

# Real-Time Processing of Structure and Its Anticipation

Tadashi Ae, Hiroyuki Araki, Saku Hiwatashi, and Ken-ichi Katakawa

Electrical Engineering, Faculty of Engineering, Hiroshima University

1-4-1 Kagamiyama, Higashi-Hiroshima, Hiroshima, 739 Japan

Fax: +81-824-22-7195 Email: ae@aial.hiroshima-u.ac.jp

## ABSTRACT

A two-level processing scheme for real-time image understanding is proposed, where an example-based ( or case-based ) reasoning in neural AI systems is introduced. The system has two levels; Component Level and Structure Level. At the component level, an elementary pattern recognition is performed as in the conventional pattern recognition, while the syntax pattern recognition is done at the structure level. Both levels are essentially time-consuming ( theoretically, NP-complete each). The pattern recognition assisted by syntax recognition reduces the total complexity of processes, and the system can perform a real-time image understanding, when the VLSI chips are introduced. As a result, we show a reasonable real-time image understanding scheme by introducing a neural pattern recognition at the component level and a case-based AI technique at the structure level.

**Keywords:** Neural Network, Artificial Intelligence, Genetic Programming, Structured Data, Real-Time Processing

## 1 INTRODUCTION

A two-level processing scheme for real-time image understanding is proposed, where an example-based ( or case-based ) reasoning in neural AI systems is introduced. The system has two levels; Component Level and Structure Level. At the component level, an elementary pattern recognition is performed as in the conventional pattern recognition, while the syntax pattern recognition is done at the structure level. Both levels are essentially time-consuming ( theoretically, NP-complete each ).

In this paper, a new architecture SBA, which is derived from human brain model, is proposed for real-time data processing[13]. The scheme of data-flow in SBA seems to be similar to that of a conventional parallel machine, but the architecture is different in realizing directly two types of memory, i.e., STM ( Short Term Memory ) and LTM ( Long Term Memory ). The STM and the LTM in SBA work mainly for adaptiveness and stability, respectively. Learning is performed on these two memories, since the SBA supports both types of learning; one for STM and the other for LTM ( with assistance of STM ), and therefore, the SBA is naturally a kind of memory-based architectures.

The computer includes both software and hardware, but the brainware is derived from human brain model[2]. In this sense, the SBA can be regarded as an extension from mixed systems of neural networks, AI systems and computers. Moreover, the device technology

is also important for fabrication, and nanometer IC technology will be expected. The SBA is a general system, but we focus on real-time image processing in this paper. The pattern recognition assisted by syntax recognition reduces the total complexity of processes, and the system can perform a real-time image understanding, when the VLSI chips are introduced. As a result, we show a reasonable real-time image understanding scheme by introducing a neural pattern recognition at the component level and a case-based AI technique at the structure level.

## 2 Two-Level Processing Scheme

The global dataflow of the special-purpose brainware architecture ( in short, SBA ) is shown as in Fig.1, where LTM and STM are the feed-forward network and the feedback network, respectively. The dataflow shows only a schematic representation of data transmission, but the LTM is important in the SBA, because the system without feedback coincides with a neural network without feedback ( e.g., MLP: Multilayer Perceptron net ). By contrast, the feedback system without LTM is the same as the recurrent network. The feedback system with both LTM and STM is essentially important in the SBA. The configuration in Fig.1 may seem to be the same as the dataflow in a conventional parallel machine, but an essential difference is that the SBA works by not programming but learning. The learning-based system is obtained by a memory-based architecture. Fig.1 shows a memory-based architecture, where the memory in the feed-forward part and the memory in the feedback part are the STM (short-term memory) and the LTM (long-term memory), respectively. In general, the LTM and the STM each are given as follows;

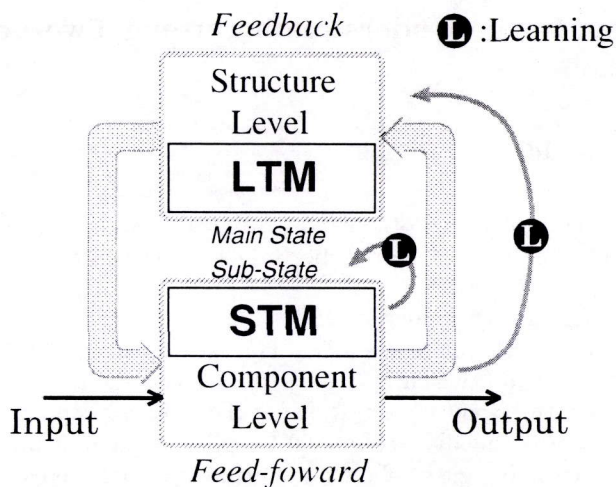
LTM: Active Ensemble Memory + Passive Ensemble Memory.

