Associating words to visually recognized objects *

Andreas Knoblauch, Rebecca Fay, Ulrich Kaufmann, Heiner Markert, Günther Palm
Abteilung Neuroinformatik, Fakultät für Informatik, Universität Ulm,
Oberer Eselsberg, D-89069 Ulm, Germany
Tel: (+49)-731-50-24151; Fax: (+49)-731-50-24156
{knoblauch,fay,kaufmann,markert,palm}@neuro.informatik.uni-ulm.de

Abstract

Using associative memories and sparse distributed representations we have developed a system that can learn to associate words with objects, properties like colors, and actions. This system is used in a robotics context to enable a robot to respond to spoken commands like "bot show plum" or "bot put apple to yellow cup". The scenario for this is a robot close to one or two tables on which there are certain kinds of fruit and/or other simple objects. We can demonstrate part of this scenario where the task is to find certain fruits in a complex visual scene according to spoken or typed commands. This involves parsing and understanding of simple sentences and relating the nouns to concrete objects sensed by the camera and recognized by a neural network from the visual input.

Introduction

When words referring to actions or visual scenes are presented to humans, distributed networks including areas of the motor and visual systems of the cortex become active (e.g., Pulvermüller, 1999). The brain correlates of words and their referent actions and objects appear to be strongly coupled neuron ensembles in defined cortical areas. The theory of cell assemblies (Hebb, 1949; Braitenberg, 1978; Palm, 1982, 1990, 1993) provides one of the most promising frameworks for modeling and understanding the brain in terms of distributed neuronal activity. It is suggested that entities of the outside world (and also internal states) are coded in groups of neurons rather than in single ("grandmother") cells, and that a neuronal cell assembly is generated by Hebbian coincidence or correlation learning where the synaptic connections are strengthened between co-activated neurons. Models of neural (auto-) associative memory have been developed as abstract models for cell assemblies.

One of our long-term goals is to build a multimodal internal representation using cortical neuron maps, which will serve as a basis for the emergence of action semantics using mirror neurons (Rizzolatti *et al.*, 1999). We have developed a model of several visual, language, planning, and motor areas to enable a robot to understand and react to spoken commands in basic scenarios of the project. The essential idea is that different cortical areas represent different aspects (and correspondingly different notions of similarity) of the same entity (e.g., visual, auditory language, semantical, syntactical, grasping related aspects of an apple) and that the (mostly bidirectional) long-range cortico-cortical projections represent hetero-associative memories that translate between these aspects or representations. This involves anchoring symbols such as words in sensory and motor representations where invariant association processes are required, for example recognizing a visually perceived object independent of its position, color, or view direction. Since word symbols usually occur in the context of other words specifying its precise meaning in terms of action, goals, and sensory information, anchoring words additionally requires language understanding.

In this work we present a neurobiologically-motivated model of language processing and visual object recognition based on cell assemblies (Hebb, 1949; Braitenberg, 1978; Palm, 1982, 1990). We have developed a system that can learn to associate words with objects, properties like colors, and actions. This system is used in a robotics context to enable a robot to respond to spoken commands like "bot show plum" or "bot put apple to yellow cup". The scenario for this is a robot close to one or two tables on which there are certain kinds of fruit and/or other simple objects. We can demonstrate part of this scenario where the task is to find certain fruits in a complex visual scene according to spoken or typed commands. This involves parsing and understanding of simple sentences and relating the nouns to concrete objects sensed by the camera and recognized by a neural network from the visual input.

In the first section we outline the concept of cell assemblies as a model for sequential associative processing in cortical areas. Then we briefly describe our robot architecture used for implementing simple scenarios of associating words to objects, and detail the visual object recognition and the language module. Finally, we summarize and discuss our results.

