TOWARDS PERCEPTION HIERARCHIES

Heikki Hyötyniemi
Helsinki University of Technology
Control Engineering Laboratory
P.O. Box 5400, FIN-02015 HUT, Finland

It has been demonstrated earlier that when the theory of complex systems is applied to cognitive science, interesting results can be reached on the level of individual perceptions. However, concentrating on just a single observation/perception pair at a time, truly convincing artificial intelligence systems can never be constructed. This paper discusses the possibilities of extending the simple model, so that various separate perception processes could interact and this interaction process could be maintained.

1 INTRODUCTION

It has been demonstrated (in a more or less convincing way) that the emergence of individual "perceptions" from observations can be modeled and maintained (see [8])[1]. However, this all happens in "toy worlds" where the tasks to be performed are simplified to an extreme. From the point of view of cognitive plausibility, or from the point of view of practical applications, this kind of constrained view is not very fruitful.

It has turned out that these issues of scaling a model up are always extremely difficult in the field of AI. And when something structurally more sophisticated should be integrated in the models, the problem is even more challenging. In [6] it was just assumed that there exists some "pool" of elementary observations that are then somehow circulated and recirculated in the processing machinery intelligently ... Even the most ambitious cognitive models like ACT-R [1] and SOAR [13] do things in a rather reductionistic way.

It is evident that some kinds of "perception hierarchies" should be constructed. It has been proposed that his kind of hierarchic modularity should be characteristic to all kinds of complex systems; Herbert A. Simon motivated this claim and further formulated the "almost decomposability" and "empty world" hypotheses (see [15]). So how to model and master hierarchical structures that interact with each other in a cognitive modeling environment? And how to do it without introducing too many extra structures above the lower-level perception machinery?

The starting point here is that of [8]: The only manipulation mechanism that is available now can be written as

$$x(k+1) = f_{\text{cut}}(Ax(k)), \tag{1}$$

where A is a real-valued matrix, and the "generalized cut" function $f_{\text{cut}} : \mathcal{R}^{\dim(x)} \rightarrow \mathcal{R}^{\dim(x)}$ is defined elementwise as

$$f_{\text{cut},i}(x) = \begin{cases} x_i, & \text{if } x_i > 0, \text{ and} \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

[1] Again, it needs to be commented that terms like *perception* are used here in a rather liberal way

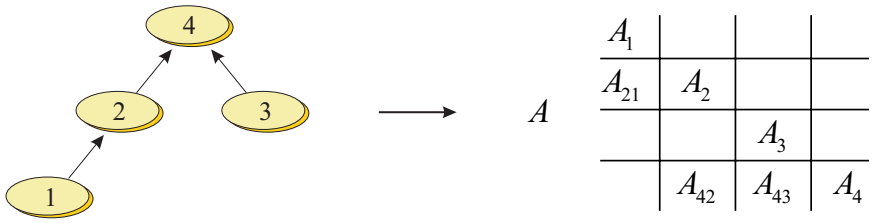

Figure 1: Hierarchy as coded in the matrix form: Matrices A_1, A_2, A_3, and A_4 are dense, meaning that the corresponding variables are closely connected, whereas matrices A_{21}, A_{42}, and A_{43} are sparse, meaning that there are only a few connections; other elements in the matrix are zeros

for all $i = 1, \ldots, \dim(x)$. This iteration is continued, starting from some initial state $x(0)$, until the state converges to some "internal image" $\bar{x}$. It has turned out that implementing individual cognitively plausible tasks is possible in such a simplistic framework (see [8]) — but how about the next level, constructing a *system* with some internal structure out of the building blocks?

As demonstrated in [7], *everything* can be accomplished in the above framework (1), that is, all computable functions can be implemented as simple dynamic systems. This means that different kinds of control structures can also be implemented that would take care of the lower-level perception processing tasks — but is it feasible, can the complexity of the resulting system be managed? What one needs is some kind of toolbox of conceptual and pragmatic tools for mastering the complexity. Studying such tools will be elaborated on in this paper. The view here is rather synthetic: The approach is more AI-like than cognitive, not trying to develop "deep" claims of the real cognitive phenomena, but hopefully developing methods for implementing some kind of practical applications where recognition of the sophisticated environments is necessary.

It needs to be recognized that the results when the iteration (1) is applied typically become sparsely coded; that is, because of the "cutting" nonlinearity, some entries in $\bar{x}$ become zeros. This kind of emerging sparsity means that the dependencies between entities become automatically structured; the zeros block the propagation of the related signals. The matrix A, if it is constructed based on observed correlations in $\bar{x}$, typically also becomes sparse with many zero entries, and such sparse structures can be interpreted in terms of hierarchies (see Fig. 1). However, these "ad hoc hierarchies" cannot easily be mastered: Analysis is needed, and more powerful conceptual tools are necessary to reach new intuitions.

In concrete terms, there are (at least) two problems concerning different kinds of fundamental structures needed if one wants to implement some "practical cognition":

1. Master the relationships and interactions between structurally related (same-level) patterns (spatial or temporal).

2. Master the control of logically/functionally ordered (successive, iterative, or conditional) processes.

These will be studied in what follows. In both cases, the hierarchies can be implemented "hard way", that is, by implementing clumsy, high-dimensional (sparse) matrices so that only simple control structures are needed, or in a "easy" way, resulting in streamlined operation but necessitating some kind of more complicated extra control structures.

