Biol. Cybern. 76, 129–137 (1997)

Biological Cybernetics
© Springer-Verlag 1997

Associative memory with the reaction-diffusion equation

Hideo Yuasa[1,2], Satoshi Ito[2], Koji Ito[3], Masami Ito[2]

[1] School of Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-01, Japan
[2] Bio-Mimetic Control Research Center, The Institute of Physical and Chemical Research (RIKEN), 8-31, Rokuban 3-Chome, Atsuta-ku, Nagoya 456, Japan
[3] Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, 4259, Nagatsuda, Midori-ku, Yokohama 226, Japan

Received: 11 September 1995 / Accepted in revised form: 6 August 1996

Abstract. Visual information processing in biological systems is thought to be performed in a distributed manner. A great number of visual neurons interact with one another; some are cooperative and some are competitive, but almost all are local and parallel. Such distributed architecture has become one of the most significant concepts in engineering. It has become very important to demonstrate the principle of acquiring structures or functions suitable to the environment by means of local interactions among neighboring components. This paper proposes an image associative memory constructed by the local connection of many elements. Natural images originally have a high correlation between neighboring pixels. Conventional methods, however, have not utilized this property of images. Our associative memory arranges the components (elements) in a plane, where connections are limited to neighbors, so that the spatial relation of pixels, at least, should be conserved. This local connection is the critical factor in the parallel processing architecture, because the amount of information which each component receives decreases dramatically with increases in the total connecting structure. Our associative memory has the dynamical property described by the reaction-diffusion equation. The reaction term expresses competition among the stored images, with the result that only one image is retrieved. The diffusion term, on the other hand, can evaluate the similarity of each stored image to the input, the comparison being executed at the pixel level. From an algorithmic viewpoint, the diffusive process is feasible with local operations.

1 Introduction

Vision is one of the most important senses by which biological systems recognize their environment. For example, humans depend on visual processing for about 60% of their sensory information. Visual perception comes from images projected onto the retina which are continuous in time, indicating that the visual system processes a large amount of spatio-temporal information in a short time. Such high-speed perception requires not only the parallel action of visual processing units but also their local connection. In fact, visual information flows in the same direction between some processing stages, implying that they never have a connection with all the other stages. It seems highly likely, from the viewpoint of computing speed, that most visual perception is achieved not in a global and sequential manner but in a local and parallel one – if the speed of visual perception is contrasted with the delay in visual cells (Kanizsa 1979). Accordingly, local parallel processing is the key method for the fast treatment of abundant data.

Natural images have the property that the correlation between pixels becomes greater as the pixels get closer to each other, and smaller as they become more distant. This correlation is conserved by keeping topography in images, which necessitates only local connections among pixels. In other words, parallel processing reflecting spatial continuity by limited connections with neighbors could improve the performance of image recognition.

Various methods of parallel processing have been proposed, such as associative memory (Kohonen 1977), neural networks (Arbib and Hanson 1987; Matsuoka 1989), Neocognitron (Fukushima 1988) and synergetic computation (Haken 1991). In Neocognitron with a multilayered neural network, the range of local operation is extended by making the receptive region large for the neurons in the higher layer, and the cells which selectively respond to the specific stimulus are self-organized in the highest layer. All the other methods, however, treat images (two-dimensional data sets) as vectors (one-dimensional data sets), with the result that the spatial relations in the images partially disappear. In addition, operations on vectors, such as inner product, usually need all the vector elements, indicating that each element is required to connect with all other elements. When considering hardware, such a totally connective structure leads to an explosive growth in the number of connecting lines between components. On the other hand, some methods make good use of the spatial relations in images. Marr (1982), for example, has proposed a stereo vision which imposes spatial continuity as one of the main constraints for depth perception. Standard regularization theory for early vision problems has

Correspondence to: H. Yuasa (Tel.:+81-52-789-2741;' Fax.:+81-52-789-2738; e-mail:yuasa@ishidalab.nuem.nagoya-u.ac.jp) also included many types of spatial differential operators, which conserve some types of spatial smoothness (Poggio et al. 1985).

