Fifth International Workshop on Computational Intelligence & Applications
IEEE SMC Hiroshima Chapter, Hiroshima University, Japan, November 10, 11 & 12, 2009

A Morphological Associative Memory Employing A Stored Pattern Independent Kernel Image and Its Hardware Model

Hidetaka Harada, Takashi Saeki†, and Tsutomu Miki
Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology
Kitakyushu 808-0196, Japan
harada-hidetaka@edu.brain.kyutech.ac.jp

Abstract—**An associative memory provides a convenient way for pattern retrieval and restoration, which has an important role for handling data distorted with noise. As an effective associative memory, we paid attention to a morphological associative memory (MAM) proposed by Ritter. The model is superior to ordinary associative memory models in terms of calculation amount, memory capacity, and perfect recall rate. However, in general, the kernel design becomes difficult as the stored pattern increases because the kernel uses a part of each stored pattern. In this paper, we propose a stored pattern independent kernel design method for the MAM and design the MAM employing the proposed kernel design with a standard digital manner in parallel architecture for acceleration. We confirm the validity of the proposed kernel design method by auto- and hetero-association experiments and investigate the efficiency of the hardware acceleration. A high-speed operation (more than 150 times in comparison with software execution) is achieved in the custom hardware. The proposed model works as an intelligent pre-processor for the Brain-Inspired Systems (Brain-IS) working in real world.**

I. INTRODUCTION

Almost all of the data which human beings treat in their daily life are incomplete information which lacks partially and/or is distorted with noise. Nevertheless human beings can retrieve and complement incompleteness of such information by using fruitful data obtained through experiences and take appropriate actions to the situations. The function is provided by associative memories and is necessary for the Brain-Inspired Systems (Brain-IS) which are intelligent system working in real world like human beings. Our aim is to realize the function by using an effective associative memory and endow the Brain-IS with an intelligent pre-processing.

From the first half of 70's, many associative memory models have been proposed [1-4]. In most of associative memory models, the function is attractive, but it has drawbacks; small memory capacity, low convergence to correct pattern, etc. On the other hand, Ritter proposed a morphological associative memory (MAM) in 1998 [5]. The MAM is based on the morphological neural networks [6] and executes storing and recalling with mathematical morphology. The model is superior to ordinary associative memory models such as Hopfield network [7] in terms of calculation amount, memory capacity, and the perfect recall rate.

The MAM stores and recalls the patterns by using the memory matrix. The different types of memory matrices "*M*" and "*W*" are introduced for association. Memory matrices "*M*" and "*W*" are effective for an erosive noise or a dilative noise, respectively. However, memory matrices "*M*", "*W*" have no power for the erosive and dilative noises, respectively. Therefore, one of the memory matrices can not cope with the noise which includes both erosive and dilative noise simultaneously. Ritter proposed the MAM utilizing properties of memory matrices "*M*" and "*W*" which execute two-stage recall process. The MAM uses a kernel image as an intermediate image. The kernel image is a part of each stored pattern and with no overlaps for other stored patterns. The Ritter's model has a problem that the determination of the kernel image becomes difficult as the stored patterns increases because the kernel image is created using a part of the stored pattern. In order to overcome the problem, Hattori et al. proposed a fast method to decide the kernel image [8], Sussner proposed the MAMs based on variations of the kernel and used the dual kernel methods [9], and Ida et al. proposed a method that uses stored patterns with redundancy bits [10]. However these models have some problems: the model using the kernel image basically has fatal problem that the perfect recall can not achieve for pattern including a corrupted kernel image; an extra post-processing is necessary. As a MAM using no kernel image, we also have proposed the block splitting type MAM, which introduced the block splitting method and a majority logic approach to avoid spreading a noise over an image in the recall process and obtain a plausible recall pattern respectively [11]. However, the model also has a problem that the perfect recall rate is inferior to the MAM using the kernel image.

In this paper, we propose a new MAM using a stored pattern independent kernel image. In the proposed model, the stored pattern independent kernel design method makes an assignment of the kernel image easy. Moreover, the introduction of the stored pattern independent kernel solves the fatal problem that the Ritter's MAM can not recall the

† Dr. Saeki works now on Hitachi, Ltd.

correct pattern when the kernel is corrupted by noise. In order to confirm the validity of the proposed model, we evaluate the perfect recall rate by auto- and hetero-association experiments. The pre-processing demands a high-speed operation and a low computational cost for working without any harmful effect to the main processing. We also discuss the implementation of the proposed method in a custom hardware. The proposed model works as an intelligent pre-processing for the Brain-IS.

