On Mental Images and 'Computational Semantics'

Heikki Hyötyniemi
Helsinki University of Technology
Control Engineering Laboratory
Otakaari 5 A, FIN-02150 Espoo, Finland
Phone: +358-9-4513327
E-mail: heikki.hyotyniemi@hut.fi

July 27, 1998

Abstract

This paper discusses the need of semantical considerations when implementing mental models. A new view of computational semantics is explained. Further, the basic nature of mental images, based on the presented approach, is studied.

Keywords. Cognitive models, mental images, neural networks, 'computational semantics'

1 Introduction

The linguistically oriented tradition of analytical philosophy has been underlying the research paradigms of cognitive science ever since Wittgenstein. However, it cannot be denied that the truth is 'out there', behind the boundaries of language. The problem of 'meaning' cannot be attacked — the Searle's Chinese room argument still holds.

In this paper, the boundaries of language are attacked in the 'bottom-up' way, starting from raw observation data. The theses that will be elaborated on now are

- semantics cannot be neglected when studying cognitive models, and
- ontological considerations are the key to concrete studies on semantics.

After this, the concrete grounding of 'syntactics' makes it possible to attack the more complex problems in a fruitful way, for example

- it can be argued that mathematically simple, static and associative constructs can explain 'mental images' also in more complex fields, without need of procedural processing.

2 Modeling of mental phenomena

2.1 About models

Various models for explaining the operation of the mental machinery have been developed, the ACT-R and SOAR being among the most prominent and well-known. These model structures are general-purpose frameworks — most of the mental activity can be explained using them, more or less fluently. Usually a great deal of parameters are needed to make the models mimic the observed behavior — but the degrees of freedom make it very difficult to compare different models (see Saariluoma, 1997).

When speaking of mental phenomena, the ‘goodness’ of a model in general is a rather heuristic matter. However, if one proceeds in an ‘engineering-like’ way, there are some alternative points of view to adopt — either,

1. one can weigh (somehow) the ratio between the number of phenomena that are explained by the model vs. the complexity of the model, or
2. one can evaluate (somehow) the value of the properties that emerge for free when that model structure is implemented (this means that the model is fundamentally more or less *correct,* reflecting the behavior of the real system also in those respects that were not specially implemented).

Selecting either of the above starting points, the contemporary mental model structures have their problems. First, even if the models are powerful, their structures are rather complex — it could be asked whether any Turing machine would do the job just as well. Second, it often seems that if one would like to explain some additional property of the mental behavior, the models need to be extended to incorporate the new feature — nothing emerges for free. The traditional models just sound a little ‘artificial’ in this sense.

Due to the very special application field of modeling approaches, the modeling of mental phenomena, one should not be satisfied if the results are not intuitively appealing. If one is trying to mimic intelligence — the real test is whether the algorithm is capable of surprising the person who implemented it! To reach concrete, intuitively appealing results, to escape the Chinese room, one cannot get long without tackling with the problem of semantics. The fundamental role of the mental representations is to capture semantically relevant constructs. The value of the model is ultimately determined by how well it can represent them.

Studying Artificial Intelligence (AI) in the ‘weak’ sense, trying to imitate a human, is a difficult task. In a way, the ‘strong’ interpretation of AI seems to be a *more realistic* goal: even if the claim of ‘self-consciousness’ belongs to science fiction, making the machine ‘understand’ its environment is perhaps not impossible[1]. Understanding the meaning — another way to put it is to speak of *semantics.*

2.2 Computational semantics

In the sense of Wittgenstein (1922), language is traditionally thought to constitute a ‘universal medium’ that cannot be, and need not be, escaped. However, if Wittgenstein were here to rewrite his theses, if he had the modern computing facilities at his disposal, he would perhaps use different words for summarizing his studies on the limits of language:

> What one cannot speak about, that must be *simulated.*

Rather than adopting the utterly skeptical attitude, one can try to attack the problems using non-verbal means.

