Integrating object recognition, visual attention, language and action processing on a robot in a neurobiologically plausible associative architecture

Rebecca Fay, Ulrich Kaufmann, Andreas Knoblauch, Heiner Markert, and Günther Palm

University of Ulm
Department of Neural Information Processing
D-89069 Ulm, Germany

Abstract. We have implemented a neurobiologically plausible system on a robot that integrates object recognition, visual attention, language and action processing using a coherent cortex-like architecture based on neural associative memories. This system enables the 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 carrying certain kinds of fruit and/or other simple objects. Tasks such as finding and pointing to certain fruits in a complex visual scene according to spoken or typed commands can be demonstrated. This involves parsing and understanding of simple sentences, relating the nouns to concrete objects sensed by the camera, and coordinating motor output with planning and sensory processing.

1 Introduction

We have implemented a neurobiologically plausible system on a robot that integrates object recognition, visual attention, language and action processing using a coherent cortex-like architecture based on neural associative memories (cf. [1]). This system enables the 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 carrying certain kinds of fruit and/or other simple objects. Tasks such as finding and pointing to certain fruits in a complex visual scene according to spoken or typed commands can be demonstrated. This involves parsing and understanding of simple sentences, relating the nouns to concrete objects sensed by the camera, and coordinating motor output with planning and sensory processing.

The underlying cortical architecture is motivated by the idea of distributed cell assemblies in the brain [2][3]. For visual preprocessing we use hierarchically organized radial-basis-function networks to classify objects selected by attention, where hidden states in this hierarchical network are used to generate sparse distributed cortical representations. Similarly, auditory input pre-processed by standard Hidden-Markov-Model architectures can be transformed into a sparse
binary code for cortical word representations. In further cortical areas for language and action the sensory input is syntactically and semantically interpreted and finally translated into motor programs. The essential idea behind the cortical architecture 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, grasp-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. These different notions of similarity can synergistically be used, for example, to resolve ambiguities within or across sensory modalities.

2 Architecture

Our architecture can roughly be divided into two large parts: (1) Sensory preprocessing and (2) cortical model. For preprocessing of auditory and visual objects we used artificial algorithms such as radial-basis-function networks for object recognition or Hidden-Markov-Models for speech preprocessing, while for the cortex model we used a biologically more plausible architecture of many interconnected associative networks. Figure 1 shows the different components of our model, their interactions and the division into two parts.

![Fig. 1. The architecture is roughly divided into two parts: Sensory preprocessing and cortical model.](image)

For the implementation of cortical cell assemblies [4][5][6][7][8][9], we decided to use the Willshaw model of associative memory as a single architectural framework. 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 [10]. 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).

Fig. 2. Interaction of the different areas of the cortical model ($v$: visual, $l$: location, $f$: contour features, $o$: visual objects, $h$: haptic / proprioceptive, $p$: phonetics, $s$: syntactic, $a$: action / premotoric, $g$: goals / planning) and their rough localization in the human brain.

Instead of classical one-step retrieval we used an extended algorithm based on spiking associative memory [11][12]. A cortical area is modeled as a local neuron population which receives input from other areas via hetero-associative Hebbian synaptic 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 auto-associative Hebbian synaptic connections which allows both the activation of single assemblies and the representation of superpositions. In comparison to the classical model, this model has a number of additional advantages. For example, assemblies of different size $k$ can be stored, input superpositions of several assemblies can more easily be separated, and it is possible to transmit more subtle activation patterns (e.g., about ambiguities or superpositions) in the spike timing.

In figure 2 the different cortical areas modeled as well as their interconnections and their rough localization in the human brain are shown.
Figure 3 illustrates the overall architecture of our cortical model. Each box corresponds to a local neuron population implemented as spiking associative memory. The model consists of phonetic auditory areas to represent spoken language, of grammar related 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 direct visual attention. The primary visual and auditory areas are part of sensory pre-processing and comprise additional (artificial) neural networks for processing of camera images and acoustic input.