*This work is supported by the MirrorBot project of the European Union. Further demonstrations of this work will be made available electronically at `http://www.his.sunderland.ac.uk/mirrorbot`

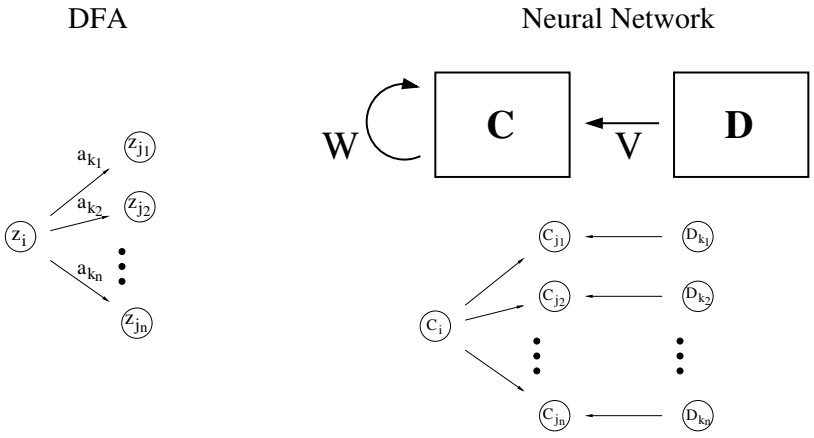

Figure 1: Comparison of a deterministic finite automate (DFA, left side) with a neural network (right side) implementing formal language. Each δ transition $\delta(z_i, a_k) = z_j$ corresponds to synaptic connections from neuron C_i to C_j and from input neuron D_k to C_j (see text for details).

Language and cell assemblies

A large part of our model is based on associative memory and cell assemblies. Anchoring a symbol first requires understanding the context in which the symbol occurs. Thus, one requirement for our system is language processing and understanding.

Regular grammars, finite automates, and neural assemblies

Noam Chomsky developed a hierarchy for grammar types (Hopcroft & Ullman, 1969; Chomsky, 1957). For example, a grammar is called *regular* if the grammar can be expressed by rules of the type

$$\begin{aligned} A &\rightarrow a \\ B &\rightarrow bC \end{aligned}$$

where lower case letters are *terminal symbols* (i.e. elements of an alphabet Σ), and upper case letters are *variables*. Usually there is a starting variable S which can be expanded by applying the rules. A sentence $s \in \Sigma^*$ (which is a string of alphabet symbols of arbitrary length) is called *valid with respect to the grammar* if s can be derived from S by applying grammatical rules and resolving all variables by terminal symbols.

There are further grammar types in the Chomsky hierarchy which correspond to more complex rules, e.g. context-free and context-sensitive grammars, but here we will focus on regular grammars. It is easy to show that regular grammars are equivalent to deterministic finite automata (DFA). A DFA can be specified by $M = (Z, \Sigma, \delta, z_0, E)$ where $Z = \{z_0, z_1, ..., z_n\}$ is the set of states, Σ is the alphabet, $z_0 \in Z$ is the starting state, $E \subseteq Z$ contains the terminal states, and the function $\delta : (Z, \Sigma) \rightarrow Z$ defines the (deterministic) state transitions. A sentence $s = s_1 s_2 ... s_n \in \Sigma^*$ is valid with respect to the grammar if iterated application of δ on z_0 and the letters of s transfers the automaton's starting state to one of the terminal states, i.e., if $\delta(...\delta(\delta(z_0, s_1), s_2), ..., s_n) \in E$ (cf. left side of Fig. 1).

In the following we show that DFAs are equivalent to binary recurrent neural networks such as the model architecture described below (see Fig. 2). As an example, we first specify a simpler model of recurrent binary neurons by $N = (C, I, W, V, c_0)$, where $C = \{C_0, C_1, ..., C_n\}$ contains the local cells of the network, $D = \{D_1, D_2, ..., D_m\}$ is the set of external input cells, $W = (w_{ij})^{n \times n}$ is a binary matrix where $w_{ij} \in \{0, 1\}$ specifies the strength of the local synaptic connection from neuron C_i to C_j, and, similarly, $V = (v_{ij})^{m \times n}$ specifies the synaptic connections from input cell D_i to cell C_j. The temporal evolution of the network can be described by

$$c_i(t+1) \quad = \quad \begin{cases} 1, \text{if } \sum_j w_{ji} c_j(t) + \sum_j v_{ji} d_j(t) \geq \Theta_i \\ 0, \text{otherwise.} \end{cases}$$

where $c_i(t)$ is the output state of neuron C_i at time t, and Θ_i is the threshold of cell C_i. Figure 1 illustrates the architecture of this simple network.