STM: Active Ensemble Memory + Passive Ensemble Memory.

If active is more powerful than passive, it is called active, and vice versa. "Ensemble" means a kind of addressable-in-parallel memory. Learning in STM is made within the feed-forward part, but learning in LTM is made within the loop including both feed-forward and feedback parts. Both learning are inductive, and learning in STM is the same as the conventional learning in the neural network. We do not restrict the type of neural networks in the feed-forward part, but focus mainly on MLP and Kohonen net, because these two are popular and features each are also well known. As learning, the BP ( error-back-propagation ) algorithm and the LVQ ( leaning vector quantization ) algorithm are supposed to be used for MLP and Kohonen net, respectively. The feedback part is not so easily determined, because inductive leaning in the loop structure provides a difficult problem. A candidate is the recurrent network with ( an extended ) BP algorithm, but the application is restricted because it is the case without LTM. Then, we need to develop a new architecture. In the memory-based architecture in Fig.1, the memory in the feed-forward (i.e., STM) represents the sub-state, while the memory in the feedback part (i.e., LTM) does the main state. In the case of image understanding, the feed-forward part and the feedback part work for the component level and the structure level, respectively, and therefore, the STM and the LTM play roles of the component (i.e., pattern recognition) and the structure (i.e., syntax understanding), respectively.

The learning algorithm in the SBA is shown as in Fig.2, where the inductive learning for LTM should be a constructive inductive learning, and the structure must be obtained by learning procedure. In other words, the structure of automatic truth maintenance or the structure of minimal systematic representation must be constructed by learning. The rule-based notation or the logical ( predicate or propositional ) notation is used for the structure construction of automatic truth maintenance. The machine ( or automaton ) notation is also used for the structure construction of minimal systematic representation. It should be noted that this learning procedure must be applied not only for symbols but also for numerical values. The notation depends on the application, but each notation must be extended to numerical case from symbolic case.

In the SBA adaptiveness ( corresponding to Cerebellum in brain ) is understood to be a parameter tuning, i.e., a small derivation in learning for a single data. On the other hand, stability ( corresponding to Basal Ganglia in brain ) is understood to be a convergence, i.e., not diverse in learning for a set of data ( a large number of data ). At a tuning for a single data, the loop for learning is easily tunable, and therefore, this process is realized mainly by STM. For a large number of data, however, we need a delicate loop-control for convergence, and realize it by LTM and also STM ( as assistance ). The algorithm in Fig.2 is given by satisfying these factors.



STM: Short-Term Memory  
 LTM: Long-Term Memory

Figure 1: Behavior of SBA

## Two-level Learning

```
begin
  LF : Logical Formula is given. {on LTM
  Local Induction Step :
    {Neural Net Learning on STM}
  Global Induction Step :
    Consistency Check of LF
    Simplification of LF {Most Probable
    if Result OK then terminat
    else Modify LF
      {Constructive Induction Step}
    goto the beginning
end.
```

Figure 2: Two-Level Learning Algorithm

### 3 Real-Time Image Understanding using Two-Level Processing Scheme

#### 3.1 Fundamental Idea

We introduce the recognition of a hand-written phrase, since it is one of real-time pattern recognition. Let a sample phrase be *Brainware Architecture*.

First Stage: Character Recognition.

For *Brainware*, each character, that is, 'B', 'r', 'a', 'i', 'n', 'w', 'a', 'r' or 'e' must be recognized, and for *Architecture* the similar process is applied. For such a process we can use any neural network which is fit in pattern recognition. Considering hardware realization, however, the Kohonen net is one of the most appropriate neural networks, and a chip realization is already performed[9,10] and its system is already evaluated[4]. These case is restricted to digit symbols due to chip size, and therefore, an extension is now being prepared. It should be noted that this character recognition is performed only in the feed-forward part.

Second Stage: Word Recognition.

In this stage, leaning of grammar or machine is required, but a simple grammar or a deterministic finite automaton is enough powerful for word recognition ( e.g., Lex in Unix ).