2.1 Robot

The suggested approach is implemented on the PeopleBot base by ActivMedia. To integrate the implemented functionality on the robot we used Miro [13], a robot middleware framework that allows to control the robot’s hardware and facilitates communication with other programs by using Corba. Miro supports distributed computing, i.e. time consuming calculations with low i/o-rates can be outsourced to other computers. Miro also facilitates the usage of the same application on different robot platforms. Hence the software developed here runs as well on the PeopleBot as on other robot bases.

2.2 Attentional system

The attentional system [14] receives input from some higher goal and motor areas which specify the current interest of the robot (e.g., searching for red-colored objects; areas M2 or M2attr in Fig. 3). Subsequently, the camera image is processed by standard computer vision algorithms in order to find regions of
Fig. 4. In the test scenario the robot is situated in front of a table. Different objects are laying on this table. The robot has to grasp or point to specified objects.

interest (see Fig. 5). If an interesting region is found, this part of the image is analyzed in more detail. Features (e.g., orientation histograms) are extracted and transmitted to the object recognition system (see next section). The object recognition system classifies the object in the region of interest and transmits the information to the corresponding visual cortical area (areas V2 and V2attr). After some time, attention shifts to the next region of interest, until this process is interrupted by a cortical area controlling attention (e.g., areas G1 and M1).

![Diagram](image)

Fig. 5. The attentional system currently works on visual sensory input. In the first step, the camera image is searched for regions containing interesting low level features (e.g., blobs of desired color). In the second step, additional features (e.g., orientation histograms) are extracted from the region of interest to be used by the object recognition system.

2.3 Visual object recognition

The visual object recognition system is currently implemented using a hierarchical arrangement of radial-basis-function (RBF) networks. The basic idea of hierarchical neural networks is the division of a complex classification task into several less complex classification tasks by making coarse discrimination at higher levels of the hierarchy and refining the discrimination with increasing depth of the hierarchy. The original classification problem is decomposed into a number of less extensive classification problems organized in a hierarchical scheme. Figure 6 shows a hierarchy for recognition of fruits and gestures which has been generated by unsupervised k-means clustering.

From the activation of the RBF networks (the nodes in Fig. 6) we have designed a binary code in order to express the hierarchy into the domain of cell
Hierarchical network for object classification. Each node represents a RBF network. The end nodes represent classes. A feature and a set of classes are assigned to each node. The corresponding neural network uses the assigned feature for classification.

This code should preserve similarity of the entities as expressed by the hierarchy. A straightforward 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.

This way, the result of the object recognition is transformed into the binary code, and using additional information about space and location from the attentional system, the corresponding neurons are activated in areas V1, V2, V2attr, and V3.

2.4 Language processing system

Our language system consists of a standard Hidden-Markov-based speech recognition system for isolated words and a cortical language processing system which can analyze streams of words detected with respect to simple (regular) grammars. For simplicity, the speech recognition system can also be replaced by direct text input via a computer terminal and a wireless connection to the robot.

Figure 7 shows 15 areas of our model for cortical language processing. Each of the areas is modeled as a spiking associative memory of 400 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 (see Fig. 8).
addition to the auto-associative connections, area A4 has also a delayed feedback-connection where the state transitions are stored hetero-associatively.

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-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 [15][16], 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 and/or Wernicke’s area, and the activation fields as thalamic nuclei.

2.5 Planning, Action and motor system

Our system for cortical planning, action, and motor processing can be divided into three parts (see Fig. 9). (1) The action/planning/goal areas represent the robot’s goal after processing a spoken command. Linked by hetero-associative connections to area A5-P, area G1 contains sequence assemblies (similar to area A4) that represent a list of actions that are necessary to complete a task. For example, responding to a spoken command “bot show plum” is represented by a sequence (seek, show), since first the robot has to seek the plum, and then the robot has to point to the plum. Area G2 represents the current subgoal, and areas G3, G3attr, G4 represent the object involved in the action, its attributes
Fig. 8. Graph of the sequence assemblies in area A4. Each node corresponds to an assembly, each arrow to a hetero-associative link, each path to a sentence type. For example, a sentence “Bot show red plum” would be represented by the sequence (S,Pp,OA1,O1,ok_SPO).