The network architecture can easily be adapted to simulate a DFA. We identify the alphabet Σ with the input neurons, and the states Z with the local cells, i.e. each $a_i \in \Sigma$ corresponds to input cell D_i, and, similarly, each $z_i \in Z$ corresponds to a local cell C_i. Then we can specify the connectivity as follows: Synapses w_{ij} and v_{kj} are active if and only if $\delta(z_i, a_k) = z_j$ for the transition function δ of the DFA (see Figure 1). In order to decide if a sentence $s = a_{i(0)} a_{i(1)} a_{i(2)} ...$ is valid with respect to the language we can specify the activation of the input units by $d_i(t) = 1$ and $d_j = 0$ for $j \neq i(t)$. By choosing threshold $\Theta_i = 2$ for choosing a starting activation where only cell c_0 is active, the network obviously simulates the DFA. That means, after processing of the last sentence symbol, one of the neurons corresponding to the end states of the DFA will be active if and only if s is valid.

The described neural network architecture for recognizing formal languages is quite simple and reflects perfectly the structure of a DFA even on the level of single neurons. However, such a network is biologically not very realistic

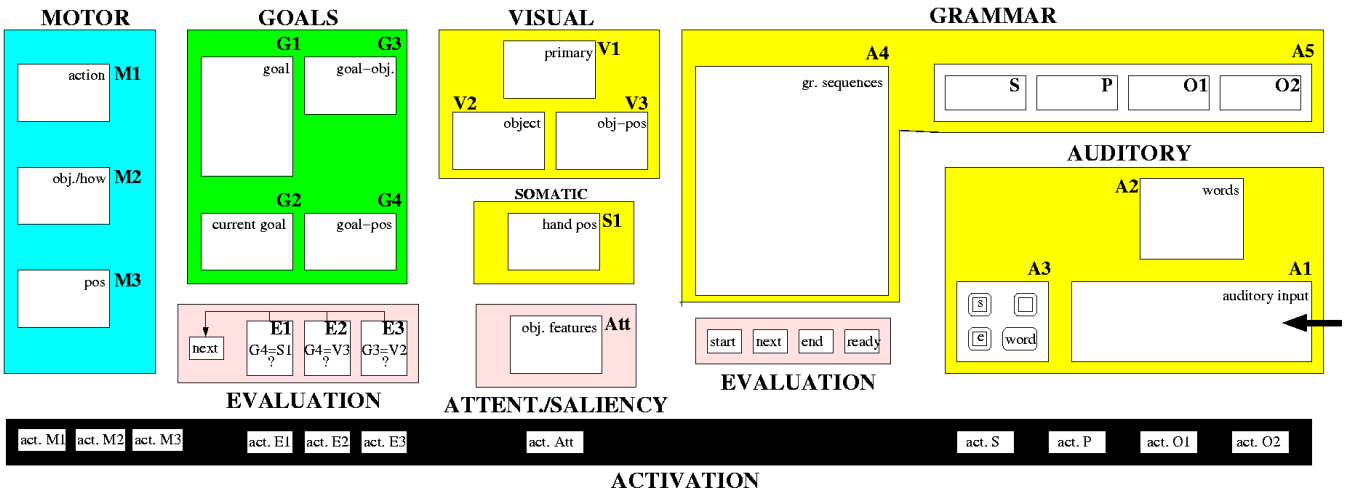

Figure 2: Cortical architecture involving several inter-connected cortical areas corresponding to auditory, grammar, visual, goal, and motor processing. Additionally the model comprises evaluation fields and activation fields (see text).

since, for example, such an architecture is not robust against partial destruction and it is not clear how such a delicate architecture could be learned. The model becomes more realistic if we interpret the nodes in Fig. 1 not as *single* neurons but as groups of nearby neurons which are strongly interconnected, i.e., local cell assemblies. This architecture has two additional advantages: First, it enables *fault tolerance* since incomplete input can be completed to the whole assembly. Second, overlaps between different assemblies can be used to express similarity, hierarchical, and other relations between represented entities. In the following subsection we describe briefly a model of associative memory which allows us to implement the assembly network analogously to the network of single neurons in Fig. 1.