Third Stage: Sentence Recognition.

The general case becomes complicated as in the recognition of a natural language. In our case, however, the third stage is included in the second stage because we focus on a simple case such as a phrase ( e.g., *Brainware Architecture* ). The hand-written phrase such as a signature does not have the complicated structure. Instead, the character is not so easy to be recognized since the sequence of hand-written characters is not easily separable into

characters. This is a reason why we use the neural network for its recognition. Moreover, the real-time processing is indispensable for recognition of a large number of signatures. We summarize that the first stage corresponds to the component level and that the second and the third stages correspond to the structure level for the two-level processing scheme.

### 3.2 Two-Level Processing Architecture

We have shown the behavioral scheme of two-level processing as in Fig.1. Here we show its architecture as in Fig.3. This is a system architecture which is naturally derived from Fig.1. As a two-level architecture the ART network is already known as in Fig.4. Our architecture is regarded as an extension from the ART network and is more general, since it can be used to various applications.

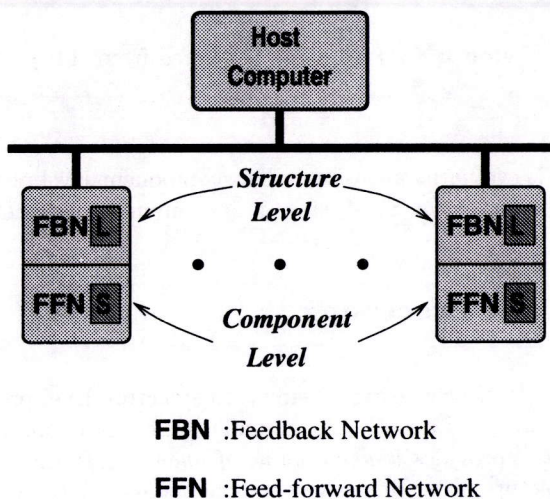


Figure 3: Global Architecture for Special-purpose Brainware

## 4 Component-Level Processing

We have introduced the Kohonen network for the component-level processing [4]. More precisely, we use the Kohonen's LVQ network, where the behavior of LVQ algorithm is shown as in Fig.5. The fundamental procedure of LVQ algorithm is shown as in Fig.6, where the parallel processing is easily introduced. As a result, we have obtained a scheme for hardware chip as in Fig.7, where a chip includes 12 neurons ( ON:Objective Neuron in Fig.7 ) and the input size of each neuron is 8x4 bit pattern. The demonstration system consists of four chips, namely, 48 neurons. It should be noted that a neuron in LVQ network corresponds to a category ( or a class ), and therefore, that 48 neurons are enough powerful for pattern recognition. This neuron corresponds to the neuron of hidden layer in the MLP network. A sample using this hardware chip is shown as in Fig.8,

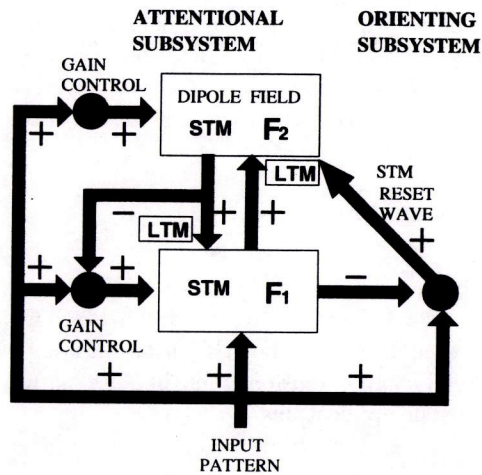


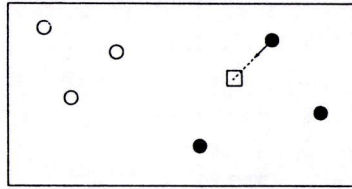
Figure 4: ART Network ( referred from [14] )

and the performance is evaluated as in Fig.9. The component-level processing is regarded fundamentally as behavior at the STM, since it is a tuning of parameters.

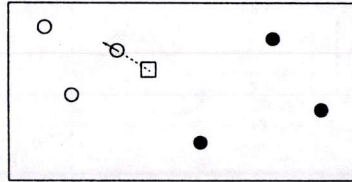
## 5 Structure-Level Processing

The feature of two-level processing scheme is in structure-level processing, where the syntactic behavior is analyzed and the image understanding is done by syntactic way. ( A operational semantics provides *understanding of images*. ) In the case of image object the construction of picture using primitive components and/or the motion of consecutive pictures correspond to this level processing.