(1) The “goal” areas represent the goal of the current action, and their activation fields correspond to the goal’s attributes and location (e.g., color), and its location, respectively. (2) The “motor” areas represent the motor command necessary to perform the current goal (area G2), and also control the low level attentional system. Area M1 represents the current motor action, and areas M2, M2attr, and M3 represent again the object involved in that action, its attributes, and its location. (3) Similar to the activation fields of the language areas, there are also activation fields for the goal and motor areas, and there are additional “evaluation fields” that can compare the representations of two different areas, for example.

Fig. 9. The cortical goal and motor areas. Conventions are the same as for Fig. 7.
3 Integrative scenario

To illustrate how the different subsystems of our architecture work together, we describe a scenario where an instructor gives the command “Bot show red plum!”, and the robot (“Bot”) has to respond by pointing onto a red plum located in the vicinity.

To complete this task, the robot first has to understand the command. Fig. 10 illustrates the language processing involved in that task. One word after the other enters areas A1 and A3, and is then transferred to one of the A5-fields. The target field is determined by the sequence area A4, which represents the next sentence part to be parsed, and which controls the activation fields which in turn control areas A5-S/P/O1/O2. Fig. 10 shows the network state when “bot”, “show”, and “red” have already been processed and the corresponding representations in areas A5-S, A5-P, and A5-O1attr have been activated. Activation in area A4 has followed the corresponding sequence path (see Fig. 8) and the activation of assembly O1 indicates that the next processed word is expected to be the object of the sentence. Actually, the currently processed word is the object “plum” which is about to activate the corresponding representation in area A5-O1.

![Diagram of language areas](image)

*Fig. 10.* The language areas when processing the sentence “Bot show red plum!”. The first three words have already been processed successfully.

Immediately after activation of the A5-representations the corresponding information is routed further to the goal areas where the first part of the sequence assembly (seekshow,pointshow) gets activated in area G1 (see Fig. 11). Similarly, the information about the object is routed to areas G2, G3, and G3attr. Since the location of the plum is unknown, there is no activation in area G4. In area G2 the “seek” assembly is activated which activates corresponding representations in the motor areas M1,M2,M3. This also activates the attentional system which initiates the robot to seek for the plum as described in section 2.2. Fig. 11
shows the network state when the visual object recognition system has detected the red plum and the corresponding representations have been activated in areas V2, V2attr, and V3. The control fields detect a match between the representations in areas V2 and G3, which initiates area G1 to switch to the next part of the action sequence.

Figure 11. System state of the action/motor model while seeking the plum.

Figure 12 shows the network state when already the “point” assembly in areas G1 and G2 have activated the corresponding representations in the motor areas, and the robot tries to adjust its “finger position” represented in area S1 (also visualized as the crosshairs in area V1). As soon as the control areas detect a match between the representations of areas S1 and G4, the robot has finished his task.

Figure 12. System state of the action/motor model while pointing to the plum.
The cortical model is able to use context for disambiguation. For example, an ambiguous phonetic input such as "bwall", which is between "ball" and "wall", is interpreted as "ball" in the context of the verb "lift", since "lift" requires a small object. Thus the sentence "bot lift bwall" is correctly interpreted.

Fig. 13. A phonetic ambiguity between "ball" and "wall" can be resolved by using context information. The context "bot lift" implies that the following object has to be of small size. Thus the correct word "ball" is selected.

4 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. The implementation in terms of Hebbian assemblies and auto-associative memories generates a distributed representation of the complex situational context of actions, which is essential for human-like performance in dealing with ambiguities.

Although we have currently stored only a limited number of objects and sentence types, we are sure that our approach is scalable to more complex situations. 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 [4][5]. This requires the representations to be 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 more complex grammars from a linguistic point
of view. 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”.

5 Acknowledgement

This research was supported in part by the European Union award #IST-2001-35282 of the MirrorBot project.

References