II. MORPHOLOGICAL ASSOCIATIVE MEMORY : MAM

A. *Characteristics of memory matrices "M" and "W" in the MAM*

Ritter proposed a morphological associative memory (MAM) by using memory matrix [5]. The different types of memory matrices "*M*" and "*W*" were introduced. The MAM uses the memory matrix "*M*" or "*W*" for association. The information of the stored patterns is stored into the memory matrix "*M*" or "*W*". Here, let assume p pattern pairs $(X^1,Y^1),\cdots,(X^p,Y^p)$ as the stored patterns. Here $X^s=(x_1^s,\cdots,x_n^s)$, $Y^s=(y_1^s,\cdots,y_m^s)$, p is the total number of the pattern pairs. The memory matrices "*M*" and "*W*" are given as:

$$m_{ij}=\bigvee_{s=1}^{p}\left(y_i^s-x_j^s\right)=\left(y_i^1-x_j^1\right)\vee\left(y_i^2-x_j^2\right)\vee\cdots\vee\left(y_i^p-x_j^p\right), \tag{1}$$

$$w_{ij}=\bigwedge_{s=1}^{p}\left(y_i^s-x_j^s\right)=\left(y_i^1-x_j^1\right)\wedge\left(y_i^2-x_j^2\right)\wedge\cdots\wedge\left(y_i^p-x_j^p\right), \tag{2}$$

where m_{ij} and w_{ij} are (*i*, *j*)*-th* element of memory matrices "*M*" and "*W*", respectively. The symbols $\wedge$ and $\vee$ denote the operations of minimum and maximum, respectively. The recalled pattern for the input pattern X^s is obtained by calculating Eq.(3) or Eq.(4).

$$y_i^s=\bigwedge_{j=1}^{n}\left(m_{ij}+x_j^s\right)\qquad i=1,\cdots,m, \tag{3}$$

or

$$y_i^s=\bigvee_{j=1}^{n}\left(w_{ij}+x_j^s\right)\qquad i=1,\cdots,m. \tag{4}$$

Memory matrices "*M*" and "*W*" are effective for an erosive noise or a dilative noise, respectively. However, the memory matrices "*M*" and "*W*" can not reduce the erosive and the dilative noises, respectively. Therefore, one of the memory matrices can not cope with the noise which includes both erosive and dilative noise simultaneously.

B. *The MAM using the kernel image*

In order to handle the noise which includes both erosive and dilative noise simultaneously, Ritter proposed the MAM adopts a two-stage recall process using matrices "*M*" and "*W*" in stages. In the recall process, the MAM uses a kernel image as an intermediate image.

The kernel image Z^s works as the index for recalling the stored pattern Y^s and consists of partial units of the stored pattern X^s. The MAM using the kernel image stores pattern pair (X^s,Y^s) with the kernel image Z^s. Here the memory matrices "*M*" and "*W*" are given as;

$$m_{ij}=\bigvee_{s=1}^{p}\left(z_i^s-z_j^s\right), \tag{5}$$

$$w_{ij}=\bigwedge_{s=1}^{p}\left(y_i^s-z_j^s\right). \tag{6}$$

The kernel image Z^s is the pattern described with a part of the input pattern X^s and it has a one-to-one correspondence the input pattern X^s. Therefore, the output pattern can be associated from the input pattern X^s by using Eqs. (5) and (6). When the input pattern X^s is fed into the MAM, the outputs can be obtained by two-stage recall process given by follows;

$$z_i^s=\bigwedge_{j=1}^{n}\left(m_{ij}+x_j^s\right)\qquad i=1,\cdots,n, \tag{7}$$

$$y_i^s=\bigvee_{j=1}^{n}\left(w_{ij}+z_j^s\right)\qquad i=1,\cdots,m. \tag{8}$$

Fig.1 shows the tow-stage recall process in Ritter's MAM using the kernel image.

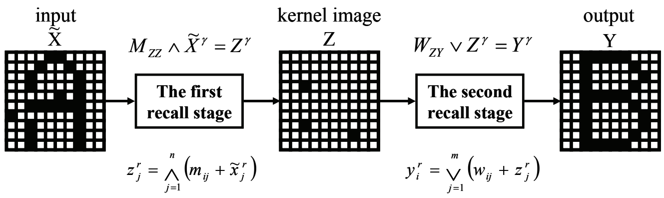

Fig.1: Recall process of Ritter's MAM using the kernel image.