In this paper we focus on the motion of consecutive pictures, i.e., the moving pictures as in Fig.10, where a picture consists of primitive patterns. After component-level processing the primitive patterns are represented by  $0, 1, 2, \dots, p-1$ , where  $p$  is the number of primitive patterns. For simplicity we explain the case of  $p=2$ , where the primitive patterns are given by  $0, 1$ . An example of pictures after component-level processing is shown as in Fig.11, which is represented by a bit pattern. Assume that the behavior from a picture at time  $t$  to a picture at time  $t+1$  is represented as in Fig.12. This provides a logical formula or a bit matrix as in Fig.13, where the size of matrix is  $n^2$  for  $n$ =number of logical variables. The problem of obtaining the minimum representation of logical formulae for two consecutive pictures is easily proved to be NP-complete. Therefore we introduce an approximative way using a neural network, i.e., Hopfield network[5]. The Hopfield network is known to provide an approximate algorithm for a class of combinatorial problems. The main procedure of structure-level processing includes a combinatorial problem described as the above. The problem includes an essential problem as follows; For instance, let  $\{x_i \rightarrow \bar{x}_j, x_j \rightarrow \bar{x}_i\}$  ( where  $x_i$  and  $x_j$  are logical variables ) be a local condition. In this



If  $C_i = C_j, m_i(t+1) = m_i(t) + a(t)[x(t) - m_i(t)]$ .



If  $C_i \neq C_j, m_i(t+1) = m_i(t) - a(t)[x(t) - m_i(t)]$ .

- (○) Reference Vector
- (□) Input Vector

Figure 5: Kohonen LVQ Algorithm

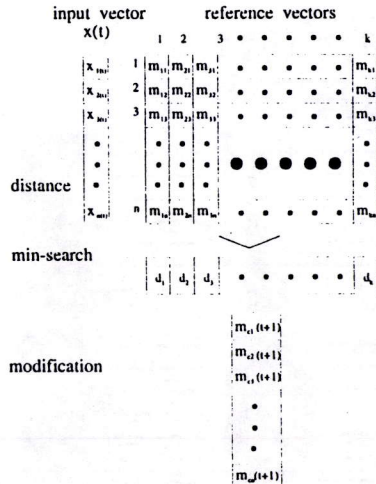
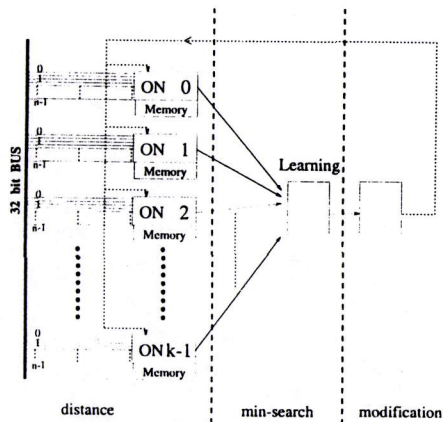


Figure 6: Procedure of LVQ Algorithm



ON : Objective Neuron  
 Memory :Reference(Real value)