Cell assemblies and neural associative memory

We decided to use *Willshaw associative memory* as a single framework for the implementation of cell assemblies in cortical areas (Willshaw, Buneman, & Longuet-Higgins, 1969; Palm, 1980, 1982, 1991; Schwenker, Sommer, & Palm, 1996; Sommer & Palm, 1999). A *cortical area* consists of n binary neurons which are connected with each other by binary synapses. A *cell assembly* or *pattern* is a binary vector of length n where k one-entries in the vector correspond to the neurons belonging to the assembly. Usually k is much smaller than n. Assemblies are represented in the synaptic connectivity such that any two neurons of an assembly are bidirectionally connected. Thus, an assembly consisting of k neurons can be interpreted as a k-clique in the graph corresponding to the binary matrix A of synaptic connections. This model class has several advantages over alternative models of associative memory such as the most popular Hopfield model (Hopfield, 1982). For example, it better reflects the cortical reality where it is well known that activation is sparse (most neurons are silent most of the time), and that any neuron can have only one type of synaptic connection (either excitatory or inhibitory).

Instead of classical one-step retrieval we used an improved architecture based on spiking associative memory (Knoblauch & Palm, 2001; Knoblauch, 2003). A cortical area is modeled as a local population of n neurons which receive input from other areas via Hebbian learned hetero-associative connections. In each time step this external input initiates pattern retrieval. The neurons receiving the strongest external input will fire first, and all emitted spikes are fed back immediately through the Hebbian learned auto-associative connections resulting in activation of single assemblies. In comparison to the classical model, this model has a number of additional advantages. For example, assemblies of different size k can be stored, and input superpositions of several assemblies can more easily be separated.

In the following section we present the architecture of our cortical model which enables a robot to associate words to visually recognized objects, and thereby anchoring symbolic word information in sensory data. This model consists of a large number of interconnected cortical areas, each of them implemented by the described spike counter architecture.

Cell-assembly based model of cortical areas

We have designed a cortical model consisting of visual, tactile, auditory, language, goal, and motor areas, and implemented parts of the model on a robot. Each cortical area is based on the spike counter architecture described in the previous section. The model is simulated synchronously in discrete time steps. That means, in each time step t each area computes its output vector $y(t)$ as a function of the output vectors of connected areas at time $t-1$. In addition to the auto-associative internal connection within each area there are also hetero-associative connections between theses areas (see Fig. 4).

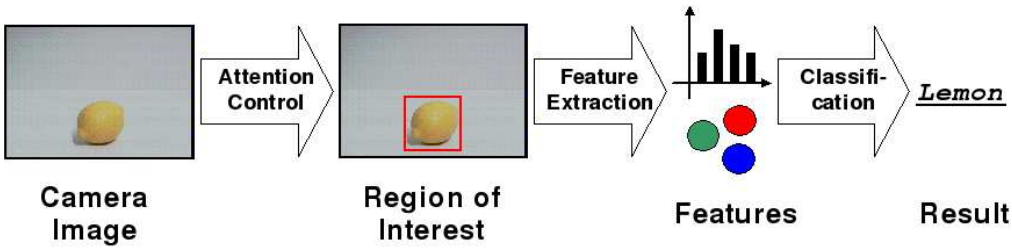

Figure 3: The visual object recognition system consists of three components: attention control, feature extraction and classification. The interconnection of the different components is depicted as well as the inputs and outputs of the miscellaneous components. Starting with the camera image the flow of the classification process is shown.

