# Attention Modulation Using Short- and Long-Term Knowledge

## Sven Rebhan, Florian Röhrbein, Julian Eggert, Edgar Körner

### 2008

**Preprint:**

This is an accepted article published in 6th International Conference on Computer Vision Systems. The final authenticated version is available online at: https://doi.org/[DOI not available]

# Attention Modulation Using Short- and Long-Term Knowledge

Sven Rebhan, Florian Röhrbein, Julian Eggert, and Edgar Körner

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

**Abstract.** A fast and reliable visual search is crucial for representing visual scenes. The modulation of bottom-up attention plays an important role here. The knowledge about target features is often used to bias the bottom-up pathway. In this paper we propose a system which does not only make use of knowledge about the target features, but also uses already acquired knowledge about objects in the current scene to speed up the visual search. Main ingredients are a relational short term memory in combination with a semantic relational long term memory and an adjustable bottom-up saliency. The focus of this work is to investigate mechanisms to use the memory of the system efficiently. We show a proof-of-concept implementation working in a real-world environment and performing visual search tasks. It becomes clear that using the relational semantic memory in combination with spatial and feature modulation of the bottom-up path is beneficial for speeding up such search tasks.

**Keywords:** scene representation, attention, semantic memory.

## 1 Introduction

The representation of important objects in a visual scene is essential for intelligent systems in real-world scenarios. In this context, important objects are defined as things that are required to solve a given task. Building up such representations comprises three main steps: acquiring task-relevant objects, keeping their position up-to-date and gathering properties like color, identity and shape.

Many state-of-the-art models only solve the task of acquiring objects and make use of an adjustable bottom-up saliency computation to speed up visual search tasks. The features of the target object are normally provided in a top-down manner. Newer approaches also learn the background statistics of the current scene to further improve the signal to noise ratio (SNR) between the target object and the background [1,2]. All these models show that one is able to gain a huge speed up in search tasks by using this kind of modulation.

There are only few models trying to tackle the remaining aspects of a scene representation as mentioned above. In [3] an architecture is shown which provides a framework for representing objects in a scene. However, it only considers long

A. Gasteratos, M. Vincze, and J.K. Tsotsos (Eds.): ICVS 2008, LNCS 5008, pp. 151–160, 2008.

term knowledge, like the features of an object. The short term memory (STM), containing information of already seen objects in the current scene, is not used. This means that information gathered on these objects, especially if they do not match the target object, is completely lost. Furthermore the spatial arrangement of objects, which would be helpful especially for immobile objects is ignored.

We show that using short and long term knowledge is important for a more realistic scenario which includes varying tasks in a real environment without starting the system from scratch every time a new task arrives. Regarding the architecture and layout of our system we stick to our idea of an intentional vision system [4] as proposed previously. Such a system is top-down driven and organizes the underlying information acquisition processes dynamically. Furthermore we use a biologically inspired relational semantic memory [5]. Even though we do not make use of this biological background in this work directly it provides a huge potential for further extensions of our system. It is also important to mention that the STM has the same relational structure as the long-term memory (LTM) which makes it easy to transfer novel knowledge to the LTM later.

In the following section we describe the three use-cases we have identified during our work and introduce our overall system architecture. In Sect. 3 we describe the basic components of the system more in detail. After this we show some results on a real-world scene in Sect. 4. Finally we discuss the results and give a brief outlook on our future work.

## 2 System Architecture

The aim of our work is to create a system that is able to represent a dynamic visual scene with respect to a given task. Currently the set of possible tasks is reduced to search tasks for known objects. These objects and their corresponding appearance are stored in the LTM. The system we propose follows the principles described in [4] and can be seen in Fig. 1. The main components of the system are the relational semantic memory, a tunable saliency similar to [1] and a feature extraction module. The functionality of all components is described in Sect. 3. As can be seen in Fig. 1 there are two inputs to the system: the images of the stereo camera head (to get depth information) and the task, which is directly passed to the LTM and activates the target object. One can identify three different use-cases for the system:

1. no target object is specified
2. target object is specified and already stored in the short-term memory
3. target object is specified, but not yet stored in the short-term memory

In the first case we simply search for interesting objects (in this case colorful and close) with a default modulation of the saliency in the current scene. The positions found for these objects are inhibited and stored in the STM together with their measured appearance (color, size, etc).

