we present a first approach to make use of the knowledge about the world and its structure to plan visual actions. We propose a method to schedule those visual actions to allow for a fast discrimination between objects that are relevant or irrelevant for the task. By doing so, we are able to reduce the system's computational demand. A first evaluation of our ideas is given using a proof-of-concept implementation.

Key words: *scene representation, scheduling, attention, memory*

1 Introduction

Cognitive systems are surrounded by a vast amount of (visual) information. To acquire the currently relevant information is a challenge for both biological and technical systems. But how do we decide what is relevant? How many details of the current scene do we process? Which locations in the scene contain information we need? And what information do we need to store about these locations?

Already the work of Yarbus [1] showed that the task one currently performs has an outstanding role in determining what is relevant. In his work, Yarbus showed that the scanpaths of a human observer on a photo vary dramatically, dependent on the task. But how does the task influence our perception of the scene? To get an insight into this question many experiments were performed. The so called "change blindness" experiments revealed that, even though our subjective perception tells otherwise, only parts of the scene are perceived (e.g. [2]). *Where* we look is determined by the task and the knowledge about both the current scene and the world [3, 4]. However, a very important question remains: Which details are stored about a visited location? In [5] an experiment was conducted, suggesting that only those object properties relevant for solving the current task are stored in memory. The subjects were blind to changes of other properties of the object. This experimental evidence was confirmed by later experiments [6]. The psychophysical experiments show that we perceive the visual vicinity only partially. *What* details we perceive is also determined by the current task and our knowledge. Here, attention is a crucial aspect [7] and guiding this attention is assumed to be an active process [8].

First models for guiding attention were proposed under the names *active perception* [9], *active and purposive vision* [10] and *animate vision* [11]. Although the idea behind these approaches is more general, these models mainly focus on the modulation of sensor parameters in order to guide attention. However, the results show that using an active system it is possible to solve some problems that are ill-posed for a passive observer. In newer approaches on scene representation, more elaborated attention control mechanisms were implemented [12]. In these models the long- and short-term memory (LTM & STM) of the system is used along with the gist of a scene to accumulate task-relevant locations in a map. The memorized properties of the objects are used to bias the low-level processing in order to speedup the visual search.

However, all models mentioned focus solely on the spatial aspect of attention. That is, they use the world and scene knowledge to determine *where* to look. Once they have focused on a certain location, the complete feature vector is stored in the STM. This contradicts the experiments showing that only the task-relevant properties are stored for an object. It is our goal to build a cognitive vision system that also accounts for this aspect of vision. Thus, it must be able to selectively acquire information in both the spatial *and* feature domain to acquire only the information relevant for solving the current task. For example: If the task requires to know the color of an object, we only want to measure and store the color of the object. If the task requires to identify the object, we only want to acquire the minimal set of information that identifies the object and so on. Here, the static processing pathways of all state-of-the-art models do not hold anymore. Rather, a more flexible solution is required that allows to dynamically "construct" a processing pathway. However, this flexibility raises a new fundamental question [13] not tackled in current approaches: In which order should the system execute visual routines to acquire information?

In this paper, we concentrate on exactly this question and give a first idea on how a scheduling algorithm for visual routines could look like. We propose a method that incorporates knowledge about the task, the world and the current scene to determine which information is relevant. To decide in which sequence visual routines should be executed, the attention guidance process itself needs to carefully plan the utilization of the system resources, taking the cost and gain of each operation into account. In this work, we concentrate on simple search tasks, as they are often a basic atomic operation for other, more complex, tasks.

In section 2, we briefly present our system architecture. Afterwards we propose a memory architecture (section 3) that accounts for both the special needs of our scheduling process and the generic representation of knowledge. In section 4, we describe our scheduling algorithm used to control attention in the spatial and feature domain. We show first results using a proof-of-concept implementation and close with a discussion and an outlook in section 6.