2 IMPLEMENTING SPATIAL/TEMPORAL STRUCTURE

It can be claimed that the most insightful studies into the human cognition were carried out already in the 1700's by Immanuel Kant — completely by introspection. There are two basic ideas in his studies relevant still today:

1. The always subjective observation can become an objective perception, common to all observers, not because there is some preprogrammed mind in us but because the underlying mechanisms that carry out the processing of the observations are similar. This idea is elaborated on — using today's tools! — in various earlier papers, for example, see [5].

2. In addition to these mechanisms, there are some architectural hard-wired principles: For example, mechanisms for perceiving *space* and *time* are innate. These special capabilities are reflected in our tendency to construct spatial and causal dependency structures, respectively.

In this section, the latter phenomenon is being studied. This kind os capability of maintaining spatial order is essential for example when visual views are being analyzed; maintenance of causal structures is necessary when trying to model "functional chunks", or when analyzing, for example, strings of phonemes, words, and sentences to construct words, sentences, and whole stories, respectively. In the visual case, the problem is two (or perhaps three?) dimensional, whereas in the auditory case, there are only two directions, backward and forward.

It is clear that the framework (1) needs to be extended to capture the necessary functionality; what is the minimal extension so that some relevant functionality can be implemented? First, it can be noted that there is no need to distinguish between the time-domain and spatial domain — modeling adjacent phenomena can follow the same principles in both cases. The seemingly significant difference between the two domains is mainly caused by the very different physical appearance of the related sensing devices[2].

One should define a framework where adjacent elementary perception processes can affect each other. An age-old AI idea called *blackboard technique* is closely related here: Different agents independently operate on the objects visible on the common workarea, this workarea being (in this case) a spatial image where higher and higher level analyses are carried out in parallel, utilizing analyses in neighboring locations (indeed, one could also formulate this within the now so popular *cellular automaton* paradigm). One only needs to implement the basic idea in the current framework.

Continuing the "hard way", preferring simplicity to sophistication, the neighboring elements can be included in the individual perception process in a straightforward way. Formally, assuming the case is one-dimensional, a single elementary perception unit has two neighbors possibly affecting its behavior. For this "focus point" one can define an extended input data structure as follows:

$$\xi = \begin{pmatrix} x \\ \hline x^{\leftarrow} \\ \hline x^{\rightarrow} \end{pmatrix}. \tag{3}$$

[2] "Time is just another coordinate" — sounds like some kind of a *cognitive relativity theory*!

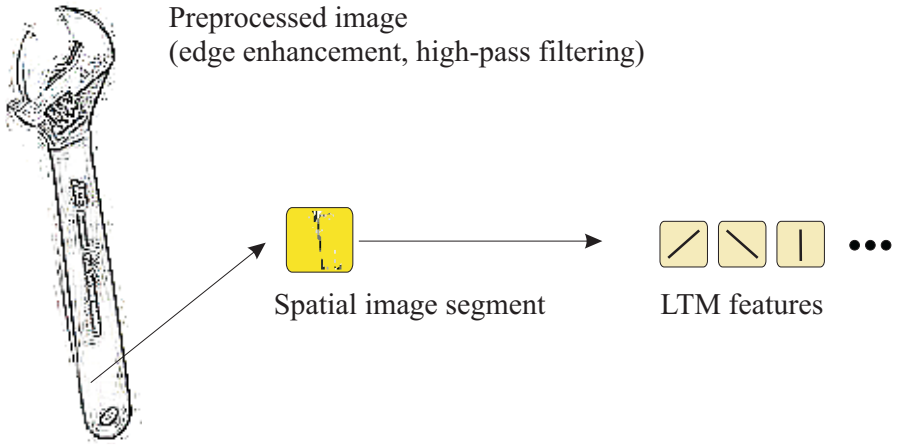

Figure 2: Decomposition of a view into "visual atoms" that are stored in the long-term memory

Here, it is intuitively assumed that the first of the neighbors $x^{\leftarrow}$ is to the "left" of the focus point, denoted by x, and $x^{\rightarrow}$ is to the "right". Simple forward chaining inference can now be implemented as presented in [8]:

$$x(k+1) = f_{\text{cut}}(A\xi(k)). \tag{4}$$

It needs to be noted that A is no more square; what is more, the input vector has to be reconstructed before each inference step, because the neighboring perceptions can also have changed. This kind of augmentation and inference has to be carried out for all perception units in the grid in parallel; this parallel operation can be presented in a matrix form conveniently as

$$X(k+1) = f_{\text{cut}}(A\Xi(k)), \tag{5}$$

where the matrices X and Ξ contain the individual perception vectors as columns:

$$X = \left(\, x^{\Leftarrow} \, \middle| \, \cdots \, \middle| \, x^{\Rightarrow} \, \right) \qquad \text{and} \qquad \Xi = \left(\, \xi^{\Leftarrow} \, \middle| \, \cdots \, \middle| \, \xi^{\Rightarrow} \, \right). \tag{6}$$

Here, $x^{\Leftarrow}$, for example, denotes the first element in the ordered set of perceptions, and $x^{\Rightarrow}$ denotes the last one. The cut function f_{cut} is no more vector-valued but a matrix; however, it still operates on data in the elementwise fashion.

As an example, analysis of a visual image is studied here, where two-dimensional spatial relationships need to be modeled. This experiment also has some cognitive and physiological evidence: It has been recognized (for example, see [14]) that on the very lowest level of visual image processing observations are deconstructed into features consisting of line segments, etc. (see Fig. 2). This kind of coding can be motivated also based on information theoretical considerations, and such features can also be automatically extracted from visual images using statistical tools (see [2] and [3]). Anyhow, it happens that it is exactly this kind of elementary visual atoms that can be processed in the proposed framework in a rather straightforward way: Patterns are matched against visual elements, and if they match, the detected patterns can be used for *next-level* pattern recognition; exhaustive pattern matching process is carried out, so that all locations of the image are studied as possible candidates for pattern centers[3].