Now we study semantics in the *naturalistic* setting: the meaning of ‘temperature’ is defined as the numeric reading of the temperature sensor, etc. The semantics of more complex constructs is defined by their context. A good discussion on *procedural semantics* is given by Jackson (1996), trying to connect the numerical atomic measurements to the conceptual level. In this paper, using his terminology, only the *extensional grounding,* or how the real world phenomena are transformed into computational representations is discussed — not the *intensional grounding,* or the connection between these representations and the constructs in a language. In this sense, the emphasis is on the microstructures below the linguistic levels. Rather than semantics, one could perhaps speak of ‘syntactics’.

The basic thing now is to start from simple atomic units that can be understood, and use efficient tools for combining them into more interesting structures.

Assuming that there are relationships between the observation data samples, statistical tools can be used for finding these dependencies automatically. Only after this mechanical manipulation of the data is carried out, the results are transferred into language, the statistical constructs now perhaps being supplied with semantical labels (see Fig. 1). Of course, a natural interpretation of the mathematical constructs is only possible if the mental machinery and the mathematical tool do the same things — what these things are is discussed later.

[1] ‘Understanding’ is necessary to reach truly intelligent behavior: to *act reasonably in a changing environment* (this is adopted as the concrete goal of AI in this study), the machine has to understand the role of different observations

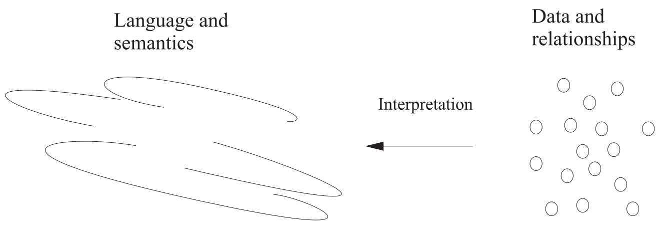

Figure 1: The mathematical machinery resides outside the boundaries of language, thus helping us to escape the Chinese room

2.3 Mathematics as a tool

If we are to abandon natural language as our general-purpose workbench, we must ask whether there are any better ways of formulating expressions. The answer is yes: *mathematics* is a formal 'language' that solves some of the problems that are acute in less formal frameworks. Natural language as the medium imposes some superfluous constraints and makes certain interpretations 'evident' — even if the underlying reality does not support these practices. Take two examples:

1. Natural language is *one-dimensional,* meaning here that the words are connected together one after another in a linear list. When this kind of sequential representations are used as a model for mental phenomena, there are two consequences: First, serial data manipulation approach seems to suffice also when simulating the mental processing; second, the structure of the sentences in a language (conclusions follow the assumptions logically) gives raise to rule-form and causal knowledge representations as exclusive alternatives.
2. Natural language is *granular,* meaning here that a word either is included in a sentence or it is not. This results in the view of 'crisp' concepts and categories with no fuzziness. What is more, the 'granules', or the linguistic concepts, are too massive to allow the manipulation of subsymbolic constructs — but these underlying structures are necessary because they constitute the basis for the higher-level ones, and ability to operate on them is vital.

In many ways, mathematics is a superior language — the above problems that are caused by natural language no more exist in this framework. First, linear algebra (matrix calculus) is well suited for procesing parallel and associative phenomena, and, second, the representations are continuous so that no granularity questions emerge. Mathematics is purely syntactic with no *a priori* semantical commitments, and unintended assumptions cannot creep in accidentally.

To motivate the use of mathematical machinery, it can be shown that concrete new results are obtained when new approaches are adopted. The results can be crystallized in the following paradoxical requirements for an intelligent system, seemingly opposing the common sense:

1. *Elimination* of connections is needed, rather than ever growing complexity. Otherwise, no structural changes in the model are possible.
2. *Scarcity* of available memory is essential, rather that unlimited storage capacity. Otherwise, no optimization of the data model is needed in the first place (of course, a lower limit for memory capacity also exists).

As will be presented later, the question of intelligence can be interpreted as a concrete modeling problem; mathematical tools are necessary to optimize the representations. These optimized representations are responsible for the illusion of 'intelligence'.