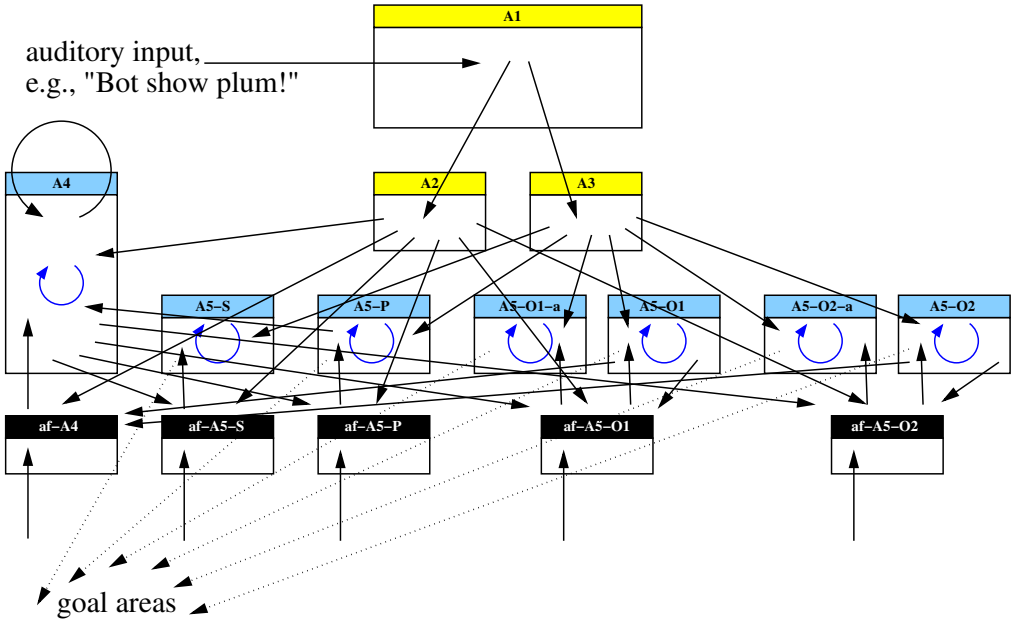

Figure 4: The language system consisting of 10 cortical areas (large boxes) and 5 thalamic activation fields (small black boxes). Black arrows correspond to inter-areal connections, gray arrows within areas correspond to short-term memory.

Overall architecture

Figure 2 illustrates the overall architecture of our cortical model. The model consists of auditory areas to represent spoken or typed language, of grammar areas to interpret spoken or typed sentences, visual areas to process visual input, goal areas to represent action schemes, and motor areas to represent motor output. Additionally, we have auxiliary areas or fields to activate and deactivate the cortical areas (activation fields), to compare corresponding representations in different areas (evaluation fields), and to implement attention. Each small white box corresponds to an associative memory as described in the previous section. The visual and auditory areas comprise additional neural networks for processing of camera images and acoustic input. Currently, we have implemented parts of the model on a robot. In the following sections we describe visual object recognition and language processing in more detail.

Visual object recognition

Figure 3 gives an overview of the object recognition system which is currently used to classify fruits and hand gestures (see Fay *et al.*, 2004). The object recognition system consists of three components: (1) The *visual attention control system* localizes the objects of interest based on an attention control algorithm using top-down information from higher cortical areas. (2) The *feature extraction system* analyzes a clip of the camera image corresponding to the region of interest. Scale and translation invariance is achieved by rescaling the clipped window and using inherently invariant features as input for the classification system. The extracted features comprise local orientation and color information. (3) The *classification system* uses the extracted features as input to a hierarchical neural network which is described in the following in more detail:

The basic idea of using hierarchical neural networks is the division of a complex classification task into several less