[3]Note that there is a traditional pattern recognition method called *Hough transform* that closely resembles this approach (for example, see [16])

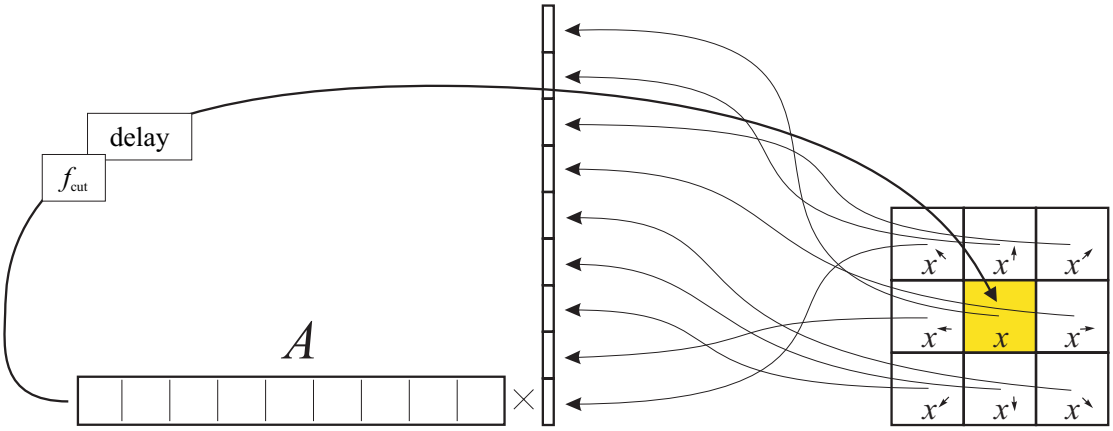

Figure 3: Segmentwise analysis of a visual image. The same structure is copied for each image element

The monochrome images are now coded as two-dimensional sets of pixels. The construction of ξ is in this case more complicated than it was above because there are more neighbors; the two-dimensional view has to be collapsed into a one dimensional representation (see Fig. 3):

$$\xi = \begin{pmatrix} x \\ \hline x^{\uparrow} \\ \hline x^{\nearrow} \\ \hline \vdots \\ \hline x^{\nwarrow} \end{pmatrix} . \tag{7}$$

On the boundaries of the image, some neighbors are missing; those missing values are 0's. Note that if the dimension of an individual perception x is n, the dimension of ξ is $9 \cdot n$.

Faces are the first patterns that a newborn baby learns to recognize. It has been claimed that a not-so-old baby can be fooled if he/she is shown an image where there are two dots above a line segment ... this kind of "visual world" will now be constructed (see Fig. 4).

Now let us concentrate on the construction of a single perception. On the lowest conceptual level there is the "Dot", on the slightly higher there is the (horizontal) "Line"; and only one more higher-level construct, "Face", is needed in our simplistic world. These are expressed in the only 5-dimensional perception vector as

$$x = \begin{pmatrix} x_1 \\ x_y \\ x_{\text{Dot}} \\ x_{\text{Line}} \\ x_{\text{Face}} \end{pmatrix} . \tag{8}$$

In addition to the above conceptual entries, there are some additional ones: $x_1 = 1$ is a dummy variable that is needed to reach more expressional power and to make more complicated inference results possible; if this were neglected, zero input (empty image) would always result in empty perception (if an external 1-input is supplied, the state

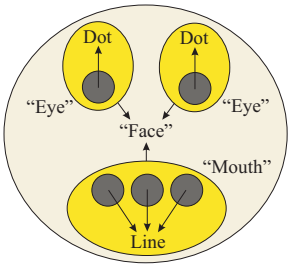

Figure 4: "Infant's world": Hierarchic construction of a *face*

element x_1 becomes obsolete). Variable x_y represents the actual sensation level, the pixel value in that location — zero means blanc and 1 means occupied (black). An (unoptimized) visual inference system could now be constructed of the following rules (note that all these can readily be implemented in the form (4)):

$$
\begin{aligned}
x_y &\leftarrow f_{\text{cut}}(x_y), \qquad x_1 = f_{\text{cut}}(x_1) \\
x_{\text{Dot}} &\leftarrow f_{\text{cut}}\left(x_y - x_y^{\uparrow} - x_y^{\nearrow} - x_y^{\rightarrow} - x_y^{\searrow} - x_y^{\downarrow} - x_y^{\swarrow} - x_y^{\leftarrow} - x_y^{\nwarrow}\right) \\
x_{\text{Line}} &\leftarrow f_{\text{cut}}\left(\tfrac{1}{3}x_y + \tfrac{1}{6}(x_y^{\leftarrow} + x_{\text{Line}}^{\leftarrow}) + \tfrac{1}{6}(x_y^{\rightarrow} + x_{\text{Line}}^{\rightarrow})\right. \\
&\qquad\qquad \left. - x_{\text{Dot}} - x_{\text{Dot}}^{\uparrow} - x_{\text{Dot}}^{\nearrow} - \cdots - x_{\text{Dot}}^{\nwarrow}\right) \\
x_{\text{Face}} &\leftarrow f_{\text{cut}}\left(\tfrac{1}{2}x_{\text{Dot}}^{\nwarrow} + \tfrac{1}{2}x_{\text{Dot}}^{\nearrow} + \tfrac{1}{2}x_{\text{Line}}^{\downarrow} - \tfrac{1}{2}\right).
\end{aligned}
$$