C. *Ritter's method of constructing the kernel image*

In Ritter's kernel image constructing method, the kernel images are determined to satisfy conditions given by the Eq. (9). The kernel image Z^ξ is a part of the stored pattern X^ξ and is selected so as not to overlap with other stored patterns.

$$Z^\xi\le X^\xi,\ \ Z^\xi\wedge X^\gamma=0\qquad \gamma\ne\xi. \tag{9}$$

If the kernel image can not be determined by calculating Eq. (9), the kernel image must be obtained by trial and error to satisfy the condition described by Eq. (10).

$$Y^\xi=W_{ZY}\vee(M_{ZZ}\wedge X^\xi). \tag{10}$$

III. MAM Using The Stored Pattern Independent Kernel Image

In Ritter's kernel design method, the kernel image is created using a part of stored pattern therefore the determination of the kernel image becomes difficult as the stored patterns increases. Moreover, the perfect recall can not be achieved if a part of the kernel image includes noise. To solve the problems, we propose a new MAM using a stored pattern independent kernel image.

In the proposed method, the kernel image consists of several bits which are equivalent to the number of stored patterns as shown in Fig.2. Only one element of the kernel image is '1', other elements are '0'. Each kernel image is set so that the kernel image doesn't overlap the other kernel images. Fig.2 shows an example of the kernel image when the maximum number of stored patterns is five.

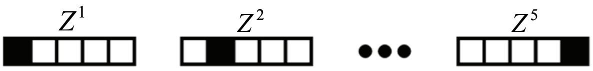

Fig.2: Example of the kernel images used in the proposed method.

The proposed model adopts the two-stage recall process as same as Ritter's MAM. However, the correct kernel image can not recall perfectly if the input image includes erosive noise because we assign a stored pattern independent image to the kernel image and use only "*M*" for recalling the kernel image.

In order to overcome the problem, we introduce the block splitting scheme in the first recall stage and the majority logic approach, which is a similar scheme which was introduced in the MAM employing no kernel image [11]. Here we apply the approach to get the plausible kernel image. By applying the block splitting manner to an input image, plural sub images are created from an input image. Here the same kernel is assigned to all generated sub images. After that, the correct kernel image is obtained with a high probability by applying the majority logic approach to the plural kernels because a MAM has a feature that a noiseless image recalls the correct image. Finally, the correct recall pattern is obtained from the plausible kernel image with a high probability.

The recalling of the proposed MAM processes through the following steps;

step 1. An input pattern is divided evenly into sub blocks,
step 2. the first recall is executed every sub block independently,
step 3. output patterns of all sub blocks are summed up,
step 4. the kernel image can be determined by a majority logic for the kernel pattern obtained in step3,
step 5. finally the output pattern is recalled using the kernel image in the second recalling stage.

The output of the *j-th* unit of *sb-th* sub block in the first recalls is given by:

$$\tilde{z}_i^{r,sb} = \bigwedge_{j=1}^{k} \left(m_{ij}^{sb} + \tilde{x}_j^{r,sb} \right), \tag{11}$$

where k represents the number of total units of each sub block, $\tilde{x}_j^{r,sb}$ is *j-th* unit of *sb-th* sub block in the corrupted pattern $\tilde{x}^r$. Here, in the proposed method, the kernel image is independent of the input pattern. Therefore, the memory matrix "*M*" is defined as Eq. (12) instead of Eq. (5).

$$m_{ij}^{sb} = \bigvee_{r=1}^{p} \left(z_i^{r,sb} - x_j^{r,sb} \right), \tag{12}$$

The summation and the majority logic operation are defined by Eq. (13). The processes are executed after the first recall stages.

$$z_j^r = \begin{cases} 1 & for \quad \sum_{sb=1}^{s} \tilde{z}_j^{r,sb} = \bigvee_{i=1}^{h} \left(\sum_{sb=1}^{s} \tilde{z}_i^{r,sb} \right) \\ 0 & for \quad \sum_{sb=1}^{s} \tilde{z}_j^{r,sb} < \bigvee_{i=1}^{h} \left(\sum_{sb=1}^{s} \tilde{z}_i^{r,sb} \right) \end{cases}, \tag{13}$$

where h represents the number of total units of the kernel image, $\tilde{z}_i^{r,sb}$ is *i-th* unit of the kernel image corresponding to the input pattern $\tilde{x}_j^{r,sb}$ in the *sb-th* sub block, and z_j^r is *j-th* unit of the kernel image corresponding to an input pattern $\tilde{x}^r$, respectively. s represents the number of the block splits. s=1 is the special case which means no block splits.