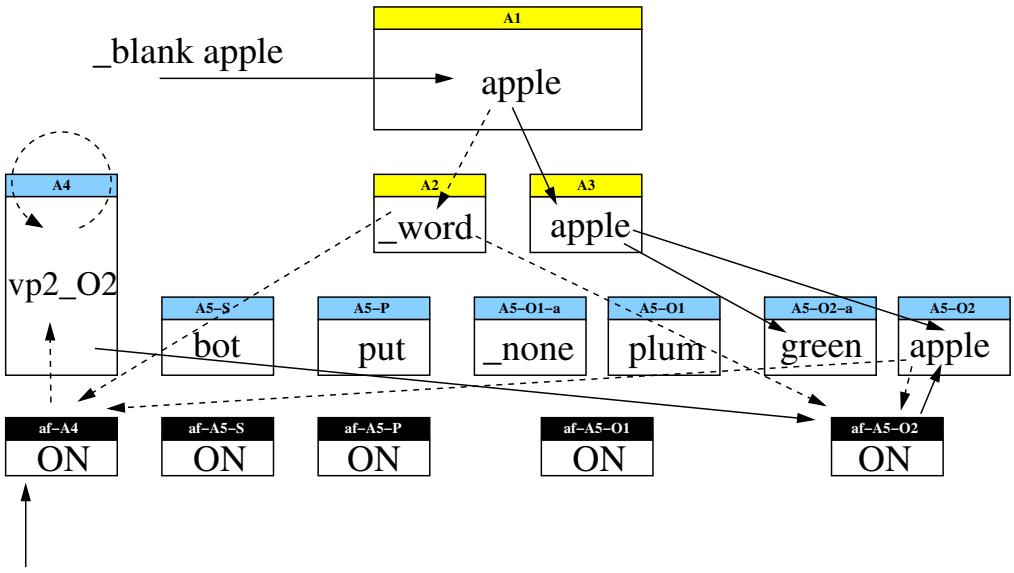

Figure 5: System state of the language model after 30 simulation steps when processing the sentence "Bot put plum to green apple". (Processing of a word requires about 5-6 steps on average; during each simulation step the state of the associative network is synchronously updated).

complex classification tasks by making coarse discrimination at higher levels of the hierarchy and refining the discrimination with decreasing depth of the hierarchy. A hierarchical neural network consists of several simple neural networks that are arranged as a rooted directed acyclic graph or a tree. Each node within the hierarchy represents a neural network. A set of classes is assigned to each node where the set of classes of one node is always a subset of the set of classes of its predecessor node. Thus each node only has to discriminate between a small number of subsets of classes but not between various classes.

In our approach, the hierarchy is generated by unsupervised k-means clustering (Hertz, Krogh, & Palmer, 1991). The hierarchy emerges from the successive partition of class sets into disjoint subsets. Beginning with the root node k-means clustering is performed using data points of all classes assigned to the current node. The partitioning of the classes into subclasses is done by determining for each class to which k-means prototype the majority of data points belonging to this class is assigned when presenting them to the trained k-means network. Each prototype represents a successor node. This procedure is recursively applied until no further partitioning is possible. Then end nodes are generated that do not discriminate between subsets of classes any longer but between single classes. As on each level there is always a division into disjoint subsets of the classes the resulting hierarchy is a tree. Once the hierarchy is established, RBF (radial basis function) networks are used as classifiers. They are trained with a three phase learning algorithm (Schwenker, Kestler, & Palm, 2001).

For anchoring the feature-based sensory data in symbolic word representations we remain to design a binary code for each entity in order to express the hierarchy into the domain of cell assemblies. This code should preserve similarity of the entities as expressed by the hierarchy. A straight-forward approach is to use binary vectors of length corresponding to the total number of neurons in all RBF networks. Then in a representation of a camera image those components are activated that correspond to the l strongest activated RBF cells on each level of the hierarchy. This results in sparse and translation invariant visual representations of objects.

Language processing

Figure 4 shows 15 areas of our model for cortical language processing. Each of the areas is modeled as a spiking associative memory of 100 neurons. Similar as described for visual object recognition, we defined for each area a priori a set of binary patterns constituting the neural assemblies stored auto-associatively in the local synaptic connections. The model can roughly be divided into three parts. (1) Primary cortical auditory areas A1,A2, and A3: First, auditory input is represented in area A1 by primary linguistic features (such as phonemes), and subsequently classified with respect to function (area A2) and content (area A3). (2) Grammatical areas A4, A5-S, A5-O1-a, A5-O1, A5-O2-a, and A5-O2: Area A4 contains information about previously learned sentence structures, for example that a sentence starts with the subject followed by a predicate and corresponds roughly to the DFA network illustrated in Fig. 1. In addition to the auto-associative connections, area A4 has also a *delayed* feedback-connection where the state transitions are stored hetero-associatively corresponding to matrix W in Fig. 1. The other grammar areas contain representations of the different sentence constituents such as subject (A5-S), predicate (A5-P), or object (A5-O1,O1-a,O2,O2-a). (4) Activation fields af-A4, af-A5-S, af-A5-O1, and af-A5-