2 System Architecture

In order to investigate the execution sequence of visual routines, we need a flexible system architecture as mentioned before. Such a flexible architecture was first proposed in [14], where different elementary visual routines are called on demand. Our system architecture as shown in Fig. 1 is based on this work and comprises four major parts: a relational short- and long-term memory, the attention control, a tunable saliency map and visual routines for extracting different object properties. The relational memory stores the knowledge about the world (LTM) and the current scene (STM). We will give a more detailed view on the memory in section 3. The focus of this paper is the attention control, as it determines which locations and features are attended (for details see section 4). Furthermore, a saliency map is used to find the objects in the current scene. By doing so, it can use top-down information to speedup the visual search task similar to [15]. Finally, the system comprises a bank of visual routines, each of them specialized to determine a certain property of a focused object [16]. The execution of a visual routine is selectively triggered by the attention control mechanism. Currently, our system comprises three elementary visual routines for measuring the color, the disparity-based distance z from the camera (calibrated stereo setting) and a pixel mask of an object. Along with the object mask we store its rectangular bounding box, having a width of w and a height of h where we define $w \geq h, \forall(w, h)$. Based on these properties, more complex ones like the position in the three-dimensional space $\boldsymbol{x}$, the physical size s and a coarse shape r can be calculated. Here, the physical size is defined as $s \propto w/< z >$ [1], and the coarse shape is defined as the aspect ratio r of the bounding box $r = h/w$.

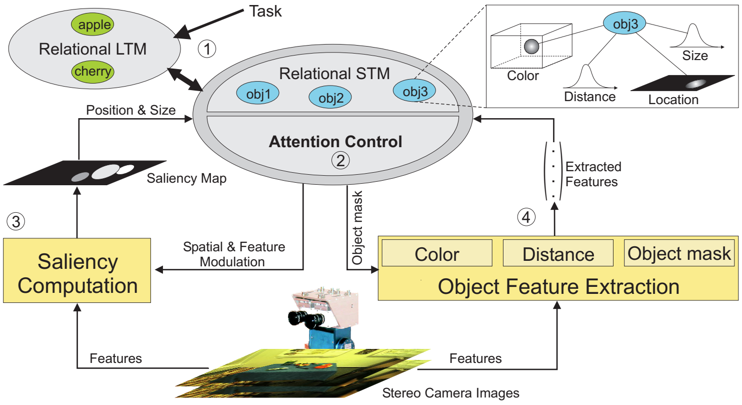

Fig. 1. The overall system architecture mainly consists of four parts: a relational memory(1), the attention control(2), a tunable saliency(3) and several feature extraction components(4).

[1] With $< z >$ being the averaged distance z using the object mask.

3 Memory Architecture

In our approach, the system's memory does not just serve as a "data store" for the world knowledge. More importantly, it provides a suitable representation for deciding which properties are characteristic for the different objects. A flexible and general memory architecture, fulfilling our requirements, was proposed in [17], which we use as a basis for our implementation as shown in Fig. 2. This memory architecture allows for freely definable link patterns, inheritance of information and hypothetical nodes in both short- and long-term memory. It is important to note that property nodes are "anchored" by storing direct links to the sensory representation (see Fig. 2). Figure 2 shows that all nodes are equivalent. The role of the node is entirely defined by its incoming and outgoing links. These properties of the memory architecture distinguish the chosen memory architecture from standard AI models. In the following illustrations, we merge the labels attached to the nodes into the node names for better readability.

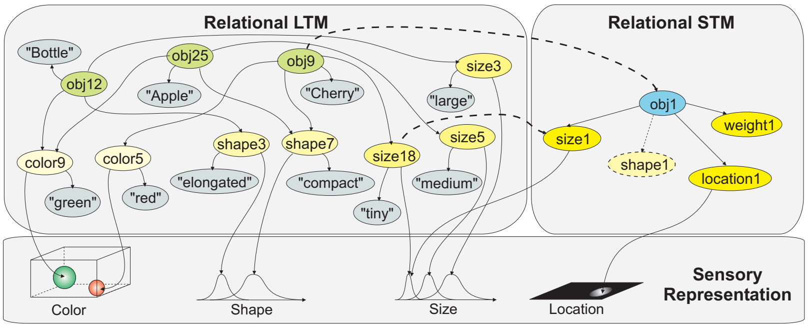

Fig. 2. The memory architecture allows a relational representation of knowledge with an identical structure for both STM and LTM. Nodes representing visual properties are "anchored" by storing direct links to sensory representations. Nodes can inherit information from other nodes (dashed line) and represent hypotheses (see *shape1*).

Additionally to storing knowledge about the world and the current scene, in our case the LTM also stores knowledge about the process of acquiring information (see [18] for details). For example, if we want to measure the color of an object, we first need to know where the object is and which dimensions it has. This dependency on other properties is consistently represented as links between those property nodes in the LTM. As both STM and LTM share the same object structure, transferring information is straightforward. When searching for a certain object in the current scene, a hypothetical object is instantiated in the STM (see *obj1* in Fig. 2). The object instance inherits (dashed line) all object properties from the long-term memory and thus can access these properties as predictions (see *shape1*). Using the visual routines, the predictions can be confirmed on demand (see *size1*). The scheduling of visual routines to confirm property predictions is the task of the attention control.