In the second and third case a target object is specified, meaning that an object with its corresponding features has been activated in the LTM. Now we

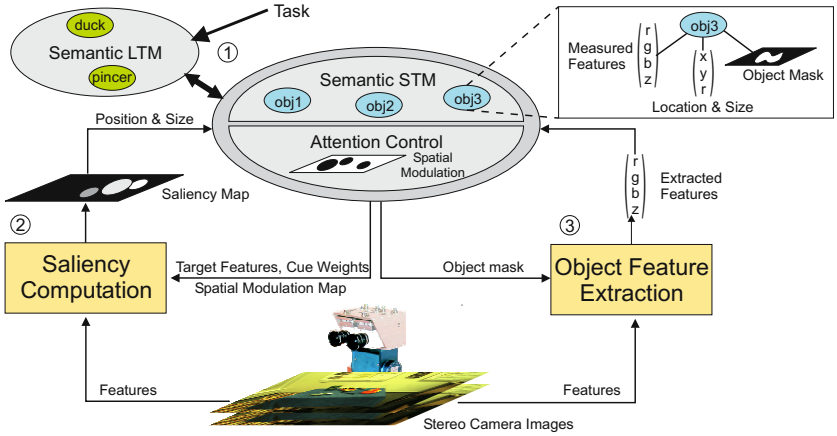


**Fig. 1.** The overall system architecture consists of three main ingredients: the semantic memory(1), the tunable saliency(2) and a feature extraction component(3)

have to decide if we have already seen the target object or not. Therefore we first search in the set of already seen objects stored in the STM for a matching object. To decide if an object matches the target, we compare the features given for the target object with the measured features of the already seen object. If a distance measure (e.g. Euclidian distance in feature space) is below a chosen static threshold, the object is considered to be the target object (case two). The previously measured features and the last known location of the matching object are used to modulate the bottom-up saliency both in the feature and spatial domain to regain the object in the current input image.

If the search in the STM was not successful (case three), the bottom-up saliency is biased by using the features of the target object retrieved from LTM. An eventually stored preferred location of the object can also be incorporated, which is especially helpful for immobile objects. The tuning of the low-level visual components towards the features of the target object is done in a similar fashion as proposed in [1] and [2].

In all cases the modulated saliency map computes a candidate map for a given bias. Additionally to the location, a rough size estimate is given by the saliency (a detailed description is given in Sect. 3) for the candidates. Now the most promising candidate is selected from the map using a WTA algorithm to create an "object blob" in the STM, containing the candidate's location and estimated size. Even if the saliency map is tuned towards the features of a target object, it might still be possible that the selected candidate is a false positive. To eliminate this possibility we extract the candidate's features at its location including a size related surrounding. This is done in the feature extraction module. The measured features are also attached to the "object blob" and the candidate is checked using an Euclidian distance measure. If the candidate is verified successfully, we assume it is the searched object. Otherwise we keep the rejected object in the STM and

inhibit its location in the input image. By doing so the next candidate is selected in the saliency map. The procedure is repeated until the target object is found.

The false positive candidates are kept in memory for later use. If the task for the system changes, the new target object might already reside in the STM and the search can be performed much faster, because we already know the position of the object. At this point two important questions arise: "How do we forget objects in the STM?" and "How do we learn new objects and transfer them to the LTM?". Forgetting is important because we only have a limited storage capacity in our STM. Our current strategy is to memorize the existence of the objects, but to forget its features over time. Learning of new objects is not subject of this paper and is therefore omitted.

## 3 Component Description

As described in the previous section, our system is comprised of three main modules: the relational semantic memory (STM and LTM), a tuneable saliency map and a feature extraction module. Having looked at the interplay between the modules in the last section, the modules themselves and their functionality are described in this section.

### 3.1 Relational Semantic Memory

To represent relational and concept knowledge we have developed a graphical model which combines ideas from classical semantic networks and processing principles of the cortical minicolumn (see [5]). A sketch of this model can be seen in Fig. 2. An important aspect of our approach is that we only use one uniform node type in the network, which is the representational entity of all concepts, both in STM and LTM. This node type represents "saliency blobs",