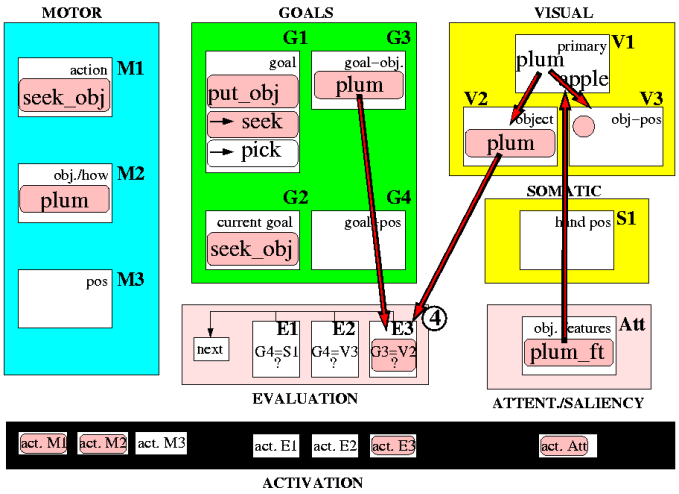

Figure 6: System state of the goal/motor module after 24 further simulation steps when performing the command “put plum (to) green apple!”. The robot is about to finish the subgoal of seeking the plum.

O2: The activation fields are relatively primitive areas that are connected to the corresponding grammar areas. They serve to activate or deactivate the grammar areas in a rather unspecific way. Although establishing a concrete relation to real cortical language areas of the brain is beyond the scope of this work (e.g., Knoblauch & Palm, 2003; Pulvermüller, 2003), we suggest that areas A1,A2,A3 can roughly be interpreted as parts of Wernicke’s area, and area A4 as a part of Broca’s area. The complex of the grammatical role areas A5 might be interpreted as parts of Broca’s or Wernicke’s area, and the activation fields as thalamic nuclei.

Figure 5 shows the result of processing the sentence “bot put plum to green apple” which means that the robot should put the plum to the location of the green apple. The sentence has been segmented into subject (A5-S), predicate (A5-P), and the two objects (A5-O1/O2), and this information is passed on to the goal areas where appropriate actions are planned, such as first seeking and moving to the plum, then picking the plum, seeking the apple, and moving to the apple, and then dropping the plum.

Integration of visual and language representations

Figure 6 illustrates the state of the cortical motor and goal areas when performing the command associated with the perceived sentence “Bot put plum (to) green apple” (cf. Fig. 5). The language representation has been interpreted as command and routed to the goal areas, where in area G1 a goal sequence assembly is activated (with a similar organization as grammatical area A4). In particular, object information has been routed to area G3 where connections to the attention system initiates searching for the plum. This means that the attention control system that searches for regions of interest uses easily recognizable plum-features, e.g., blue blobs. The search goes on until the classification system has recognized a plum in the current region of interest. This will lead the visual system to extract information about the plum and its position from area V1 to areas V2 and V3. Next E3 records the result of an associative matching between V2 and G3. Thereby it “realizes” that the “visual plum” is indeed the desired object (“the symbolic plum”). After recognizing that V2 contains the desired object, the sequence assembly in G1 will switch to the next subgoal, from ’seek’ to ’pick’.

Discussion

We have presented a cell assembly based model for visual object recognition and cortical language processing that can be used for associating words with objects, properties like colors, and actions. This system is used in a robotics context to enable a robot to respond to spoken commands like ”bot put plum to green apple”. The model shows how sensory data from different modalities (e.g., vision and speech) can be integrated to allow performance of adequate actions. This also illustrates how symbol grounding could be implemented in the brain involving association of symbolic representations to invariant object representations (see Fig. 6).