3 Towards numerical semantics

3.1 About 'computational intelligence'

There are various paradigms that have been developed to capture the computational properties of the brain — most notably, perhaps, *neural networks* and *fuzzy systems.* The problem with these is that implementing just one aspect of human behavior, either the neuronal structure as in the perceptron networks or the fuzziness of the categories, does not necessarily result in optimal realizations from the holistic point of view.

For example, one can say that recurrent perceptron networks do have the computational power of a Turing machine, so that the mental functions can be implemented also in this medium. A set of perceptrons certainly can be used, but managing thousands of them is not simple — and, after all, what you get is mapping from input to output, with no intuition on the internal structure. Another point is that because the perceptron network with feedback can do anything, it takes very much data to constrain its behavior to what one wants. This problem that becomes more and more acute as the input dimension grows is familiar to anybody having practical experience with back-propagation algorithms. The trivial solution is to add the constraint of function 'smoothness' — but this additional restriction is unnatural if one thinks of the evidence that we already know about the human categorization.

What one needs, is a tailored structure that can do only the 'smart things'.

3.2 Hypotheses

There are some assumptions that are now made. The assumptions are rather bold, but they are useful during the later analyses, and they make the starting point clear.

1. There are no preprogrammed structures in the brain — only the underlying mechanisms of the mental machinery are available. The assumed basic mechanism here is the capability of constructing relationships between observations (note that the Kantian conclusion included causal dependencies between observations, but in this context we will limit ourselves to static structures with no emphasis on the time dimension).
2. The mechanisms on all levels of mental processing are the same. This is an extension of the universality assumption due to Anderson (1983), where it is stated that there cannot exist a separate faculty with specialized computational structures in the brain for all diverse mental skills[2]. Now it is further assumed that the same mental principles apply on the low level, for example in visual pattern recognition, but also on higher levels, like in the internal structures underlying semantic nets and expertise.
3. The mental processes can be studied in reductionistic way, with no emphasis on 'feelings', 'motivations', etc. It is assumed that the learning, or the adaptation of the structures is autonomous and happens with no external control.
4. Evolution has optimized our mechanisms for tackling with the information we get through our senses. This assumption is another interpretation of the famous 'we live in the best of the possible worlds'.

The last of these assumptions is crucial — it means that we can forget about the actual implementation of the neuronal structures and concentrate solely on the information processes (this is the functionalistic argument). Putting it in concrete terms: the problem of understanding the mental processes boils down to the task of modeling the observation data! If our assumed model structure is correct, the computational tools that mechanically optimize the parameters within that framework give essentially the same data model as the mental machinery does.

The mathematical machinery that will be presented is not very complex, so that one cannot capture all properties of the input data. However, this does not matter if the ontological and epistemic assumptions hold: we can only be aware of things that match our own mental structures — what lies outside our cognitive capacity remains there for good[3].

What is the nature of our observation data, then?

[2] Also Wittgenstein did comment on this — he said that the 'meta-levels' of language have to be collapsed, so that it is the same language that is used also for speaking of the language itself. Contrary to his ideas, however, now it is not the linguistic level that is regarded as the universal medium — it is the subsymbolic level instead. It turns out that this approach offers a firm basis for concrete discussion

[3] The epistemological consequences are rather interesting, too — the problem of (subjective) 'truth' changes into the *question of existence of an appropriate data structure.* It is *relevance* that becomes a key concept when speaking of the subject's beliefs: if an observed state of affairs is consistent with other observations, or if it has been encountered sufficiently many times, the corresponding data structures emerge. This view goes well together with the ideas of *constructivism* — it is the old knowledge that controls the interpretation of the new information, defining a 'filter' between the real world and the mental representation of it

3.3 Data ontology

Usually statistical data is modeled using the assumption of Gaussian distribution. If the measurements consist of independently distributed random variables, this assumption is asymptotically optimal and the best one can do. However, the complex and high-dimensional data usually comes from various mutually independent distributions. What is important is that in each of these sub-distributions, *different variables* are needed to express the variations around the center of the sub-distribution. This means that only a subset of the available variables is needed to present the data in a clever way. This means that the data model becomes *sparse* (concrete examples of the practical implications of these sub-distributions are presented later).