This paper proposes a new model of associative memory. Two main points should be noted. First, the associative memory consists of many elements, connections between which are limited to nearest neighbors, so that the spatial relations of the image can be reflected. Secondly, all the elements have the same dynamical property, which depends on the information of elements in a local area. In concrete terms, we have described the dynamics by the reaction-diffusion equation,

$$\partial a_k(\boldsymbol{\xi}, t)/\partial t = -\partial V/\partial a_k + D\Delta a_k \qquad (1)$$

with

$$V = -\frac{1}{2}\sum_{k=1}^{M} a_k^2 + \frac{1}{4}\sum_{k=1}^{M}\sum_{\substack{k'=1 \\ k' \neq k}}^{M} a_k^2 a_{k'}{}^2 + \frac{1}{4}\left\{\sum_{k=1}^{M} a_k^2\right\}^2 \qquad (2)$$

The notation and detailed explanation will be found in Sect. 2.2. This reaction-diffusion equation can achieve image association in a local parallel manner by good use of image properties.

The next section first explains the methodological background which led us to develop this associative memory, and then presents the proposed method: associative memory with the reaction-diffusion equation. Section 3 shows the simulation results and Sect. 4 discusses the advantages of the method and future studies.

2 Image association

2.1 Basic ideas

2.1.1 Orthogonal projection.

Firstly, this section deals with what auto-associative memories are (Haken 1990). Associative memories store in advance a large amount of coupled data $(x_1, y_1), \ldots, (x_n, y_n)$. When a datum x_i is given, associative memories should provide the other one of the pair, i.e., y_i. In the case where the paired data x_i and y_i are the same, it is called an auto-associative memory. The given datum x_i generally contains some kinds of noise. Auto-associative memories select the datum most similar to the input in the stored data and provide it in the complete (noiseless) form.

Orthogonal projection is one of the powerful methods for achieving auto-associative memories. Many papers have defined orthogonal projection in vector space. In this case, it is necessary first to transform images into vectors. For example, we can change images into vectors by picking up pixel values from each image one by one and arranging them in order. Here, we introduce some notation. $\mathbf{q}$ denotes an input image and $\mathbf{v}_i$ $(i = 1, \ldots, M$; M is the number of memorized images) denote memorized images. In addition, Π defines the vector subspace spanned by all stored images $\mathbf{v}_i$ $(i = 1, \ldots, M)$.

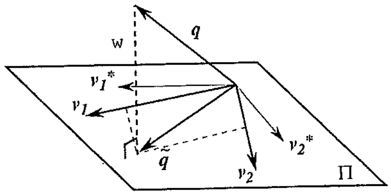


Fig. 1. Orthogonal projection in vector space. The input $\mathbf{q}$ is projected into the vector subspace Π spanned by the memorized images $\mathbf{v}_1$ and $\mathbf{v}_2$. $\tilde{\mathbf{q}}$ defines the projected vector

Figure 1 shows the concept of orthogonal projection. It firstly projects input image vector $\mathbf{q}$ into vector subspace Π (the projected vector is denoted by $\tilde{\mathbf{q}}$),

$$\mathbf{q} = \tilde{\mathbf{q}} + \mathbf{w} \qquad (3)$$

and then decomposes $\tilde{\mathbf{q}}$ into the components of each memorized image within Π,

$$\tilde{\mathbf{q}} = \sum\nolimits_{i=1}^{M} a_i \mathbf{v}_i \qquad (4)$$

Here, $\mathbf{w}$ denotes a residual vector, which corresponds to the distance between input vector $\mathbf{q}$ and subspace Π, implying, from the property of orthogonal projection, that $\tilde{\mathbf{q}}$ is closer to $\mathbf{q}$ than any other vector in subspace Π. Therefore $\tilde{\mathbf{q}}$ resembles the input image most in subspace Π.

As mentioned in Sect. 1, neighboring pixels generally indicate a high correlation. Such topographical information (i.e., how pixels are connected, or the distance between two pixels) should be preserved in an effective association process. However, previous works using the concept of the orthogonal projection (Kohonen 1973; Fuchs and Haken 1988; Matsuoka 1990) have processed images in the vector form like this. In addition, vectorial operations, such as inner product, require all the elements of the vectors simultaneously, implying that the total connection of vector elements is indispensable. A method is needed that reflects the topography with local connection only.