4 Attention Control and Scheduling

We now want to focus on the key element of this paper, the attention control mechanism. So what is the role of attention? As mentioned earlier, we understand attention as a selection process, deciding where to look and which details to store about that location. So the problem is twofold. First, there is a spatial aspect of attention, namely to locate object candidates in the current scene. A lot of work has been done in this direction, the probably most prominent one is [15]. The authors state that modulating low-level features using knowledge about an object can speedup visual search. Once focusing on a location the system needs to assure that the attended object has all properties requested by the task. This leads to the second, not well researched aspect of attention: attention in the feature domain. The system needs to acquire the information *relevant* for solving the current task. But how does it know what is relevant? For tasks already containing a hint on which property is relevant, the system can simply trigger the respective visual routine. If the task is to "find a small object", the system immediately knows that it needs to analyze the size of an object candidate.

However, for finding a specific object the procedure is more complex. In order to keep the computational and storage demand low, the goal is to find the minimal set of measurements ensuring that the attended object is the searched one. This way, the amount of information that needs to be stored in the STM and the computation time are minimized. In our approach the system uses its LTM knowledge to determine characteristic properties of the searched object. Please note that the discriminative power of a certain property strongly depends on concurrently active object hypotheses. In Fig. 3, the system has to search an apple and knows that an apple is "green", "small" and "compact". Now the system must decide on which property it wants to focus. If it measures the color "green", there are two valid hypotheses (bottle and apple), for the size "small" also two hypotheses remain (lemon and apple) and for the shape "compact" four hypotheses remain (see Fig. 3a). So the gain is highest for the color and the size measurements, as they reduce the set of possible interpretations most. Now a second factor comes into play, the cost of a certain measurement. Here, we interpret the computation time of a certain visual routine as the cost of the measurement and store this time for each visual routine in the system. In our system the color measurement is faster than the size measurement, so the attention control decides to measure the color. As you can see in Fig. 3a, an object (*obj1*) is predicted to have the color "green" (*color1*). To measure the color of an object, one first needs to locate an object candidate using the saliency map (*location1*). See [18] on how these dependencies are resolved by the system. After confirming the color "green", only one further object hypothesis (bottle) beside apple remains as shown in Fig. 3b. As a consequence of the color measurement, most hypotheses were rejected and the discriminative power of both the size and shape increased. Now, either measuring the size "small" or the shape "compact" would uniquely confirm the object candidate to be an apple. Again, the speed of the visual routines biases the selection. For our system the measurement of the size is faster, so the prediction that the focused object is small is added to

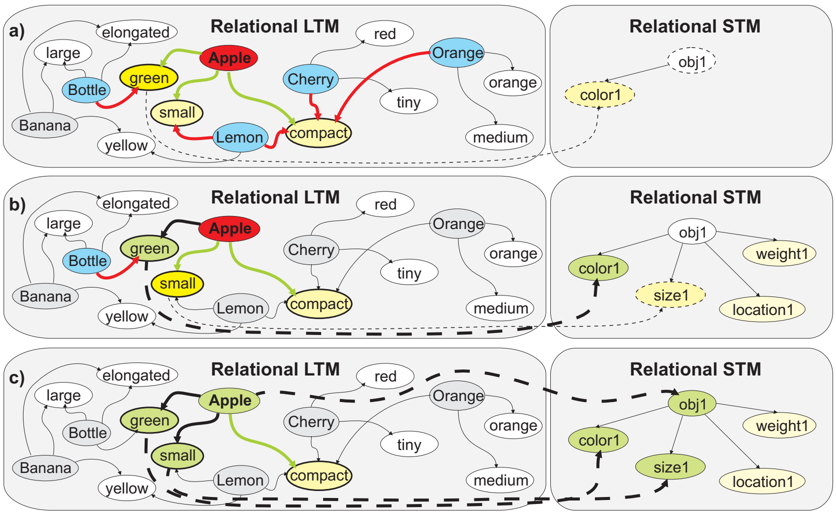

Fig. 3. a) To search the apple it is activated in the LTM (red). By propagating the activation, attached sensory representations are activated in the LTM (yellow). Propagating the activation further triggers competing object hypotheses (blue). b) After measuring the color of the object candidate, two hypotheses (bottle and apple) remain. c) By measuring the size the object candidate is confirmed to be the apple.