Figure 7: Hardware Realization of LVQ Algorithm

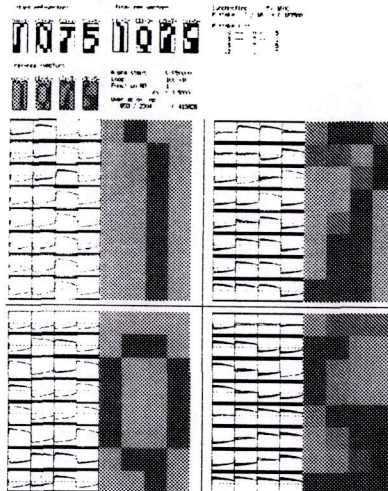


Figure 8: Real-Time Learning using Prototype Chip; An Example



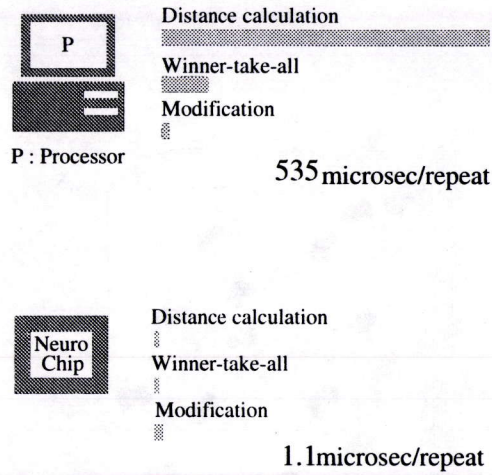


Figure 9: Performance Comparison

case we have no solution such that both  $x_i$  and  $x_j$  become “true”. This means that the consistency on the set of logical variables decides also the local condition. For the general case we have more relaxed conditions, but we need to develop how to obtain the solution. The problem is NP-complete if we need the best solution[8], and therefore, approximated solving ways must be introduced. We proposed a way of using Hopfield network for this problem, and show a good result[8]. A behavior using Hopfield network is shown as in Fig.14. This is a problem to seek a compact knowledge form under holding consistency of a knowledge database. The LTM plays a main role of structure-level processing, although the STM is also used subsidiarily. The two-level processing scheme is realized by an architecture as in Fig.3.

## 6 Anticipation

The n-dimensional data  $\mathbf{X} = (x_1, x_2, \dots, x_n)$  is represented as in  $\mathbf{X} = (X_1, X_2, \dots, X_k)$ , in two-level processing, where  $X_i (i = 1, 2, \dots, k)$  is a subset of dimensions. If  $X_i \cap X_j$  is empty for  $i \neq j$ , i.e., disjoint to each other,  $\mathbf{X}$  is decomposed into a set of blocks, where the total of all blocks is equal to  $n$ .

The data is a spatially-structured data, since the data is composed of blocks. However, the structured data on time-axis is represented as in Fig.15, where a trajectory of data is shown from  $\mathbf{X}_i$  to  $\mathbf{X}_j$ .

For the data on time-axis, the quantization on time axis is first introduced as in Fig.16, and the fundamental functions are obtained on the spatial structured data. Fig.17 shows a spatial structure, where the concept of window is introduced for restricting the domain of spatial functions. The spatial function plays a role of classification of spaces ( in fact, the window-spaces ).

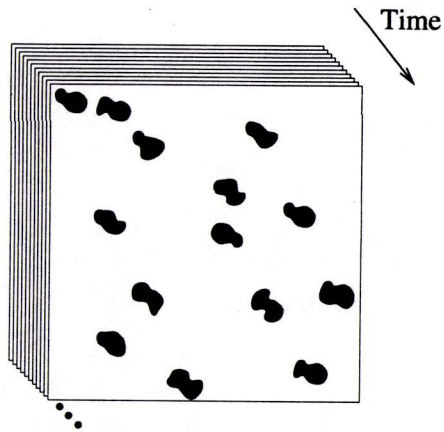


Figure 10: Motion of Consecutive Pictures

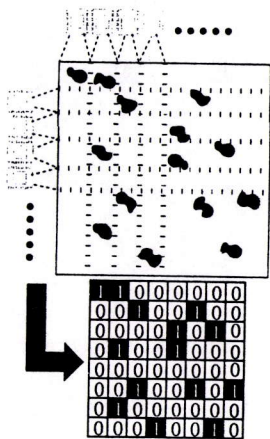


Figure 11: Bit Representation of Moving Pictures

X <sub>1</sub>	X <sub>3</sub>	X <sub>5</sub>
X <sub>2</sub>	X <sub>4</sub>	X <sub>6</sub>