2.1.2 Associative memory with the diffusion equation.

In order to utilize topography in images, we first express images as functions of the spatial position (x, y), i.e., $q(x, y)$ and $v_i(x, y)$ $(i = 1, \ldots, M)$. They respectively provide a pixel value of the input and stored images at the coordinate (x, y). Here, M denotes the number of images in store. This expression reflects the spatial relation of pixels on the two arguments x and y.

According to the orthogonal projection in the functional space, we decompose the input image $q(x, y)$ as follows:

$$q(x, y) = \sum\nolimits_{i=1}^{M} a_i v_i(x, y) + w(x, y) \qquad (5)$$

where $w(x, y)$ is the residual function. For the sake of simplicity, $w(x, y)$ as well as $q(x, y)$, $v_i(x, y)$ $(i = 1, \ldots, M)$ are all assumed to be C^∞. In the same way for vector space, we define $\tilde{q}(x, y)$ as follows:

$$\tilde{q}(x, y) = \sum\nolimits_{i=1}^{M} a_i v_i(x, y) \qquad (6)$$

$\tilde{q}(x, y)$ is the closest to $q(x, y)$ in function space Π. [Π denotes function subspace spanned by the stored images $v_i(x, y)$ $(i = 1, \ldots, M)$.] Thus, we consider $\tilde{q}(x, y)$ as the desired association result from $q(x, y)$.

a_i $(i = 1, \ldots, M)$ in (5) can be calculated by using the adjoint functions $v_i^*(x, y)$ of each $v_i(x, y)$ $(i = 1, \ldots, M)$,

$$a_i = \iint_\Omega v_i^*(x, y) q(x, y) dxdy \tag{7}$$

where adjoint functions $v_i^*(x, y)$ $(i = 1, \ldots, M)$ satisfy

$$v_i^*(x, y) = \sum_{j=1}^{M} c_j v_j(x, y) \tag{8}$$

$$\iint_\Omega v_i^*(x, y) v_j(x, y) dxdy = \delta_{ij} \tag{9}$$

Here, Ω denotes the whole image ($\|\Omega\| < \infty$) and δ_{ij} is Kronecker's delta.

Equation (8) means that the adjoint function $v_i^*(x, y)$ exists within the function subspace Π. Equation (9), on the other hand, means that every adjoint function $v_i^*(x, y)$ is orthogonal to each $v_j(x, y)$ $(j \neq i)$ and normalized so that the inner product with $v_i(x, y)$ becomes equal to 1. From the characteristics of orthogonal projection, the following equation is obviously satisfied:

$$\iint_\Omega v_i^*(x, y) w(x, y) dxdy = 0 \tag{10}$$

The integral computation generally requires all the values of integrand in the integral domain, implying that a totally connecting structure is necessary in an ordinary method. Here, we have considered the integral computation in (7) by parallel processing with local restricted connection (Yuasa et al. 1992). Then, diffusion plays a crucial role.

We explain here that the diffusion equation with the periodic boundary condition averages the values in the diffusive region while keeping the total value the same. For simplicity, we consider the one-dimensional case. The diffusion is described by a partial differential equation of a parabolic type, sometimes used to express heat conduction. Thus we use the example of heat diffusion in our explanation. First, the function expressing the initial heat distribution is given in the diffusive region Ω. As is well known, the diffusive process transports heat from an area of high temperature to low temperature in a spatially continuous manner. Then, if we impose a boundary condition such that the inflow and outflow of heat are equal, the total amount of heat never changes, with the result that the heat distribution becomes averaged while conserving the total amount of heat (Fig. 2). Because the periodic boundary condition is one such condition (Fig. 3a), every part of the diffusive region will contain the same amount of heat. In the case of (7), the diffusive region corresponds to the plane. Then the periodic boundary condition is equivalent to considering the diffusive region as a two-dimensional torus (Fig. 3b).