Note that x_y behaves like an integrator, always retaining its original value; this means that the original image is not affected by the perception refining process. The value of $x_y(0)$ is 1 or 0, depending on whether there is a pixel there or not in the original image — this is the only prefilled value in the perception vector when the iteration is started (in addition to $x_1(0) = 1$); other entries are originally zeros but may change during the inference process. A pattern is a "Dot" if the corresponding pixel is occupied but all of its neighbors are empty. A (horizontal) line is a combination of adjacent pixels; if the neighbors have already been perceived as line segments, this enhances the "line-likeness" — meaning that *positive feedback* is introduced in the system. A Face, then, is a combination of two dots and a line in an appropriate constellation. The "standard face", according to the rules, is shown in Fig. 5, whereas more or less deformed faces are shown in Figs. 6 and 7. In Fig. 8, a wider image is shown, where the random sets of dots are more or less face-looking.

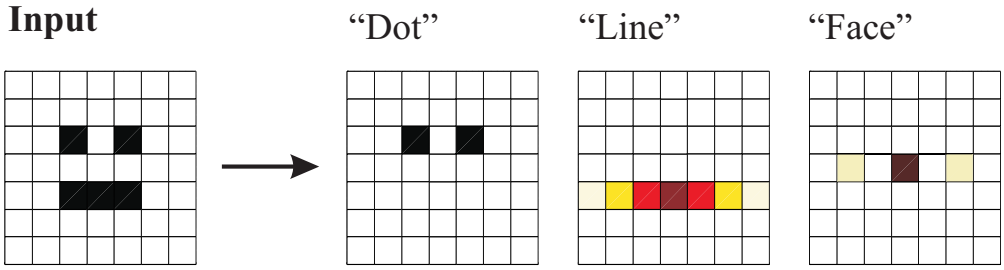

Figure 5: Nominal "face". The first image represents the original image coded pixelwise, that is, the X matrix has been projected along the x_y axis, and the resulting 49 dimensional vector has been rearranged as a two dimensional 7×7 image; in the latter images, the converged $\bar{X}$ is projected along the x_{Dot}, x_{Line}, and x_{Face} dimensions, respectively. Black denotes "1", whereas blanc stands for "0". This world is not binary — also non-integer values are possible

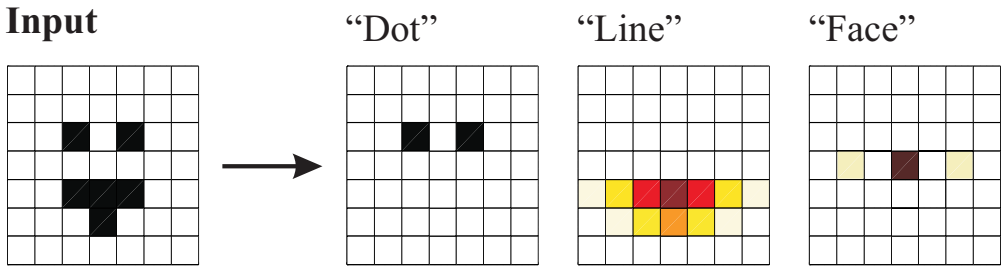

Figure 6: Deformed face: "Smile"

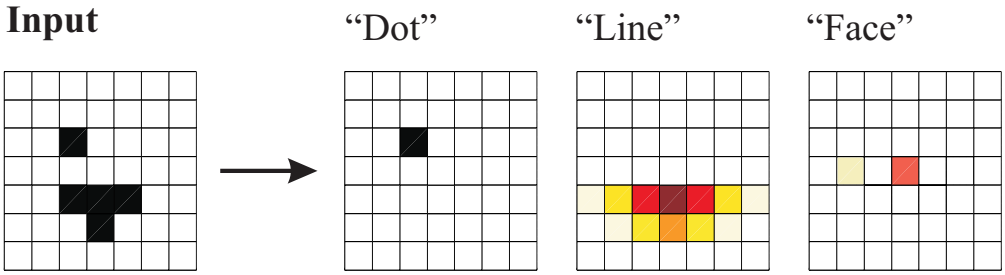

Figure 7: More deformed: Face or not a face?

The determination of the parameters in the inference tree can be carried out using not only explicit "programming" but also, for example, using the *learning by examples* scheme (see [8]); on the lowest level, where the statistical relevance is more fundamental than conceptual, the correlation structures detected in the natural images can be exploited by, for example, the GGHA algorithm [4].

If there are various alternative patterns that share the same components the concept definitions interact; a special perception somewhere can affect the other ones considerably, and maintaining stability (not to speak of optimality) becomes difficult if some kind of manual programming is applied. However, it can be assumed that the *feature matching* alternative (as presented in [8]) becomes more and more prominent as correlations between perceptions are being detected and utilized. This kind of "spread of activation" makes the perception process more robust. It can even be claimed that the shift from novice to expert (see [11]) can be seen as such a transition from following rules