It is this sparseness that makes the representations tailor-made and 'smart'. It is not the number of connections between the processing units — it is the number of *missing connections* that is needed to achieve good representation for data.

The individual distributions create clusters in the data space. In principle, all individual measurements define clusters of their own. Because of the memory limitations (and to achieve compression of data), large numbers of these trivial clusters need to be presented using only a low number of parameters. These parameters span linear, rather low-dimensional subspaces around the prototype cluster center. It is also assumed that the nonlinearities there are can be represented using separate clusters for different operating regimes.

This approach can be interpreted also in statistical terms: it is an unsupervised combination of cluster analysis and principal component analysis. It can also be seen as non-orthogonal, sparsely coded factor analysis approach[4].

We only need to implement the above view of data.

3.4 Implementation

There are various ways to apply neural networks to pattern recognition tasks — for example, see Bishop (1995). However, the algorithms often neglect the underlying structure of the data.

The sparseness of the resulting data structures suggests that only nonlinear operation of the algorithms can be capable of doing that. However, to achieve better analyzability and other benefits, the final data model is defined linear.

When to define a new cluster, when not (that means, whether to use qualitative or quantitative representation) — there is no external critic available, and some kind of *self-organization* is needed in the adaptation processes. Due to the difficulty of selecting between the two alternatives, only heuristic approaches exist.

The memory structure that is next concentrated on is a derivation of the Kohonen self-organizing map (Kohonen, 1984). There are N *nodes,* each of which is characterized by a *prototype vector* θ_i, where $1 \leq i \leq N$. The dimension of the vectors is n. The prototype vectors should represent the observed input vectors as accurately as possible — to reach this goal, the standard self-organization algorithm has been modified: rather than constructing a set of cluster centers characterized by the prototype vectors, the prototype vectors are interpreted now as 'coordinate axes' in the input data space, spanning a rather low-dimensional subspace.

The algorithm can be implemented in a straightforward manner as follows.

1. Take the next input vector sample f.
2. Select the node with the best correlation with the (weighted) input vector Wf, that is, determine the 'winner' index c such that the absolute value $|\phi_c|$, where $\phi_c = \theta_c^T Wf$, reaches its maximum value.
3. Calculate the 'neighborhood' parameter $h_{c,i}$ between the network nodes i and the winning node c. This parameter has value near 1 if the nodes are 'near' each other in the net, and lower value otherwise, as presented in (Kohonen, 1984).
4. Apply the Kohonen type adaptation (Kohonen, 1984) of the network using the vector $\phi_c \cdot f$ as input, weighted by W. That means, for each network node i update the vector θ_i as $\theta_i \leftarrow \theta_i + \gamma h_{c,i} \cdot W(\phi_c f - \theta_i)$. The parameter γ is a decaying function of time to assure that the network finally converges.

[4] The findings in neurobiology, specially the explorations in the operation of the visual cortex, seem to justify the assumption of sparse coding of representations (see Olshausen, 1997). What is more, perceptron networks with combined Hebbian – anti-Hebbian learning can be shown to create sparse codes (Földiák 1990)

5. Normalize the feature vectors: $\theta_i \leftarrow \theta_i/\sqrt{\theta_i^T\theta_i}$ for all $1 \leq i \leq N$.
6. Eliminate the contribution of the feature number c by setting $f \leftarrow f - \phi_c/\theta_c^T W\theta_c \cdot \theta_c$.
7. If m features have not yet been extracted, go back to Step 2, otherwise, go to Step 1.

After the network has converged, the prototype vectors represent *features* that can be used to construct the input patterns. That means, given an input vector f, find the sequence of θ_i values as presented in Steps 2 – 7 above (ignoring the updating steps 3 – 5), so that the estimate for f can be constructed as a weighted sum of the features:

$$\hat{f} = \theta_1\phi_1(f) + \cdots + \theta_N\phi_N(f).$$

The features that are not utilized have zero weight. This means that the approach implements a *sparse coding scheme.*

In the presented algorithm, the (diagonal) weighting matrix W is included for generality. Normally, it is an identity matrix, having only ones on the diagonal, but if some of the input vector elements need to be emphasized, the corresponding element can have higher value. On the other hand, if some of the elements are unknown, the weighting is set to zero — this means that input vectors with missing values can also be utilized.