Boundary conditions are worth discussing further. The above explanation states: the function $f(x, y; t)$ is defined whose initial value is given by

$$f(x, y; 0) = v_i^*(x, y) q(x, y) \tag{11}$$

Then, the integral computation

$$S = \iint_\Omega f(x, y, t) dxdy \tag{12}$$

is executable by using the diffusion equation

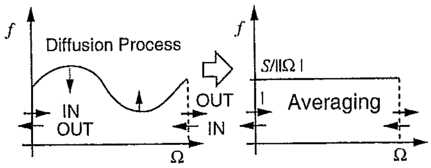


Fig. 2. Averaging by diffusion. Initially, the heat is distributed unevenly. After heat conduction by diffusion, however, the distribution becomes even. If the inflow is equal to the outflow, the total amount of heat is kept constant

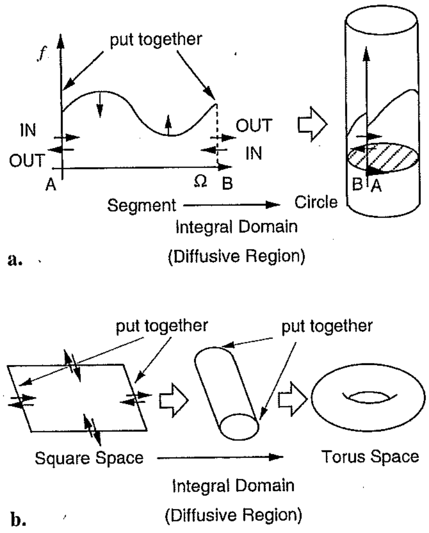


Fig. 3a,b. Periodic boundary condition. **a** One-dimensional case. The periodic boundary condition makes an integral domain circle, where the two vertices A and B are regarded as the same if the integral domain is given as the segment AB. **b** Two-dimensional case. The integral domain becomes a two-dimensional torus

$$\frac{d}{dt} f(x, y; t) = D\nabla^2 f(x, y; t) \tag{13}$$

Here, D denotes the diffusion coefficient and

$$\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2 \tag{14}$$

When using (12),

$$\begin{aligned} \frac{dS}{dt} &= \frac{d}{dt} \iint_\Omega f(x, y, t) dxdy \\ &= \iint_\Omega D\nabla^2 f(x, y, t) dxdy \end{aligned} \tag{15}$$

Gauss's divergence theorem turns this into

$$\frac{dS}{dt} = D \int_{\partial\Omega} \nabla f(s, t) \cdot \mathbf{n}(\mathbf{s}) ds \tag{16}$$

where $\partial\Omega$ denotes the boundary of Ω and $\mathbf{n}(\mathbf{s})$ denotes the normal vector of the boundary. In order to keep the area S, i.e., $dS/dt = 0$, the following equation should be satisfied at any time:

$$\int_{\partial\Omega} \nabla f(s, t) \cdot \mathbf{n}(\mathbf{s}) ds = 0 \tag{17}$$

We use the periodic boundary condition as one of the conditions satisfying (17). The boundary condition of a free end can also be considered as such a condition. From the point

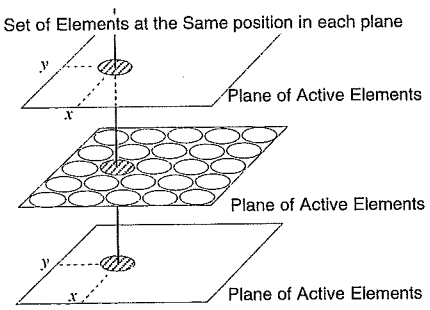


Fig. 4. Planes and the same coordination set of elements. The plane consists of the active elements, all of which are in the same order as the pixels in the stored image. The same coordination set includes the elements all of which are located at the same position [in the figure, (x, y)] in each plane

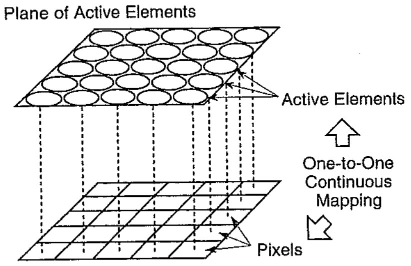


Fig. 5. Mapping pixels to active elements. This mapping is one-to-one and continuous. Here, only one stored image is drawn

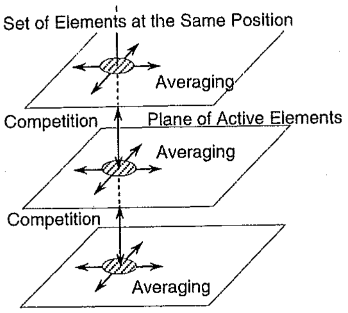


Fig. 6. Interaction. This is cooperative (averaging) within each plane but competitive between planes, i.e., within a set of elements at the same position in each plane