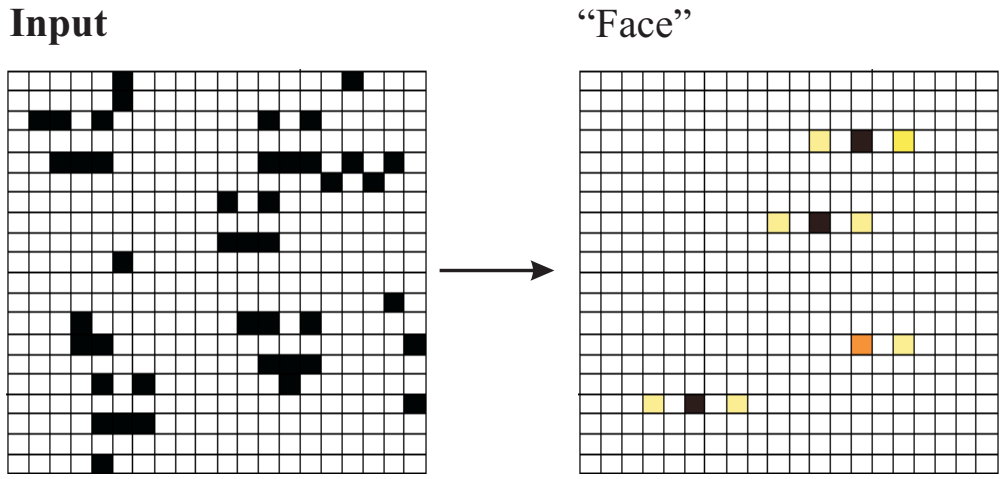

Figure 8: Detecting noisy faces in an image

Figure 9: An example where feedbacks from later processing levels (should) affect the lower-level analyses

to matching of patterns. The simple one-pass hierarchic forward chaining changes into more holistic process, where alternative patterns compete against each other, and also higher-level results can be recirculated to lower levels. For example, if a table has been detected, it should propagate the probability of detecting also chairs (see Fig. 9).

Above, the wealth of spatial information was coded in the hard way, explicitly constructing an inference machinery for all image locations, and representing all partial perception results explicitly in the vectors x. When implementing a larger system, where dozens of different level concepts need to be implemented, the dimension of the vectors x becomes huge. A more sophisticated approach would be to implement some kind of *attention control,* so that the emphasis would be concentrated only on those locations where some kind of *novelty* or mismatch between prior analyses were detected. This kind of attention control should take place in two dimensions: First spatially, so that different parts of the image were studied at a time; but, in addition to this location-wise attention control, some mechanism for *layer-wise* attention control would be needed, so that inferences that are not topical are not tried in vain — typically only a short segment of x is needed and affected (see Fig. 10). However, when studying this kind of added sophistication, one soon is dealing with problems of consciousness and motivations, and such discussions are skipped here. Perhaps the most appropriate approach is, nonetheless, to divide the complicated many-level perception tasks into segments and model them separately, explicitly determining the evaluation orders (see next section)?

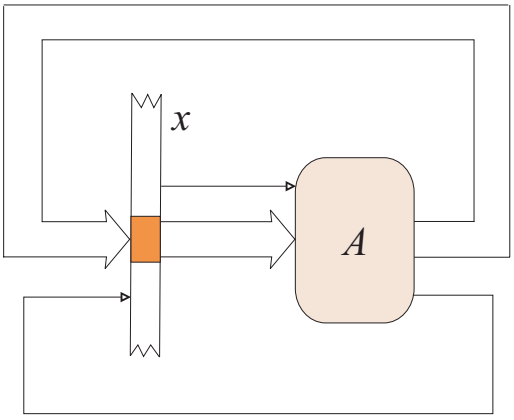

Figure 10: How to express blockwise manipulation of the perceptions in a mathematically efficient and natural way?

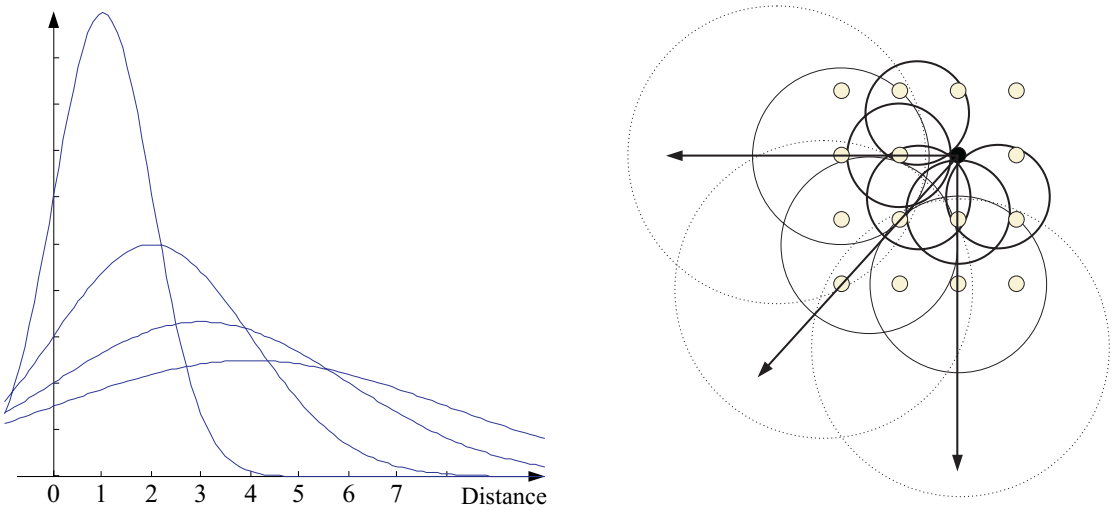

Figure 11: How some kind of scaling invariance could be reached: Different kinds of neighborhoods are studied, the vectors $x^{\leftarrow}$, $x^{\swarrow}$, $x^{\downarrow}$, etc., being calculated as weighted averages of various x's, the weightings being determined by some appropriate distribution functions. When the radius σ varies (randomly), different scales are catered