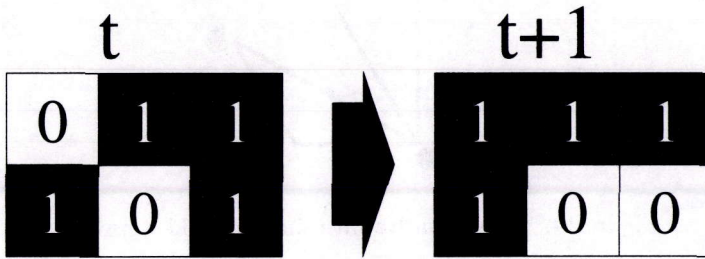


Figure 12: Behavior from Picture at time  $t$  to Picture at time  $t+1$

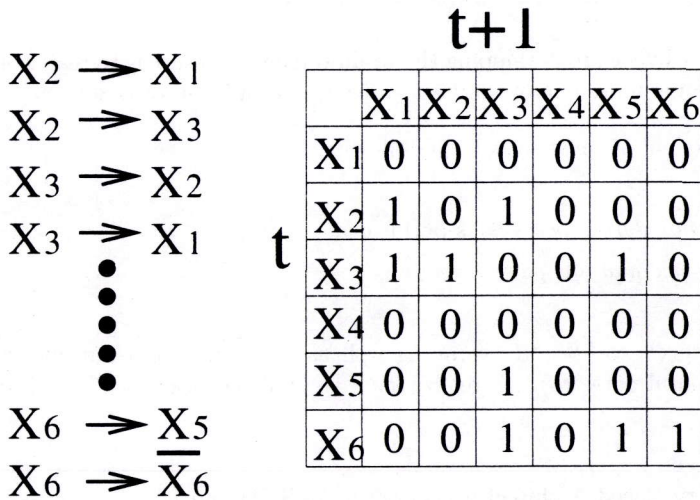


Figure 13: Bit Matrix of Single Clause (Column is time at  $t+1$ , and Row is time at  $t$ )

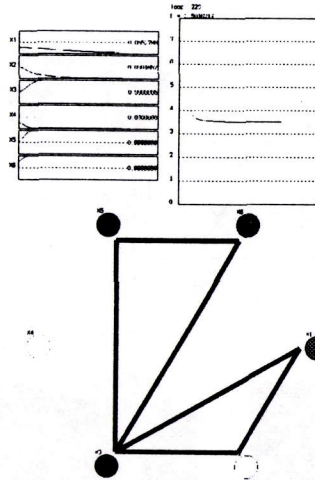


Figure 14: Simulation Result using Hopfield Network

Therefore, we can focus on the behavior of window-spaces as in Fig.18, and realize a real-time processing system with a reasonable size of hardware. We can introduce the GP (Genetic Programming) technique for real-time anticipation on time-axis.

## 7 Conclusions

The reason why we are developing the brainware architecture is to realize a human-like brain which seems to be one of the best realization of a brain of robot.

The brain of robot must satisfy the following;

- 1) *Real-time data processing* is performed.
- 2) *Small-size and low-power* are realized.

We also expect *real-time learning* as well as real-time input-output response (i.e., *recall* in the neural network). To realize such an artificial brain, we have introduced the following;

- a) Memory-Based Architecture consisting of STM and LTM.
- b) Chip realization with (possible) advanced technology.

Several types of memory-based architectures are already proposed, but we focus on the memory-based AI architecture[11,12]. In this paper, we specify more clearly the