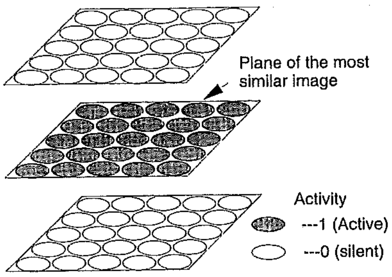


Fig. 7. Stable state. In this state, only one plane has elements that are active (activity 1) while all other planes have elements that are silent (activity 0)

of view of reflecting spatial continuity, we should impose the latter, because the periodic boundary condition does not reflect the spatial relation in the boundary. However, the essence of integral computation with the diffusion process does not change according to which boundary conditions are used. Therefore, we adopt the periodic boundary condition for simplicity. In addition, the periodic boundary condition has the advantage that it makes convergence of dynamics fast.

Finally, we define the output image as follows:

$$q(x, y; t) = \sum_{i=1}^{M} a_i(x, y; t) v_i(x, y) \tag{18}$$

Equation (18) means that the output is the most similar image to the input in the function subspace Π and the residual function $w(x, y)$ has been removed from the input. It also implies that noise within subspace Π is not removable.

It should be noted that the diffusive process only needs information in the local area, indicating that nearest-neighbor connections are sufficient for the diffusive process. The diffusive process transfers local information throughout the diffusive region by interaction restricted to neighbors.

2.2 Associative memory with the reaction-diffusion equation

2.2.1 Recognition strategy

In the above method, however, we still have two problems. Firstly, (7) includes adjoint functions, which cannot be calculated by local operations alone. Secondly, the output never becomes exactly the same as one of the stored images [i.e., $v_i(x, y)$], but becomes $\tilde{q}(x, y)$ in (6). This is because the orthogonal projection also computes components other than the main one. We have solved the first problem by changing the way the initial values are set, and the second one by introducing a reaction term representing competition among images.

We will first explain the hardware architecture (Fig. 4). Associative memory possesses a number of planes consisting of active elements (such as neurons). Each of these elements has a one-dimensional state, here called activity. The number of planes is the same as the number of images in store, and the number of elements is the same as the number of pixels in one image. The active elements in the planes are arranged such that the topographical relation of the pixels is conserved, indicating that there is one-to-one continuous mapping between pixels and elements (Fig. 5).

There are two types of connections which are distinguishable both structurally and functionally. One is cooperative within the planes and the other is competitive between the planes. The former connections link the nearest neighbors in the plane, with the result that each element connects with four others. The latter connections link all the elements at the same coordinate in every plane (Fig. 6).

The interaction varies according to the type of connection. At connections within the plane, the interaction works so that the activity of elements takes an equal value in each plane. At the connections between planes, on the other hand, the interaction acts as winner-takes-all, implying that, in a set of elements at the same position in each plane, only one is active while the others are silent.

By adjusting the strength of the two interactions, the following stable state is established: one plane has elements

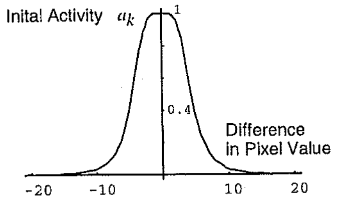
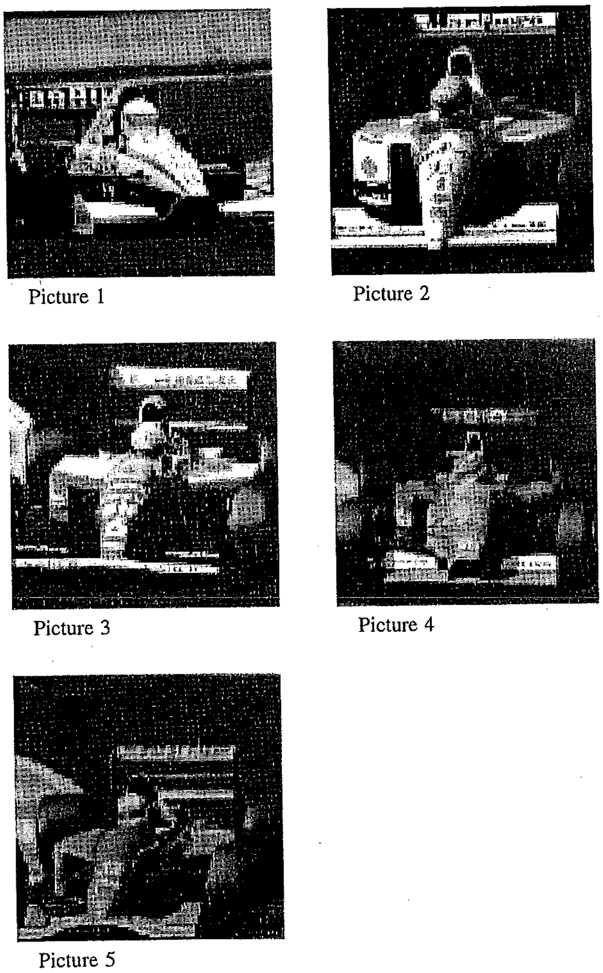