To conclude — the extracted features convey the dependency relations between the input data elements. The dependencies are reflected in their cross-correlation. But it is not only statistical compression that makes the new representation 'cleverer' than the original — it is the sparse coding of tailor-made constructs, or the 'cut connections' that determine the emergent structure in the model. The analysis and other applications of the algorithm below are presented in Hyötyniemi (1997).

3.5 Model interpretation

The computational structure assumed in the algorithm can be interpreted in mental terms as follows. The number N stands for the capacity of the long-term memory, while the parameter m is the size of the short-term (working) memory. It is also assumed that at any instant only the references to the static memory structures and the respective weights are operated on.

The coding of the input data is a rather important thing. The perhaps multimodal input data is transformed mechanically into a single modality, that means, all information is represented as a vector of real-valued data. This means that the complexity of the channels is changed into the problem of high dimensionality. All variables, qualitative as well as quantitative, augment the input vector. The problem space becomes a metric space.

The matrix W can be used for 'attention control', that means, only those diagonal elements differ from zero that are assumed to be known. The unknown variables do not affect the operation of the algorithm — they are only 'filled in' automatically, in an associative fashion.

4 About mental images

4.1 Back to Hume?

It seems that the adopted approach implements the ideas of Hume: if some concept cannot be represented fundamentally by the observation data, it has no semantics (or, in this case, 'syntactics').

Due to the 'economy of thinking' our mental constructs are meaningful also in terms of real world observations. It can be assumed that if the process of dependency structure extraction is continued long enough, linguistic concepts emerge (Fig. 2). This claim is heuristic and only examples can be presented to motivate it. The examples are given in the subsequent sections — briefly, one can summarize that the clusters in the data space can be called features, patterns, or categories, and the subspace coordinates are the data atoms themselves, chunks, or attributes:

- on the lowest levels, features are varied by raw observation data atoms,
- on the intermediate levels, patterns are varied by chunks, and
- on the highest levels, categories are varied by attributes.

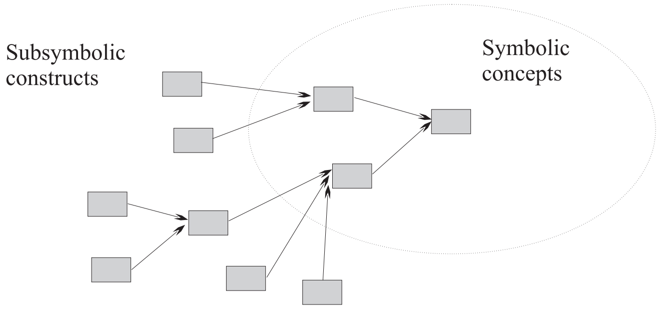

Figure 2: The extracted dependency relations can be further combined, and the results become more and more conceptual — the construction will hopefully have some interpretation also within natural language

4.2 Diverse experiments

In Hyötyniemi (1997) various applications of the presented modeling approach are summarized. There are applications on the low level (face feature recognition) and on the intermediate levels (modeling of chess configurations). Additionally, more or less conceptual information is extracted in two applications (dynamic system structure analysis and contextual modeling of text documents). In the technical field, the low-level feature extraction scheme is being experimented in the image analysis of flotation froths.

It needs to be emphasized that the data model does not distinguish between input and output variables. That means, one can include also *motoric responses* in the mental images — the same model can be applied, at least in principle, for studying the motoric skills.

4.3 Modeling of expertise

It is a well-known fact that the experts have probably not stored their knowledge in the linguistic form — and it is also a well-known fact that often this knowledge cannot be expressed explicitly at all. However, when the language-centered views of expertise are adopted, the these facts have to be ignored. Inevitably, this results in sequential knowledge processing; how could the shift from novice to expert be explained (see Elio and Scharf, 1990).