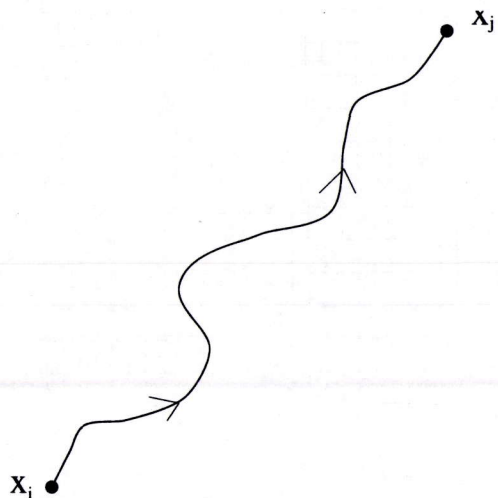


Figure 15: A Trajectory from  $x_i$  to  $x_j$

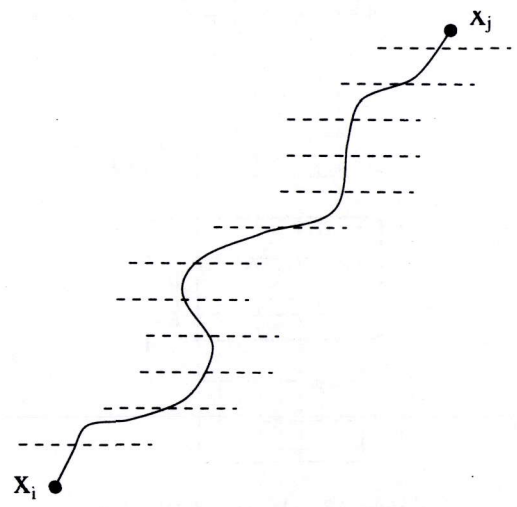


Figure 16: Quantization of Trajectory

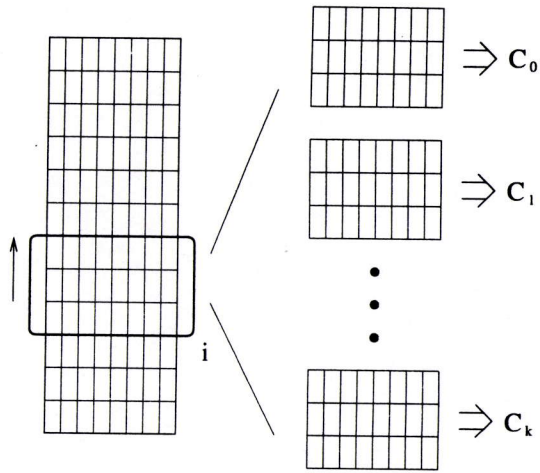


Figure 17: Table Look-up for Window Function

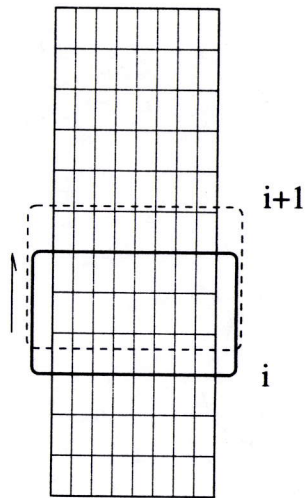


Figure 18: Moving of Window

memory-based AI architecture with STM and LTM. As hardware realization we have several candidates on this architecture, and it depends on the application.

About chip technology we are now using the conventional one (e.g.,  $0.5\mu\text{m}$  design-rule in [10]). In future, however, an artificial brain for robot will be realizable using more advanced nanometer device technologies.

## REFERENCES

- [1] M.Ito; Expecting for Brain Research in the 21st Century, Extended Abstract of Brainware Workshop, Tsukuba, FED-148, pp.3-5 (1996, in Japanese).
- [2] G.Matsumoto, Y.Shigematsu, M.Ichikawa; Brain-computing, i.b.i.d., pp.31-36 (1996, in Japanese).
- [3] R.Agrawal; Database Mining: A Performance Perspective, IEEE Transaction on Knowledge and Data Engineering, Vol.5, No.6, pp.914-925 (1993).
- [4] H.Araki, H.Fukumoto, T.Ae; Image Processing using Simplified Kohonen Network, Proceedings SPIE, San Jose, Vol.2661 (1996).
- [5] J.J.Hopfield; Neurons with Graded Response Have Collective Computational Properties like Those of Two-State Neurons, Proceedings National Academy of Science, USA81, pp.3088-3092 (1984).
- [6] K.Aihara; Chaos and Brainware, Extended Abstract of Brainware Workshop, Tsukuba, FED-148, pp.8-12 (1996).
- [7] K.Murota, X. Wu, T.Ae; High-Speed Processing for Knowledge Representation from Decision Diagram, JSAI Technical Report, SIG-PPAI-9502-2 (1995, in Japanese).
- [8] H.Fukumoto, S.Hiwatashi, X.Wu, T.Ae; Knowledge Discovery by Neural Reasoning, JSAI Technical Report, SIG-PPAI-9503-6 (1996, in Japanese).
- [9] T.Ae, T.Toyosaki, H.Fukumoto, K.Sakai; ONBAM: An Objective-Neuron-Based Active Memory, Proceedings 1st ICA3PP, Brisbane, Vol.1, pp.231-234 (1995).
- [10] T.Toyosaki, T.Ae; A Neuro-Chip for Kohonen's LVQ Algorithm, Proceedings 6th ISIC, Singapore, pp.107-111 (1995).
- [11] T.Ae; New Functional Machine, (New) Information Processing Handbook (JIPS Ed.), 3.8.7., pp.455-459, Ohm Co. Ltd. (1995, in Japanese).
- [12] M.Kawada, X.Wu, T.Ae; A Construction of Neural-Net Based AI Systems, Proceedings 1st ICECCS, Ft. Lauderdale, pp.424-427 (1995).
- [13] T.Ae, H.Fukumoto, S.Hiwatashi; Special-Purpose Brainware Architecture for Data Processing ( To appear )
- [14] G.A.Carpenter and S.Grossberg; The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network, IEEE Computer, Vol.21, No.3, pp.77-88 (1988).