the STM (*size1*). After confirming the size, only the apple remains as a valid object hypothesis and is thus linked to the object candidate (see Fig. 3c). If a measurement contradicts the searched object, another object candidate will be located. To formalize our approach, we use the following notation: The LTM graph $G = (V, E)$ consists of the object nodes O, property nodes P where $V = O \cup P$ and edges E. In summary, the scheduling works as follows:

1. Locate an object candidate o_c using the saliency map and set $O_r = O$.
2. Activate the searched object $o_s \in O$ and collect its attached properties $P_s = \{p \in P | (o_s, p) \in E\}$.
3. Find all remaining competing object hypotheses O_h sharing properties with the searched object $O_h = \{o \in O_r | \exists p \in P_s : (o, p) \in E\}$.
4. Calculate the discriminative power $d_i = |D_i|^{-1}$ against the remaining hypotheses where $D_i = \{o \in O_h | \exists (o, p_i) \in E\}$, $\forall p_i \in P_s$.
5. Trigger the visual routine on the object candidate o_c for the most discriminative property $p_i : d_i \leq d_j \forall j$. If multiple properties minimize the set, select the fastest one. Remove the selected property from the set $P_s = P_s \setminus p_i$.
6. Find the property node $p_m \in P$ that matches the measurement best and determine the attached objects $O_m = \{o \in O | \exists (o, p_m) \in E\}$. Calculate the remaining objects $O_r = O_h \cap O_m$. If the search object is rejected $o_s \notin O_r$, go to step 1. Otherwise, if $|O_r| > 1$, continue with step 3. For $O_r = \{o_s\}$ we have found the object.

5 Results

To test our scheduling algorithm, we have implemented a proof-of-concept system with a hand-crafted LTM. We use an artificial visual scene with precomputed sensor values to neglect sensor noise and gain more control over the scene. However, in [19] we have shown that the memory architecture and the visual routines are capable of dealing with real sensory data. In a first experiment, the system's task is to search for a cherry. The content of the long-term memory as shown in Fig. 4a is the same for all following experiments. The system interprets the search task by activating the cherry in the LTM (marked red in Fig. 4a). We have observed, that by spreading this activation in the memory the properties "red", "compact" and "tiny" were activated. As Fig. 4b shows, the system decided to measure the color of the object first (marked yellow), because red is a unique property identifying the cherry in the system's LTM. The computed location of an object candidate was stored together with its measured color in the STM (see Fig. 4b). The system identified the measured color *color_1* as "red" using a nearest neighbor classifier (dashed line in Fig. 4b). Figure 4c shows that after this measurement the system classified the object candidate (*obj_1*) as a cherry (dashed line). Starting from this point, the system could predict further properties of that object like its size or shape. The system only stored one property for this object in its STM, where other systems would have stored the property vector containing three elements.

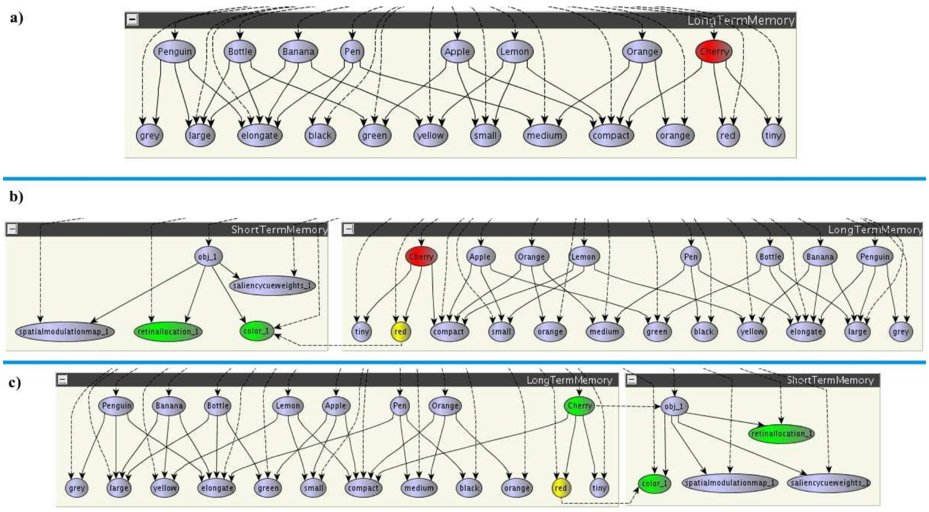

Fig. 4. a) Search for a cherry. Based on the long-term memory, the system knows cherries are red. b) The saliency map selects an object candidate and its color is measured. c) Because object *obj_1* is red, it is identified as a cherry. See the text for the color code.