The hypothesis here is that all areas of expertise can be interpreted as high-dimensional metric spaces with characteristically distributed observation data — the experts have seen many observation instances, and their internal model of the problem field can have just the same structure as was discussed above. Compared to the simpler modeling tasks, the complexity of the expert knowledge is usually caused by the high dimensionality and large number of modalities that need to be mastered. If the expert knowledge is stored as a mental image that has the presented structure, finding the missing pieces of information from it can readily be accomplished using associative search strategies; this means that inference becomes a *matching process.*

Look at Fig. 3: in this very simple case, there are two clusters, and this knowledge can (approximately) be coded in a rule form as

IF $\zeta = \zeta_1$ THEN $\xi = \xi_1$
IF $\zeta = \zeta_2$ THEN $\xi = \xi_2$.

However, it can be argued that the above deconstructibility is just a lucky coincidence: normally, low-dimensional projections cannot be achieved. What is more, the linguistic (rule-form) representation for the knowledge neglects the special characteristics of the categories — see Fig. 4. Using the feature-based approach, this kind of typically 'elongated' clusters can be efficiently modeled: the features define 'hidden variables', the corresponding subspaces being tailored to fit the data distribution.

Expert systems have been constructed using probabilistic reasoning; the granularity problem caused by the logic formalisms can be relieved when real-valued probabilities are involved — for example, see Pearl (1988). Trying to construct longer sequences of reasoning based on projections of the probability distribution, however, often results in inconsistencies caused by the neglected mutual dependencies between the variables. The same problem is visible in fuzzy inference systems: only the most significant variables can be included in the fuzzy rules. The alternative approach that is adopted now is not to try to project the high-dimensional distribution, but model it as a whole. No variables, no matter how negligible, are ignored altogether; along with other minor variables they can have some macroscopic effects in the final outcome.

Very much can be done in the framework of static, associative data recall, but, of course, more complex

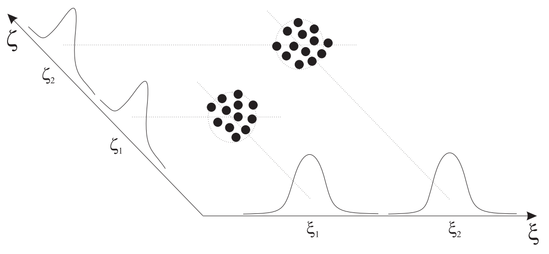

Figure 3: The projections of the high-dimensional data space onto two dimensions. The dots represent observed data samples being concentrated on two distinct clusters

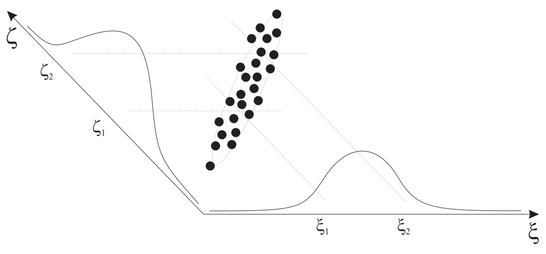

Figure 4: A more typical complex data distribution, where the simple rule-form knowledge representation approach no more works: the categories should have inner fine structure

information processing cannot be accomplished using only these data manipulation mechanisms. After all, the presented approach only tries to fill the gap between symbolic and subsymbolic.

4.4 Sructured knowledge representations

The feature-based knowledge representation can directly be used for associative reconstruction of observation data: there is a close connection to the discussion in the previous section, and the basic data structure will now be concretized using a simple example.