Fig.3 is the recall process of the proposed MAM. The MAM improves the problem of Ritter's method that design of the kernel image is difficult. Furthermore, the introduction of the stored pattern independent kernel solves the fatal problem that the Ritter's MAM can not recall the correct pattern when the kernel is corrupted by noise.

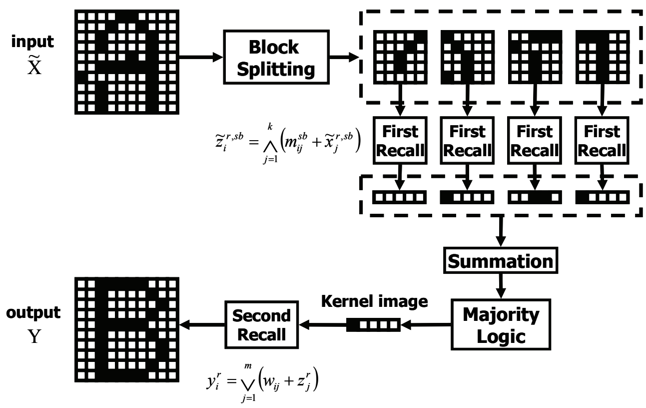

Fig.3: Recall processing of the MAM using the proposed method.

IV. Experimental Results

The performance of the proposed method is investigated through auto- and hetero-association experiments. The perfect recall rate is evaluated by an average of 10,000 trials in the simulations. Here, the perfect recall is defined as to recall the stored pattern which corresponds to noise less input one. The unit of patterns takes '1' or '0'. The '1' represents black and '0' white, and the noise is defined as to change '1' to '0' (or '1' to '0').

A. Auto-association

We investigated the perfect recall rate of the proposed MAM in auto-association experiments. In the experiments, twenty alphabet capital letter patterns are used as shown in Fig.4. Each pattern consists of 10 x 10 = 100 binary units.

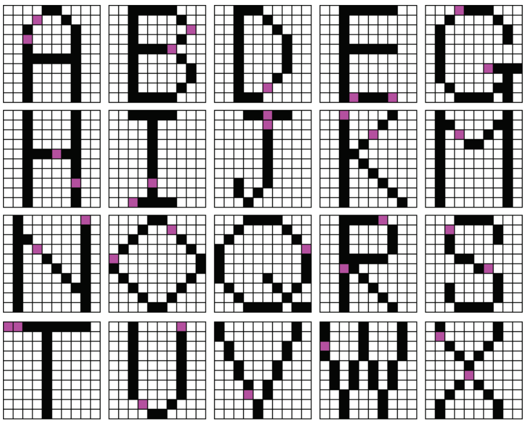

Fig. 4: Stored patterns: twenty alphabet capital letter patterns. ■ denotes the kernel image for Ritter's MAM.

Fig.5 shows the noise tolerance of the proposed MAM and Ritter's MAM for alphabet patterns in auto-association experiments.

We investigated the perfect recall rate with changing the number of block splits. The results show in Fig.5. Fig.5 shows that the perfect recall rate improves as increase of the number of block splits. Here, *sb* represents the number of the block splits. *Sb* = 1 is the special case, which means no block split.

B. Hetero-association

We investigated the perfect recall rate of the proposed MAM in hetero-association experiments. Fig.4 and Fig.6 show the stored patterns used in the experiment. Fig.6 shows twenty alphabet small latter patterns corresponding to patterns illustrated in Fig.4.

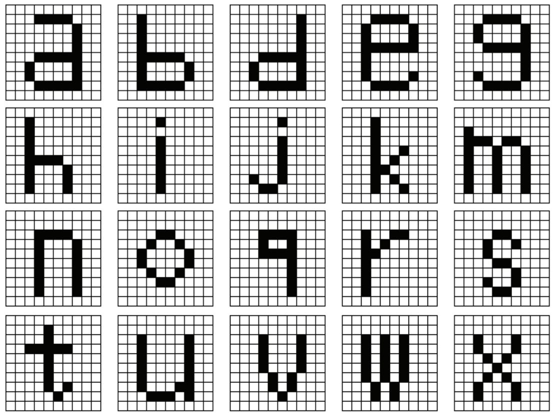

Fig.6: Stored patterns: alphabet small letter (twenty patterns).