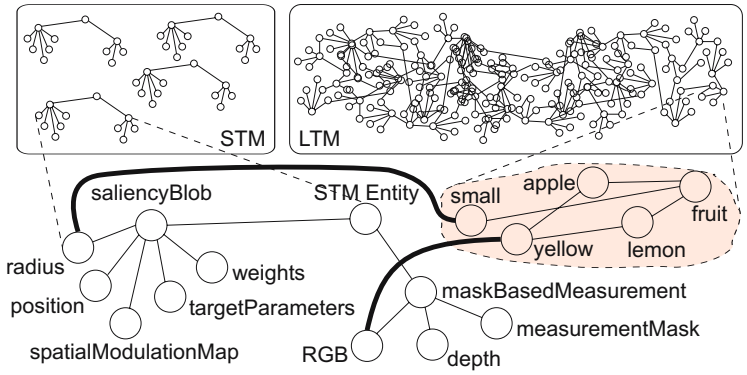


**Fig. 2.** The activated object in the LTM (here *lemon*) is instantiated in the STM, where all known properties of the object are inherited (e.g. *small*, *yellow*)

sensory measurements, properties, instances, categories and even relations between nodes. A relation in general is represented explicitly via nodes. For example *MadeOf(Table,Wood)* requires the five nodes *Table, Object, MadeOf, Material* and *Wood* (for more details see [6]). Very few basic relations like partonomic ones, for which we think biological evidence exist, are coded directly with specific link types, making our approach quite different from standard AI systems like [7].

For our current system we use a rather flat ontology consisting of a few objects and properties. The entities consist of a node "saliencyBlob" and a node "maskBasedMeasurement", which resembles the two information pathways in the system (saliency and feature extraction as shown in Fig. 1). To these nodes properties like position and color are attached, together with their modulatory influences (e.g. weights or masks). We provide an entity prototype, which is specialized into a new instantiation each time a task requires the generation of a new entity. This instantiation inherits all procedures and default values of the prototype which can be overwritten by objects activated in the LTM. In a next step, the attached values are passed down to the saliency computation and measurement process as modulatory inputs. All incoming measurements are stored in the corresponding nodes of the object.

### 3.2 Saliency Computation and Size Estimation

In our system the saliency map provides locations of possible object candidates and their estimated size. As shown in Fig. 3 the saliency computation can be separated into three steps: tuning of the cues (blue), center-surround contrast calculation (yellow) and computation of the lateral dynamics (green). First input features are modulated to enhance the contrast of the target object against the

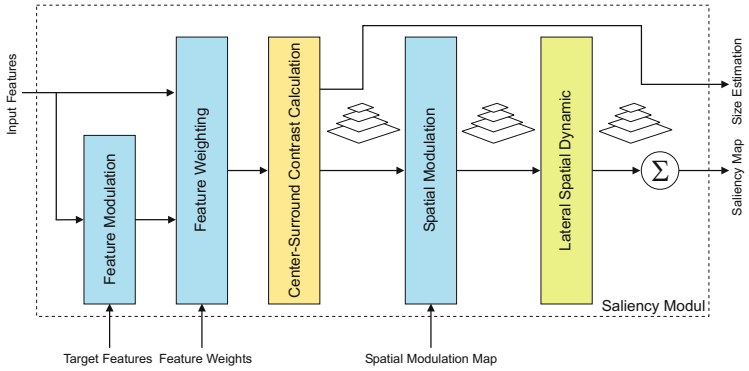


**Fig. 3.** The input features are modulated and weighted according to the modulatory inputs (bottom). After the contrast computation, spatial modulation and lateral dynamic which work on spatial pyramids, the resulting pyramid is integrated into a final saliency map. In addition center-surround contrast is also used to roughly estimate the object size (see Fig. 4).

background. This is done by calculating the similarity between the input vector $\boldsymbol{f}$ and the top-down provided Gaussian target distributions ($\boldsymbol{t}$,$\boldsymbol{\sigma}$).

$$\hat{f} = e^{-\frac{1}{2}\sum\limits_{n}\frac{(f_n - t_n)^2}{\sigma_n^2}} \quad . \tag{1}$$

Here $n$ is the $n$-th channel of the vectors. In the case of RGB-color this reads as

$$\hat{f}_{rgb} = e^{-\frac{1}{2}\left(\left(\frac{f_r - t_r}{\sigma_r}\right)^2 + \left(\frac{f_g - t_g}{\sigma_g}\right)^2 + \left(\frac{f_b - t_b}{\sigma_b}\right)^2\right)} \quad . \tag{2}$$

After tuning the features, the center-surround contrast over all features within the same spatial scale is computed. Therefore the Euclidian distance between the center and the surround part of the feature vector is calculated for each channel of $\boldsymbol{f}$ and different sizes

$$c_s = \| \left(F_s^{center} - F_s^{surround}\right) * \boldsymbol{f} \|^2 \ , \tag{3}$$

where $s$ is a certain size of the filter $F$. For choosing the size of the center and the surround the scheme proposed in [8] is used. Each filter is normalized to have a mean of zero. The contrast maps are now biased with the spatial modulation map, provided in a top-down manner.