Fig. 8. The function that initializes the activity of the element. The equation for this is $y = c/(x^{\alpha} + c)$, where $\alpha = 4$ and $c = 400$


Picture 1

Picture 2

Picture 3

Picture 4

Picture 5

Fig. 9. Five stored images. They are all 100 × 100 pixels and the gray value ranges from 0 to 255

all of which are active, while the other planes have elements all of which are silent (Fig. 7). The activated plane should then correspond to the image most similar to the input.

Here, we should explain some notation. $a_k(\xi, t)$ denotes the activity of the element at the coordinate $\xi = (x, y)$ in the kth image at time t. The activity varies from 0 to 1: activity 1 means active while activity 0 means silent. $v_k(\xi)$, on the other hand, denotes the pixel value at the coordinate ξ in the kth image, which never evolves in time. It is assumed that the number of image in storage is M, implying that k ranges from 1 to M. And $p(\xi)$ denotes the pixel value at position ξ in the input image.

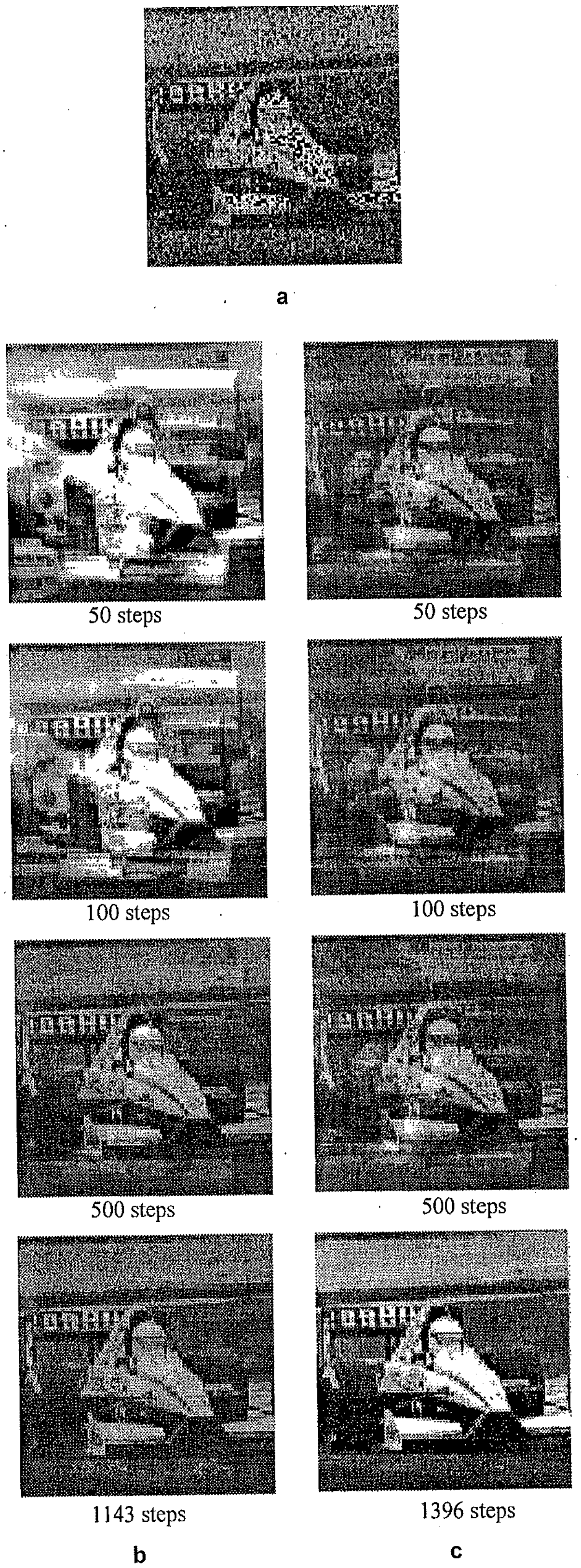


Fig. 10a–c. Simulation results for the input with high-frequency noise. **a** Input image. **b** Output with the diffusion equation. From the *top*: 50, 100, 500, 1143 steps. **c** Output with the reaction-diffusion equation. From the *top*: 50, 100, 500, 1396 steps

2.2.2 Initialization.

The plane corresponding to the image most similar to the input should have elements of greater activity in the initial state. As such an initialization, we use the following equation (Fig. 8):

$$\begin{aligned} a_k(\xi, 0) &= f(p(\xi), v_k(\xi)) \\ &= c/\left[\{p(\xi) - v_k(\xi)\}^{\alpha} + c\right] \end{aligned} \tag{19}$$

Equation (19) describes the comparison at the pixel level. If two pixel values are equal, the initial activity is set to 1. The greater the difference in pixel values, the smaller the initial value that is set.

It should be noted that this is not the only possible initialization and many other methods exist – Gaussian error function, for example. It would be worth exploring which initialization is the best.

2.2.3 Dynamics

We have described the dynamics of activity as follows:

$$\partial a_k(\xi, t)/\partial t = -\partial V/\partial a_k + D\nabla^2 a_k \tag{20}$$

Here

$$V = -\frac{1}{2}\sum_{k=1}^{M} a_k^2 + \frac{1}{4}\sum_{k=1}^{M}\sum_{\substack{k'=1 \\ k' \neq k}}^{M} a_k^2 a_{k'}{}^2 + \frac{1}{4}\left\{\sum_{k=1}^{M} a_k^2\right\}^2 \tag{21}$$

$$\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2 \tag{22}$$