Fig.7 shows the noise tolerance of the proposed MAM and Ritter's MAM in the hetero-association experiments for alphabet patterns illustrated in Fig.4 and Fig.6. As well as auto-association experiments, Fig.7 shows that the perfect recall rate improved as increase the number of block splits as the same as the auto-association experiments.

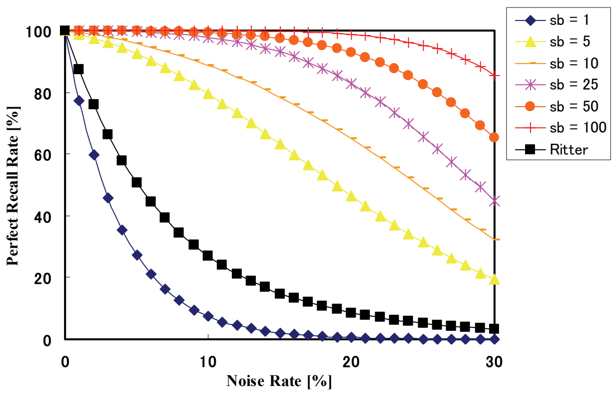

Fig.5: Noise tolerance of the proposed MAM in auto-association for twenty patterns illustrated in Fig.4.

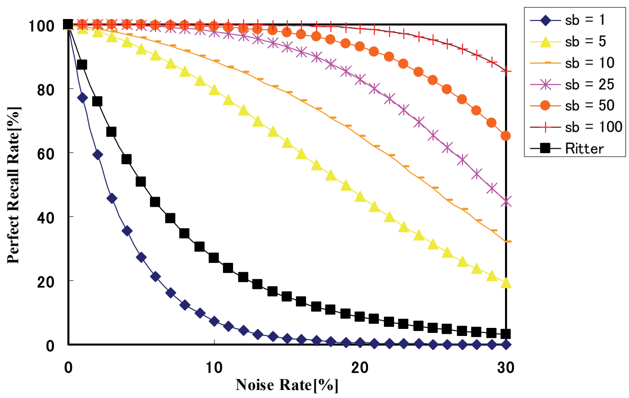

Fig.7: Noise tolerance of the proposed MAM in hetero-association for twenty patterns illustrated in Fig.4 and Fig.6.

V. HARDWARE IMPLEMENTATION OF THE PROPOSED MAM MODEL

The MAM employing the proposed kernel design has parallel architecture in the first and second recall processing. Therefore we try to accelerate the operation by using a dedicated custom hardware.

The first recall of the MAM employing the proposed method, the memory matrix "*M*" is added to input pattern *X*. The output pattern is obtained by taking minimum of it in each sub block as defined in Eq. (11). In the second recall, the memory matrix "*W*" is added to the kernel image *Z*. The final output can be obtained by taking maximum of it as defined in Eq. (8).

We designed the hardware system by using a standard digital scheme in massively parallel manner. Here, the system handles thirty two stored patterns. Each pattern consists of 64 bits. Fig.8 shows the block diagram of the system.

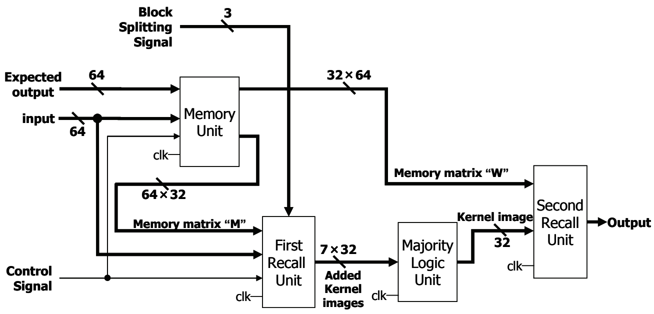

Fig.8: Block diagram of the proposed MAM hardware.

The system consists of a memory unit, a first recall unit, a majority logic unit, and a second recall unit. The memory unit calculates memory matrices "*M*" and "*W*" with input and the corresponding output patterns and stored them.