When studying the problem of visual image analysis closer, one soon detects that the above dsicussions were much too simplistic: The key problems that remain unanswered are how to implement transformations like scaling and rotation of images. Should one introduce some kind of "neighborhood functions" with varying radiuses to circumvent the problems of scaling, etc.? For example, in Fig. 11, an approach towards reaching scaling invariance is presented: During different time steps, the vectors ξ are determined in different ways, varying the scaling factor σ. Similarly, rotations could in principle be implemented in the same way, modifying the determination of the neighbors. To implement this scheme, the deterministic inference process formulation (5) needs to slow down to achieve robustness against stochastic phenomena; this can be reached if one defines, for some "forgetting factor" $0 \ll \lambda < 1$,

$$X(k+1) = f_{\text{cut}}(\lambda \cdot X(k) + (1-\lambda) \cdot A\Xi(k)). \tag{9}$$

3 MAINTAINING SEQUENTIAL ORDERING

In the previous section, maintaining order among parallel perceptions was studied. An equally challenging problem is to maintain the ordering when the perceptions should be kept strictly sequential. In engineering work, block diagrams have turned out to be good, intuitive tools for designing and understanding such systems, so that our goal here is the somehow represent the huge A matrix in easier-to-manage blocks.

As shown in [7], it is exactly the same structure (1) that can be exploited to implement any imaginable algorithm. And it is algorithms that are powerful conceptual tools that can be used to define sequential tasks. However, the algorithmic subsystems and pattern matching subsystems cannot directly be implemented in the same matrix A: The reason for this is that the data processing takes place in so different forms in these two cases, and one first has to construct some kind of *interfaces* between them.

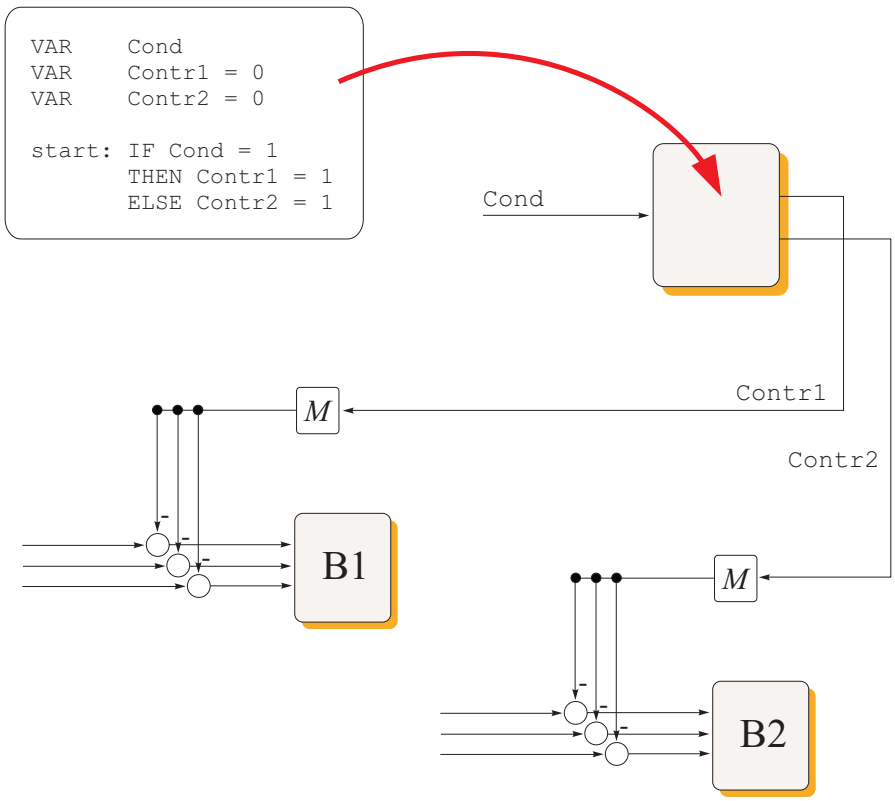

Figure 12: *From logic domain to continuous variables.* Separate "control blocks" can control other blocks. If the signal `Cond` equals 0, block B1 is deactivated; if it equals 1, block B2 is deactivated

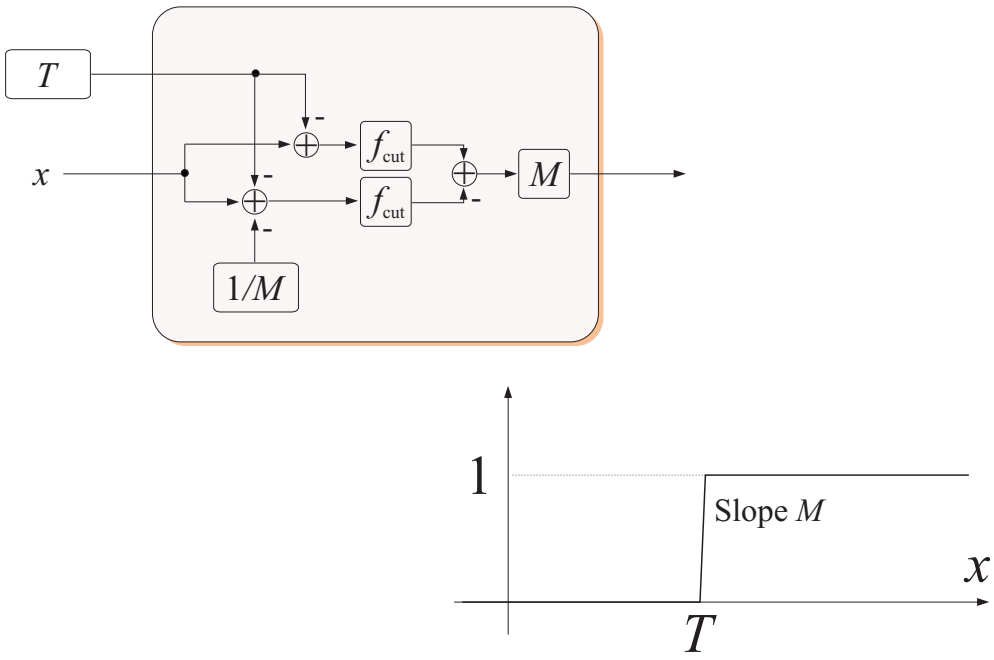

Figure 13: *From continuous variables to logic domain.* Two additional dimensions in the system matrix A are needed to implement a comparator ("Is x larger than threshold T?")