In a second experiment, the system's task was to search for an apple (see Fig. 5a). For better visualization, we reset the STM. In Fig. 5b we observed that

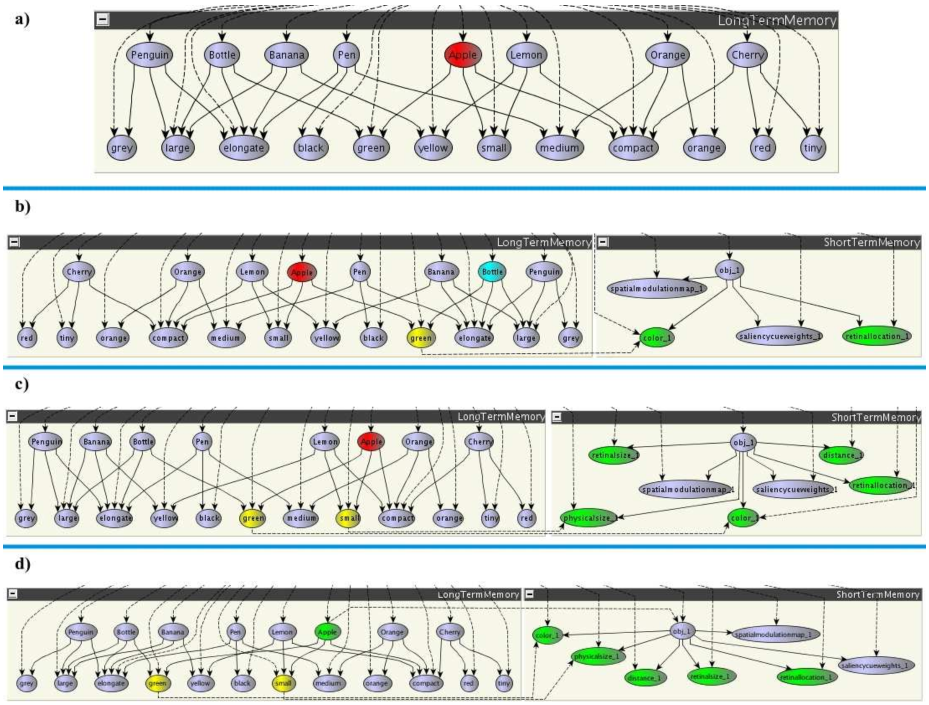

Fig. 5. a) Search for an apple. b) First, the color of the object candidate is measured to minimize the set of remaining hypotheses. c) Then, the need to distinguish between the bottle and the apple triggers the measurement of the size. d) Finally, the object *obj_1* is identified to be an apple.

the system decided to measure the color first. It did so, even though two possibilities were given because both the color "green" and the size "small" would trigger two hypotheses. This decision was due to the fact that the predefined computation time (cost) was smaller for the visual routine measuring the color. Figure 5b shows that a green object was found in the scene (dashed line). Furthermore, a second hypothesis (bottle) beside the apple remained (see Fig. 5b). The system started another refinement step as shown in Fig. 5c. Here, the system decided to measure the size "small" to distinguish between the remaining hypotheses. Again the system chose the faster visual routine although a shape measurement would have identified the object. Figure 5c shows that the object candidate was indeed small, which only left the apple as a valid hypothesis. The system identified object *obj_1* as an instance of the apple (see dashed line in Figure 5d). Again the system only stored the minimal number of properties required to identify the object.

To emphasize the memory and computation savings of our algorithm, we measured the number of computed and stored properties for all objects in the LTM (see Table 1). As current state-of-the-art models (e.g. [15]) always store the complete property vector, they always perform three measurements. Compared to this, our algorithm performs only half of those measurements on average. Along with the saving of computation time, the required memory size is reduced because only those properties measured are stored. Of course, the number of required measurements depends on the memory structure, nevertheless, in a worst-case scenario, the number of measurements is identical to current models.

Object	Penguin	Bottle	Banana	Pen	Apple	Lemon	Orange	Cherry	Ø
without scheduling	3	3	3	3	3	3	3	3	3.0
our approach	1	2	2	1	2	2	1	1	1.5

Table 1. Comparison of properties measured per object.