The first recall unit consists of several sub blocks which generated by block splitting. In each sub block, the kernel image is obtained. After that, the kernel images of all sub blocks are stacked up and fed into the majority logic unit. The majority logic unit determines the final kernel image by applying majority logic to the stacked kernel image. Finally, in the second recall unit, the associated pattern is calculated by using the final kernel image and the matrix "*W*".

Here, Control signal switches the states which are storing and recalling. The block splitting signal assigns the number of block splits.

We investigate the performance of the proposed MAM hardware model by using a logic simulator ModelSim. Table.1 shows the simulation result of the proposed hardware model in comparison with the result using a standard PC. Here a CPU of the PC is Intel Xeon@3.0 GHz.

Table I: Performance of the MAM hardware model employing the proposed method.

Model	Maximum number of stored patterns	Maximum frequency (MHz)	Speed (μsec)
Software*	32	3000	26.00
Hardware	32	59.47	0.17

- CPU: Intel Xeon @3.0GHz

As the result, the hardware model achieves about 150 times speedup in comparison with software execution. Here we assume the Spartan3 (xc3s15000) as a target device which has 13,312 logic slices. The model uses 99% of slices of the FPGA device.

VI. CONCLUSIONS

In this paper, we proposed the MAM using a stored pattern independent kernel image. Usage of the stored pattern independent kernel image offers fruitful benefits: 1) a good design ability of kernel images, 2) improvement of recall impossibility for corrupted kernel image, and 3) improvement of the perfect recall rate using block splitting manner. Furthermore, it was confirmed by the logic simulation that the hardware implementation of the proposed method achieved the great acceleration (more than 150 times) in comparison with software execution. The proposed method is necessary for the Brain-IS working in real world like human beings and is expected as one of associative memories treating natural images.

When an input pattern is completely included with the other pattern in the stored patterns or vice versa, the MAM can not discriminate those patterns. In this case, the perfect recalling can not be achieved even if the input image does not include any noise. In the future works, we will try to solve the problem and develop a practical associative memory.

ACKNOWLEDGMENTS

A part of this work was supported by the COE Program <Center#J19> granted in 2003 to Kyushu Institute of Technology by MEXT of Japan.

REFERENCES

[1] T. Kohonen, "Correlation marix memories," IEEE Trans. Computers, C-21, pp.353-359, 1972.

[2] D.H. Ackley, G.E. Hinton, and T.J. Sejnowski, "A learning algorithm for Boltzmann machines," Cognitive Science, Vol.9, pp.147-169, 1985.

[3] B. Kosko, "Bidirectional Associative Memories," IEEE Trans. Systems, Man and Cybernetics, Vol.18, No.1, pp.49-60, 1988.

[4] M. Hagiwara, "Multidirectional associative memory," International Joint Conference on Neural Networks, I, pp.3-6, 1990.

[5] G.X. Ritter, P. Sussner, and J.L. Diaz-de-Leon, "Morphological associative memory," IEEE Trans. Neural Networks, Vol.9, No.2, pp.281-293, 1998.

[6] J.L. Davidson and G.X. Ritter, “A theory of morphological neural networks,” in Digital Optical Computing Ⅱ, Vol.1215, Proc. of SPIE, pp.378-388, July 1990.

[7] J.J. Hopfield, “Neural network and physical systems with emergent collective computational abilities,” Proc. of the National Academy of Sciences of the United States of America, Vol.79, No.8, pp.2554-2558, 1982.

[8] M. Hattori, A. Fukui, and H. Ito, “A fast method to decide kernel patterns for Morphological Associative Memory,” Transactions of the Institute of Electrical Engineers of Japan. C, Vol.123, No.10, pp.1830-1838, 2003.

[9] P. Sussner, “Associative morphological memories based on variations of the kernel and dual kernel methods,” Neural Networks, Vol.16 (5-6), pp.625-632, 2003.

[10] T. Ida, S. Ueda, M. Kashima, T. Fuchida, and S. Murashima, “On a method to decide kernel patterns of morphological associative memory, ” D-Ⅱ, Vol.J83-D-Ⅱ, No.5, pp.1372-1380, 2000.

[11] T. Saeki, T. Miki, “Block splitting type morphological associative memory and its recall rate improvement by majority logic approach,” International Journal of Innovative Computing Information and Control, Vol. 4, No. 9, pp.2195-2204, 2008.