The example here resembles the experiments presented in (Ritter and Kohonen, 1989): concepts are described in terms of a few attributes, and using these attributes as input data, automatic classification in categories is carried out using a self-organizing map. Now, however, not only the categories are searched for but also the fine structure, or the features, using the presented algorithm. The input data is as follows (note that the number of samples is too low to make any definite statistical conclusions):

		f_{goose}	f_{canary}	f_{penguin}	$\cdots$	f_{horse}	f_{dog}	f_{mouse}	$\cdots$	f_{bat}	f_{monkey}	f_{man}
goose	→	1										
canary	→		1									
penguin	→			1								
robin	→											
sparrow	→											
swallow	→											
swan	→											
pigeon	→											
eagle	→											
chicken	→											
horse	→					1						
dog	→						1					
mouse	→							1				
donkey	→											
cow	→											
sheep	→											
cat	→											
bat	→									1		
monkey	→										1	
man	→											1
size	→	0.5	0.1	1.0		2.0	1.0	0.1		0.1	1.0	1.7
flies	→	1	1							1		
feathers	→	1	1	1								
fur	→					1	1	1		1	1	
2 legs	→	1	1	1						1	1	1
4 legs	→					1	1	1				
eggs	→	1	1	1								

Above, the elements that are not explicitly shown are zeros. The input element `'size'` is fuzzy-valued (note that the maximum is not 1 now: for better readability, the value directly reflects the actual size in meters), while other variables are qualitative or logical. Complete sets of attributes are used in adaptation; the presented f vectors are input in the algorithm iteratively in random order, and all attributes are equally weighted. The parameters are as follows: $n = 27$, $N = 6$, and $m = 3$; the resulting normalized (slightly streamlined) feature vectors are shown below:

		θ_{bird}	θ_{animal}	θ_{special}	θ_{big}	θ_{flying}	θ_{human}
goose	→	0.05			0.03	0.0	
canary	→	0.05			0.0	0.0	
penguin	→	0.05			−0.17	−0.27	
robin	→	0.05			0.0	0.0	
sparrow	→	0.05			0.01	0.0	
swallow	→	0.05			−0.03	0.0	
swan	→	0.05			0.09	0.0	
pigeon	→	0.05			0.03	0.0	
eagle	→	0.05			0.19	0.0	
chicken	→	0.05			−0.09	−0.23	
horse	→		0.09		0.24	−0.0	
dog	→		0.09		0.11	−0.0	
mouse	→		0.09		−0.57	−0.0	
donkey	→		0.09		0.12	−0.0	
cow	→		0.09		0.23	−0.0	
sheep	→		0.09		−0.02	−0.0	
cat	→		0.09		−0.19	−0.0	
bat	→			0.21	−0.06	0.28	
monkey	→			0.23	0.04	−0.17	−0.45
man	→			0.25	0.03	−0.14	0.62
size	→	0.12	0.36	0.35	0.61	−0.41	0.30
flies	→	0.40		0.21	0.17	0.71	−0.0
feathers	→	0.52			−0.03	−0.0	
fur	→		0.64	0.44	−0.11	0.0	−0.55
2 legs	→	0.52		0.69	−0.02	−0.0	0.0
4 legs	→		0.64		−0.07		
eggs	→	0.52			−0.03	−0.0	

First, it can be noticed that three basic categories are formed (the first three of the features): the first stands for 'birds', the second for 'typical animals', and the third for the rest 'special animals' (seemingly two-legged ones). When attribute vectors are input, one of these features is always selected first to fix the cluster center. The other three features are (kind of) special 'flavors' that make the input vectors separable. The first of these features can be labeled as 'big', because it has high positive correlation with the size input; the second represents the ability to fly, and the third could perhaps be paraphrased as 'human-like'. As an example, when the 'horse' attributes are input in the algorithm, the following reconstruction is generated:

$$f_{\text{horse}} \approx 0.83 \cdot \theta_{\text{animal}} + 0.25 \cdot \theta_{\text{big}},$$

so that the internal view of a horse is a 'rather big animal'. Note that there is not enough memory resources to contain all information that is available (only 6 memory units available, whereas 27 input units should be stored). That is why, the horse instance cannot be exactly recalled. Traces of the horse input exists, however, in the animal prototype; the prototypes contain much of the 'forgotten' information, where these hidden pieces of information together constitute the actual *semantics,* or the context of the concept. The interconnections between variables actually define a kind of *semantic network* between concepts.