Although we have currently stored only a limited number of objects and sentence types, it is well known for our model of associative memory that the number of storable items scales with $(n/\log n)^2$ for n neurons (Willshaw, Buneman, & Longuet-Higgins, 1969; Palm, 1980). However, this is true only if the representations are sparse and distributed which is a design principle of our model. As any finite system, our language model can implement only regular languages, whereas human languages seem to involve context-sensitive grammars. On the other hand, also humans cannot “recognize” formally correct sentences beyond a certain level of complexity suggesting that in practical speech we use language rather “regularly”.

References

Braitenberg, V. 1978. Cell assemblies in the cerebral cortex. In Heim, R., and Palm, G., eds., *Lecture notes in biomathematics (21). Theoretical approaches to complex systems.* Berlin Heidelberg New York: Springer-Verlag. 171–188.

Chomsky, N. 1957. *Syntactic structures.* Mouton, The Hague.

Fay, R.; Kaufmann, U.; Schwenker, F.; and Palm, G. 2004. Learning object recognition in an neurobotic system. *submitted to 3rd workshop SOAVE2004 - SelfOrganization of AdaptiVE behavior, Illmenau, Germany*.

Hebb, D. 1949. *The organization of behavior. A neuropsychological theory.* New York: Wiley.

Hertz, J.; Krogh, A.; and Palmer, R. 1991. *Introduction to the theory of neural computation.* Redwood City: Addison-Wesley.

Hopcroft, J., and Ullman, J. 1969. *Formal languages and their relation to automata.* Addison-Wesley.

Hopfield, J. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Science, USA* 79:2554–2558.

Knoblauch, A., and Palm, G. 2001. Pattern separation and synchronization in spiking associative memories and visual areas. *Neural Networks* 14:763–780.

Knoblauch, A., and Palm, G. 2003. Cortical assemblies of language areas: Development of cell assembly model for Broca/Wernicke areas. Technical report, Department of Neural Information Processing, University of Ulm. Report 5 of the MirrorBot project of the European Union.

Knoblauch, A. 2003. Synchronization and pattern separation in spiking associative memory and visual cortical areas. *PhD thesis, Department of Neural Information Processing, University of Ulm, Germany*.

Palm, G. 1980. On associative memories. *Biological Cybernetics* 36:19–31.

Palm, G. 1982. *Neural Assemblies. An Alternative Approach to Artificial Intelligence.* Berlin: Springer.

Palm, G. 1990. Cell assemblies as a guideline for brain research. *Concepts in Neuroscience* 1:133–148.

Palm, G. 1991. Memory capacities of local rules for synaptic modification. A comparative review. *Concepts in Neuroscience* 2:97–128.

Palm, G. 1993. On the internal structure of cell assemblies. In Aertsen, A., ed., *Brain Theory*. Amsterdam: Elsevier.

Pulvermüller, F. 1999. Words in the brain's language. *Behavioral and Brain Sciences* 22:253–336.

Pulvermüller, F. 2003. *The neuroscience of language: on brain circuits of words and serial order.* Cambridge, UK: Cambridge University Press.

Rizzolatti, G.; Fadiga, L.; Fogassi, L.; and Gallese, V. 1999. Resonance behaviors and mirror neurons. *Archives Italiennes de Biologie* 137:85–100.

Schwenker, F.; Kestler, H.; and Palm, G. 2001. Three learning phases for radial-basis-function networks. *Neural Networks* 14:439–458.

Schwenker, F.; Sommer, F.; and Palm, G. 1996. Iterative retrieval of sparsely coded associative memory patterns. *Neural Networks* 9:445–455.

Sommer, F., and Palm, G. 1999. Improved bidirectional retrieval of sparse patterns stored by hebbian learning. *Neural Networks* 12:281–297.

Willshaw, D.; Buneman, O.; and Longuet-Higgins, H. 1969. Non-holographic associative memory. *Nature* 222:960–962.