6 Discussion

In this paper, we presented a system that contrary to state-of-the-art models selects both the locations *and* the features it attends. Furthermore, our proposed scheduling algorithm actively triggers visual routines based on the knowledge about the object to search and its LTM. The goal is to minimize the set of hypotheses applicable to a certain object candidate. This way, the number of measurements and thus the amount of data stored in the STM is reduced to the information necessary to solve the current task. We proposed that in situations where more than one visual routine leads to the same minimal size of the hypotheses set, the costs (in our case the computation time) of the different visual routines are taken into account. In this paper, the cost parameters where chosen by hand, representing the approximated computation time of the different visual routines.

In future work, the performance of our system needs to be tested on real-world scenes. One possible problem in such a setup could be the influence of noise of real sensor data on the scheduling algorithms. Here, a more sophisticated and probabilistic activation spreading algorithm might be required. The reason is that the activation for the property nodes and thus also for the object nodes is more ambiguous for noisy measurements. Another interesting aspect for further investigations is the triggering of an object hypothesis using a fast feed-forward pathway for prominent features as proposed for neocortical structures in [20]. This would confine the initial set of hypotheses and speedup the identification. In such a regime, the proposed algorithm would act as a refinement process. Furthermore, we want to investigate the relation between our scheduling algorithm and decision and game theory problems, where a gain (the number of excluded hypotheses) is often weighted against a risk (an ambiguous measurement).

References

1. Yarbus, A.L.: Eye Movements and Vision. Plenum Press, New York (1967)
2. Pashler, H.: Familiarity and visual change detection. Perception and Psychophysics **44**(4) (1988) 369–378
3. Just, M.A., Carpenter, P.A.: Eye fixations and cognitive processes. Cognitive Psychology **8**(4) (1976) 441–480
4. Henderson, J.M., Weeks, P.A., Hollingworth, A.: The effects of semantic consistency on eye movements during complex scene viewing. Experimental Psychology: Human Perception and Performance **25**(1) (1999) 210–228
5. Ballard, D.H., Hayhoe, M.M., Pelz, J.B.: Memory representations in natural tasks. Cognitive Neuroscience (1995)
6. Triesch, J., Ballard, D.H., Hayhoe, M.M., Sullivan, B.T.: What you see is what you need. Journal of Vision **3**(1) (2003) 86–94
7. Intraub, H.: The representation of visual scenes. Trends in Cognitive Sciences **1**(6) (1997) 217–221
8. Tsotsos, J.K.: On the relative complexity of active vs. passive visual search. International Journal of Computer Vision **7**(2) (1992) 127–141
9. Bajcsy, R.: Active perception vs. passive perception. In: Proceedings of the IEEE Workshop on Computer Vision: Representation and Control. (1985) 55–62
10. Aloimonos, Y.: Introduction. In: Active Vision Revisited. Lawrence Erlbaum Associates, Hillsdale (1993) 1–18
11. Ballard, D.H.: Animate vision. Artificial Intelligence **48**(1) (1991) 57–86
12. Navalpakkam, V., Itti, L.: Modeling the influence of task on attention. Vision Research **45**(2) (2005) 205–231
13. Hayhoe, M.: Vision using routines: A functional account of vision. Visual Cognition (7) (2000) 43–64
14. Eggert, J., Rebhan, S., Körner, E.: First steps towards an intentional vision system. In: Proc. International Conference on Computer Vision Systems. (2007)
15. Navalpakkam, V., Itti, L.: Search goal tunes visual features optimally. Neuron **53**(4) (2007) 605–617
16. Ullman, S.: Visual routines. Cognition **18** (1984)
17. Röhrbein, F., Eggert, J., Körner, E.: A cortex-inspired neural-symbolic network for knowledge representation. In: Proceedings of the 3rd International Workshop on Neural-Symbolic Learning and Reasoning. (2007)
18. Rebhan, S., Einecke, N., Eggert, J.: Consistent modeling of functional dependencies along with world knowledge. In: Proceedings of World Academy of Science, Engineering and Technology: International Conference on Cognitive Information Systems Engineering. Volume 54. (2009) 341–348
19. Rebhan, S., Röhrbein, F., Eggert, J.: Attention modulation using short- and long-term knowledge. In: Proceedings of the 6th International Conference on Computer Vision Systems. (2008) 151–160
20. Körner, E., Gewaltig, M.O., Körner, U., Richter, A., Rodemann, T.: A model of computation in neocortical architecture. Neural Networks **12**(7–8) (1999) 989–1006