The default values are defined by the category center; for example, a typical animal has four legs and fur, and its average size is 36 cm. An interesting detail is that being 'big' seems to mean different thing for birds and for animals — this parallels with the generally accepted context-sensitivity of properties.

5 Conclusions

One of the most popular models for mental images is due to Kosslyn (1980). The presented scheme differs very much from that — for example,

- there is no fixed imagery — the mental image is a 'personal' view, filtered through the learned schematic image prototypes, and

- the presented model is easier to analyze — there is only one storage and information representation formalism.

The traditional view of mental imagery, consisting of a large number of very elementary spatial matrix-form representations of visual images, has been criticized also by Pylyshyn (1981). It needs to be noted that now the idea of mental images has been directly extended to other, non-visual domains: it is assumed that the same principles govern all of the mental representations — motoric skills as well. More traditional views of concept formation are presented, for example, by Fisher *et al* (1991).

Compared to the standard architectures of cognition, there are some nice surprises (cf. the discussion above): for example, the model has some properties that have emerged automatically. One of these properties is the fuzziness of categories. There also exists biological justification for the presented model: the visual V1 level on the cortex is known to extract features from image data. The sparse coding scheme can also be implemented using simple perceptron elements using Anti-Hebbian learning (Haykin, 1994).

Only static and associative information representation was discussed above. Combining various levels of the presented analysis, more complex situations can be managed (note that one level is enough to eliminate the redundancies between the data elements, and nothing is gained if many levels are used; however, additional information, like results from various other analysis can be processed on the proceeding levels). It is an open question how spatially or temporally distinct and separately analyzed mental images could be combined in a general way to create a 'higher-level' mental image that could incorporate, for example, causal dependencies.

Acknowledgements

I want to thank professor Pertti Saariluoma and his research group for constructive comments and discussions. Financially this research was supported by the Academy of Finland.

References

Anderson, J.R. (1983): *The architecture of cognition.* Harvard University Press, Cambridge, Massachusetts.

Bishop, C.M. (1995): *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford.

Elio, R. and Scharf, P.B. (1990): Modeling novice-to-expert shifts in problem-solving strategy and knowledge organization. *Cognitive Science,* **14**, pp. 579–639.

Fisher (Jr.) D.H., Pazzani, M.J., and Langley, P., eds. (1991): *Concept Formation: Knowledge and Experience in Unsupervised Learning.* Morgan Kaufmann Publishers, San Mateo, California.

Földiák, P. (1990): Forming sparse representations by local anti-Hebbian learning. *Biological Cybernetics,* **64**, No. 2, pp. 165 – 170.

Haykin, S. (1994): *Neural Networks. A Comprehensive Foundation.* Macmillan College Publishing, New York.

Hyötyniemi, H. (1997): On the Statistical Nature of Complex Data. In *SCAI'97 — Sixth Scandinavian Conference on Artificial Intelligence; Research Announcements* (ed. Grahne, G.), Helsinki University, Department of Computer Science, Report C-1997-49, Helsinki, Finland, pp. 13 – 27.

Jackson, S.A. (1996): *Connectionism and Meaning: From Truth Conditions to Weight Representations.* Ablex Publishing, New Jersey.

Kohonen, T. (1984): *Self-Organization and Associative Memory.* Springer-Verlag, Berlin.

Kosslyn, S.M. (1980): *Image and Mind.* Harvard University Press, Cambridge, Massachusetts, 1980.

Olshausen B.A. and Field D.J. (1997): Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vision Research,* **37**, pp. 3311 – 3325.

Pearl, J. (1988): *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, San Mateo, California.

Pylyshyn, Z. (1981): The imagery debate: analogue media versus tacit knowledge. *Psychological Review,* **88**, pp. 16 – 45.

Ritter, H. and Kohonen, T. (1989): Self-organizing semantic maps. *Biological Cybernetics,* **61**, pp. 241–254.

Saariluoma, P. (1997): *Foundational Analysis: Presuppositions in Experimental Psychology.* Routledge, London.

Wittgenstein, L. (1922): *Tractatus Logico-Philosophicus.* Routledge and Kegan Paul, London (2nd edition).