To calculate the size estimation at a certain position the distribution over all filter scales is used as shown in Fig. 4. Here we exploit the fact that we use Gaussian filters with different sizes to calculate the contrast. If we assume that an object is circular and homogeneously structured, the response stays constant for increasing sizes of the filter until the filter exceeds the object size. At this point the structure changes because of the object boundaries and so does the filter response. Using the standard deviation of the corresponding filter plus a constant scaling factor we are able to generate an initial hypothesis of the object size.

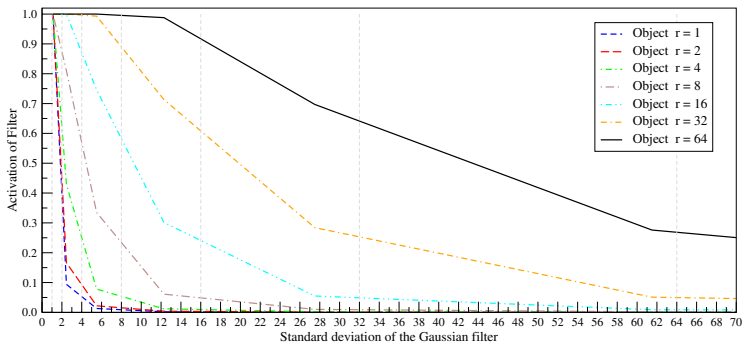


**Fig. 4.** The Gaussian filter response stays nearly constant until the filter exceeds the object size. Beyond this point the filter response decreases rapidly. This property can be used to estimate the object size.

After calculating the contrast on different scales we perform a lateral spatial competition within each scale with the aim to enhance the SNR. Contrary to the dynamic in [8], we use an Amari dynamic [9], which shows a hysteresis over time that is controllable by different parameters. We use this behavior to model a sensory memory for the saliency map. Finally the maps of different scales are combined to one single saliency map, which contains the location of the candidates for the target object.

### 3.3 Feature Extraction

The feature extraction module measures the features of an object in the current input using a given segmentation mask of the object. So far a segmentation process is not part of our system, so we use the location with an object-size dependent surrounding (see Sect 3.2) as the mask. The measurement itself is done by calculating the mean and standard deviation in each feature channel of the input image. The top-down provided object mask is used to exclude parts of the image which do not belong to the object.

## 4 Results

In this section we show how the system behaves in a real scene. In Fig. 5 one can see the first of the three cases (see Sect. 2). Without a given task the system sequentially acquires knowledge about its environment over time until all locations in the input image are inhibited. In each timestep the STM is filled with the one

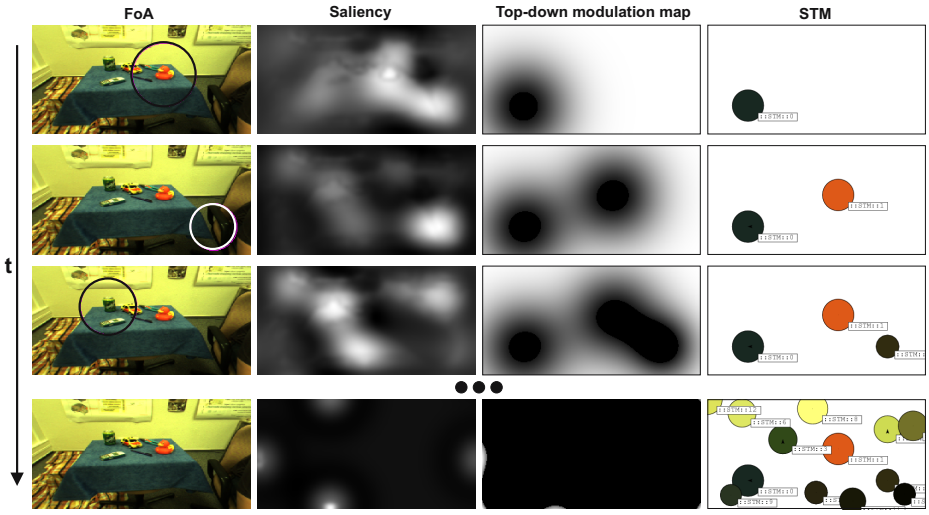


**Fig. 5.** The STM is filled over time, without a specific target. The internal state of the STM is shown on the right, representing all known objects with their measured color and size. The spatial modulation map is generated from all known objects in the STM.