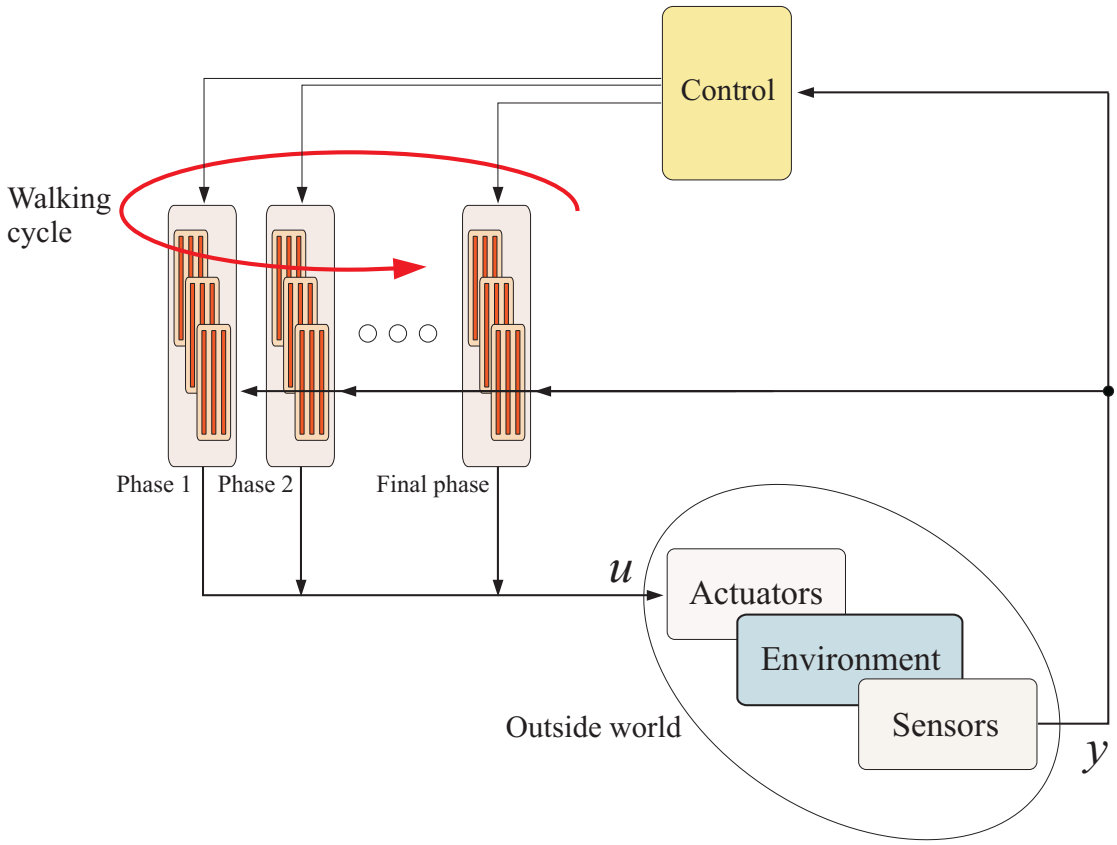

Figure 14: An example where sequential control is needed — *walking*. The individual control blocks for different phases of the cycle are further divided in sequences of linearized operating regions (see [9])

The pattern matching process operates with continuous-valued signals, whereas the algorithmic process with integer values. To integrate these, the two-way connections need to be supplied — first, transferring activation commands from the algorithmic module to the pattern matching modules can be accomplished as shown in Fig. 12; second, changing of continuous variables into logic values can be carried out as shown in Fig. 13 (M is a constant of high value). In both cases, the structures can be implemented in the form (1) — this means that the visually constructed block structure can readily be compiled into an A matrix for execution, with the expense of a few additional state components in the x vector.

Programmable control of simpler cognitive tasks is needed, for example, if *walking* or some other complicated task is being learned by a multi-legged robot: There are phases of different kinds during the walking cycle (see Fig. 14). There is an ordering between the phases, and switching between the modes is dependent of the state. If the appropriate conditions are not fulfilled, the deactivation signal blocks the underlying blocks. During each sub-phase, the behavior can locally be optimized (see [9]).

Now there are two levels of hierarchy in the cognitive model, the lower level being continuous-valued and thus adaptable according to measurement data, but the higher level is fixed and preprogrammed, and this "wiring" cannot be adapted by any continuous update strategy. That is why, good tools are needed to make this manual labor feasible — see Fig. 15 (however, the threshold values T, etc., can still be automatically adapted).