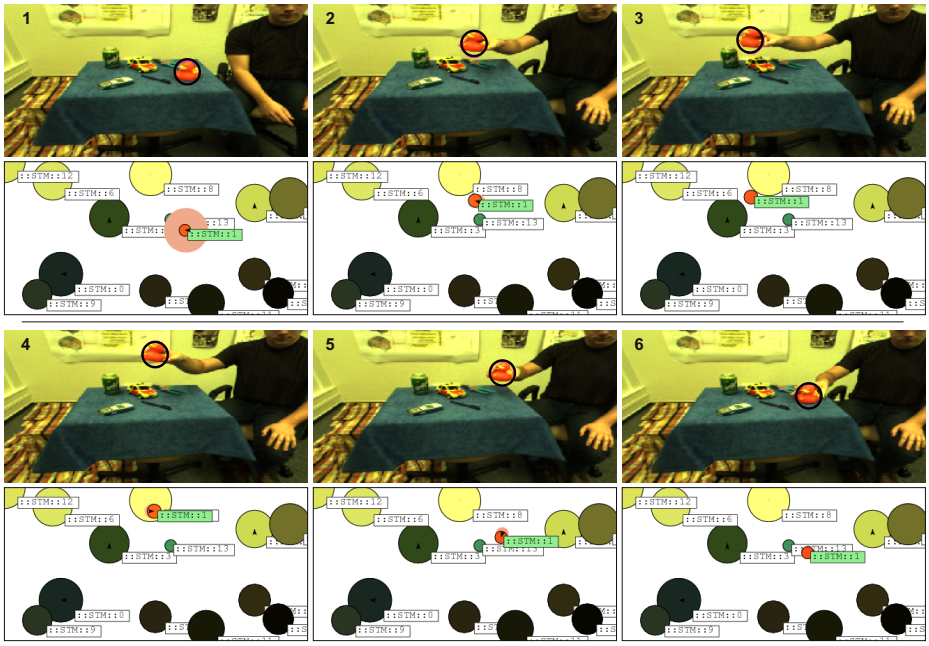


**Fig. 6.** The task is to find the red duck. The object was found in STM (marked with a green text label) and its features (color, depth and position) were used to bias the saliency. In this sequence one can see that even while moving the object the content of the STM (size, position, depth and color) is continuously updated.

object that currently has the focus of attention (FoA). The underlying saliency map is only modulated by the top-down spatial modulation map to inhibit the already known objects, the features are not modulated. The modulation map itself is generated using the position and estimated size of the objects in the STM. The rightmost column of Fig. 5 shows a dump of the STM at different times. Here the position, size and color of the "object blobs" represent the measured properties of the objects, the measured depth is attached but not shown.

In Fig. 6 the second case (see Sect. 2) is shown, meaning that a task is given to the system and the searched object is already in the STM. This case is not handled in state-of-the-art systems. In our example we specified the task to find the duck. A lookup in the LTM shows that our duck is red ($r = 1.0, g = 0.0, b = 0.0$). Now a search in the STM for a red object is performed, revealing that a red object is already known to the system (marked green in Fig. 6). By finding the target object in the STM all information about the object are instantaneously accessible without referring to the bottom-up saliency or feature extraction. However, to keep track of the changes in features and position we now bias the saliency, both in feature and spatial domain, to concentrate on the object found and verify the properties at the known position. The feature bias is performed

using the color of the object found in the STM, similarly the position and size is used to generate an excitatory spatial bias at the last known position. As can be seen in the first image of Fig. 6, the biasing of the saliency also leads to a better size estimation. After verifying that the object is still there and that the measured properties match the target criteria, we concentrate the system's attention on the object found and keep a reference to this object in the STM. To concentrate the attention we keep the current biasing of the system (spatial and feature). By keeping the attention on the object we are able to continuously update its position, depth, size and color. After each update of a property, the modulation for the corresponding property is updated. If for example the size decreases, the spatial bias becomes more narrow (as in the first image of Fig. 6). The same is done for the position of the object. By continuously updating the object's position in the STM, a tracking of simple objects is possible as shown in the sequence of Fig. 6.

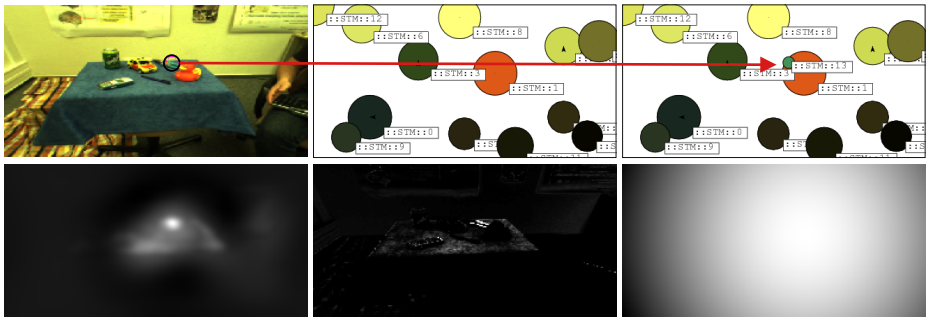

**Fig. 7.** The task is to find the cyan pincer. Therefore we bias the features to highlight cyan objects (second row middle). Additionally we provide the system with a spatial hint (second row right), because pure color information is not enough to exceed the WTA threshold. The final saliency can be seen on the left hand side of the second row. The STM content (first row) before and after finding the object shows that the pincer has been added.

In Fig. 7 the third case (see Sect. 2) is shown, meaning that a task is given to the system and the searched object is not yet in the STM. In this example the task is to find the pincer in the scene. Given this task, a search in the LTM for features of the pincer is performed, revealing that it has the color cyan ($r = 0.0, g = 1.0, b = 1.0$). Because a search in the STM for a cyan object was not successful we bias the features of the saliency map to prefer cyan objects. The result of this biasing can be seen in the middle of the second row of Fig. 7. The biasing of the feature channels is not sufficient to exceed the threshold of the WTA selection, therefore we additionally give a spatial hint to the system, indicating that the searched object is on the right hand side of the current scene. Using this hint the system is able to find the object and adds it to the STM.

## 5 Discussion

In the system we propose, the short-term memory plays a crucial role. We showed that by using the knowledge about already seen objects, we are able to instantaneously access the information about a given target object if it is already stored in the short-term memory. Otherwise, we used knowledge about the object's features and potential locations, coming from the long-term memory, to find "difficult" objects like the pincer as well. We furthermore showed that by deliberately directing attention to a certain object and continuously updating its properties in memory, a simple tracking mechanism can be established.

To cover highly dynamic scenes which are not covered yet, our future systems will use a more elaborated tracking system, together with a more powerful modulation-mechanism for the saliency. Good examples for such mechanisms were shown in [1] and [2]. However, a mandatory step to go is the learning of new objects and its properties to fill the long-term memory. For this, additional components such as a classifier and a segmentation mechanism will be required.

## References

1. Navalpakkam, V., Itti, L.: Search goal tunes visual features optimally. Neuron 53(4), 605–617 (2007)
2. Frintrop, S., Backer, G., Rome, E.: Goal-directed search with a top-down modulated computational attention system. In: Kropatsch, W.G., Sablatnig, R., Hanbury, A. (eds.) DAGM 2005. LNCS, vol. 3663, pp. 117–124. Springer, Heidelberg (2005)
3. Navalpakkam, V., Arbib, M.A., Itti, L.: Attention and scene understanding. In: Neurobiology of Attention, pp. 197–203 (2005)
4. Eggert, J., Rebhan, S., Körner, E.: First steps towards an intentional vision system. In: Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007) (2007)
5. Roehrbein, F., Eggert, J., Körner, E.: A cortex-inspired neural-symbolic network for knowledge representation. In: Proceedings of the IJCAI Workshop on Neural-Symbolic Learning and Reasoning (accepted, 2007)
6. Roehrbein, F., Eggert, J., Körner, E.: Prototypical relations for cortex-inspired semantic representations. In: Proceedings of the 8th International Conference on Cognitive Modeling, pp. 307–312 (2007)
7. Brachman, R., Levesque, H.: Knowledge Representation and Reasoning. Morgan Kaufmann Publishers Inc. San Francisco (2004)
8. Itti, L., Koch, C.: A saliency-based search mechanism for overt and covert shifts of visual attention. Vision Research 40(10–12), 1489–1506 (2000)
9. Amari, S.: Dynamics of pattern formation in lateral-inhibition type neural fields. Biological Cybernetics 27, 77–87 (1977)