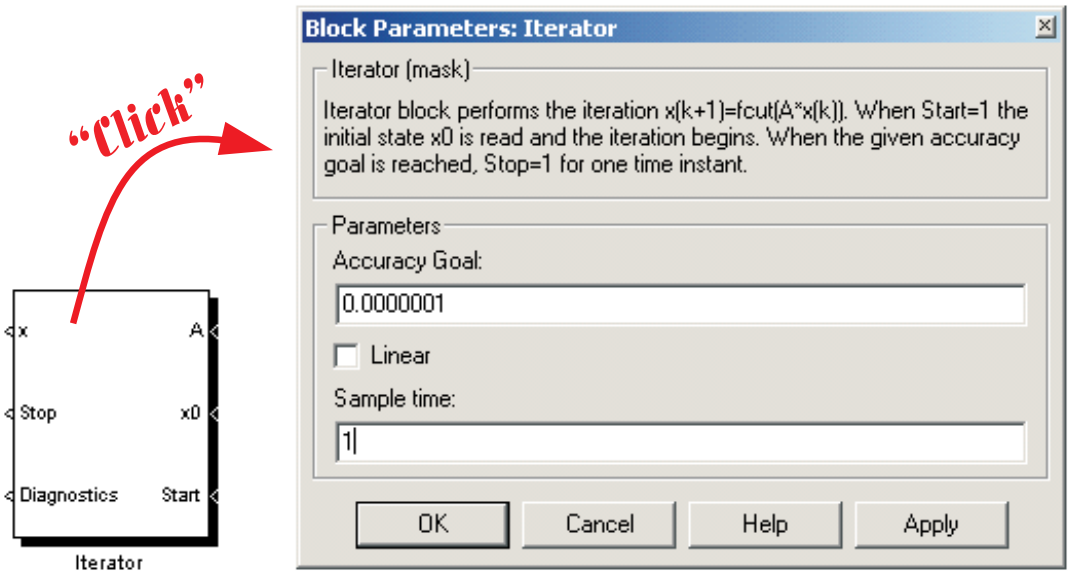

Figure 15: Outlook of a single "cognitive block" implemented in `Simulink`

4 CONCLUSIONS

One of the guiding principles in this presentation was to keep the architectures simple, and introduce only minimum amount of extra structural sophistication. The nice thing is that the added structural enhancements can readily be integrated in the assumed framework (1).

Two different examples of hierarchic structuring needed in cognitive systems were presented, and it was noticed that there is always a compromise to be made between clumsy but simple and lightweight but complicated solutions. Typically, the simple structures can be adapted automatically whereas the sophisticated structures need to be explicitly wired by someone. It seems that no general guidelines could be reached, one cannot say whether some of the approaches would always be better: In the parallel manipulation of the perceptions the clumsy approach seemed to work nice, whereas when managing sequential processes, the complex, decomposed approach was preferred, giving the most maintainable models.

There is a difficult compromise to be made here: If the emphasis is on the low level, structurally simple structures are produced, with possibility of perhaps being self-adaptive, but being difficult to master conceptually; if the emphasis is on the high level, on the other hand, the tailor-made structures become more rigid, and need to be hand-wired by someone. It seems that in truly complex systems with many hierarchic levels, the holistic and reductionistic approaches need to *alternate* from level to level.

ACKNOWLEDGEMENT

I am grateful to Mr. Olli Haavisto who implemented the "cognitive simulation environment" in Simulink during the Summer 2002.

REFERENCES

[1] Anderson, J.R.: *The Architecture of Cognition.* Harvard University Press, Cambridge, Massachusetts, 1983.

[2] Földiák, P.: Forming sparse representations by local anti-Hebbian learning. *Biological Cybernetics,* Vol. 64, 1990, pp. 165–170.

[3] Hoyer, P.O. and Hyvärinen, A.: A Multi-Layer Sparse Coding Network Learns Contour Coding from Natural Images. *Vision Research,* Vol. 42, No. 12, pp. 1593–1605, 2002.

[4] Hyötyniemi, H.: *Constructing Non-Orthogonal Feature bases.* Proceedings of the International Conference on Neural Networks (ICNN'96), June 3–6, 1996, Washington DC, pp. 1759–1764.

[5] Hyötyniemi, H.: On Mental Images and 'Computational Semantics'. In *Proceedings of the 8th Finnish Artificial Intelligence Conference STeP'98* (eds. Koikkalainen, P. and Puuronen, S.), Finnish Artificial Intelligence Society, Helsinki, Finland, 1998, pp. 199–208.

[6] Hyötyniemi, H.: *Systems Theory and Theory of Mind — Towards a Synthesis?* Finnish Artificial Intelligence Days STeP 2000, August 28–30, 2000, Vol. 3, pp. 123–131.

[7] Hyötyniemi, H.: *Complex Systems — Searching for Gold.* Arpakannus 2/2002, special issue on Complex Systems, pp. 29–34.

[8] Hyötyniemi, H.: *Studies on Emergence and Cognition — Parts 1 & 2.* Finnish Artificial Intelligence Conference (STeP'02), December 16–17, 2002, Oulu, Finland.

[9] Hyötyniemi, H.: *Life-Like Control.* Submitted to STeP'02 to be organized in Oulu, Finland, December 2002.

[10] Kant, I.: *Critique of pure reason,* 1781. A simplified version of this philosophy is given in *Prolegomena: To Any Future Metaphysics That Can Qualify as a Science.*

[11] Kellogg, R.T.: *Cognitive Psychology.* SAGE Publications, London, 1995.

[12] Kohonen, T.: *Self-Organizing Maps.* Springer–Verlag, Heidelberg, 1995.

[13] Newell, A.: *Unified Theories of Cognition.* Harvard University Press, Cambridge, Massachusetts, 1990.

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

[15] Simon, H.A.: *Sciences of the Artificial.* MIT Press, 1969 (first edition).

[16] Vernon, D.: *Machine Vision.* Prentice-Hall, 1991.