$a_k = a_k(\xi, t)$, and D is a diffusion coefficient expressing the trade-off between two types of interaction. In addition, we have to impose a boundary condition on (20) such that inflow and outflow become equal. Here we select the periodic boundary condition (see Sect. 2.1.2).

Equation (20) takes the form of the reaction-diffusion equation: the first term denotes the reaction term acting in a set of elements at the same position, i.e., between planes, while the second term denotes the diffusion term working within each plane.

For the reaction term we have used the gradient system of the potential function (21) introduced by Haken (1988). He has proved that this potential works as winner-takes-all, with the result that only one element becomes active (activity 1) in a set of elements from the same position at stable state. On the other hand, we have also adopted the diffusion term which averages the values in the diffusive region, as shown in Sect. 2.2.

As shown in Fig. 7, we design the stable state as follows; only one plane has elements that are active (activity 1) while the other planes have elements that are silent (activity 0). A sufficiently large diffusion coefficient D can achieve this stable state. In this case, the diffusive process first makes the activity equal in all planes. Then the competition introduced by the reaction term becomes dominant. As a result, the plane with the largest total activity suppresses the other planes, implying that the dynamics converges to the desired stable state.

The evolution equation (20) also becomes a gradient system in function space

$$\frac{\partial a_k(\xi)}{\partial t} = -\frac{\delta \tilde{V}}{\delta a_k(\xi)} \tag{23}$$

the potential functional of which is given by $\tilde{V}$:

$$\tilde{V} = \int \left\{ V(\xi) + \frac{D}{2}\sum_{k=1}^{M}\left(\frac{\partial a_k(\xi)}{\partial \xi}\right)^2 \right\} d\xi \tag{24}$$

(Mikhaikov 1990). Equation (24) can be regarded as one type of standard regularization expanded to function space (Poggio et al. 1985). Actually, we can consider the reaction-diffusion equation as one type of standard regularization.

The dynamics of (20) lead to the stable state as in Fig. 7. Therefore, we can construct the output image as follows:

$$p(\xi, t) = \frac{\sum_{k=1}^{M} v_k(\xi) a_k(\xi, t)}{\sum_{k=1}^{M} a_k(\xi, t)} \tag{25}$$

where each activity plays the role of a weight in the summation.

3 Simulations

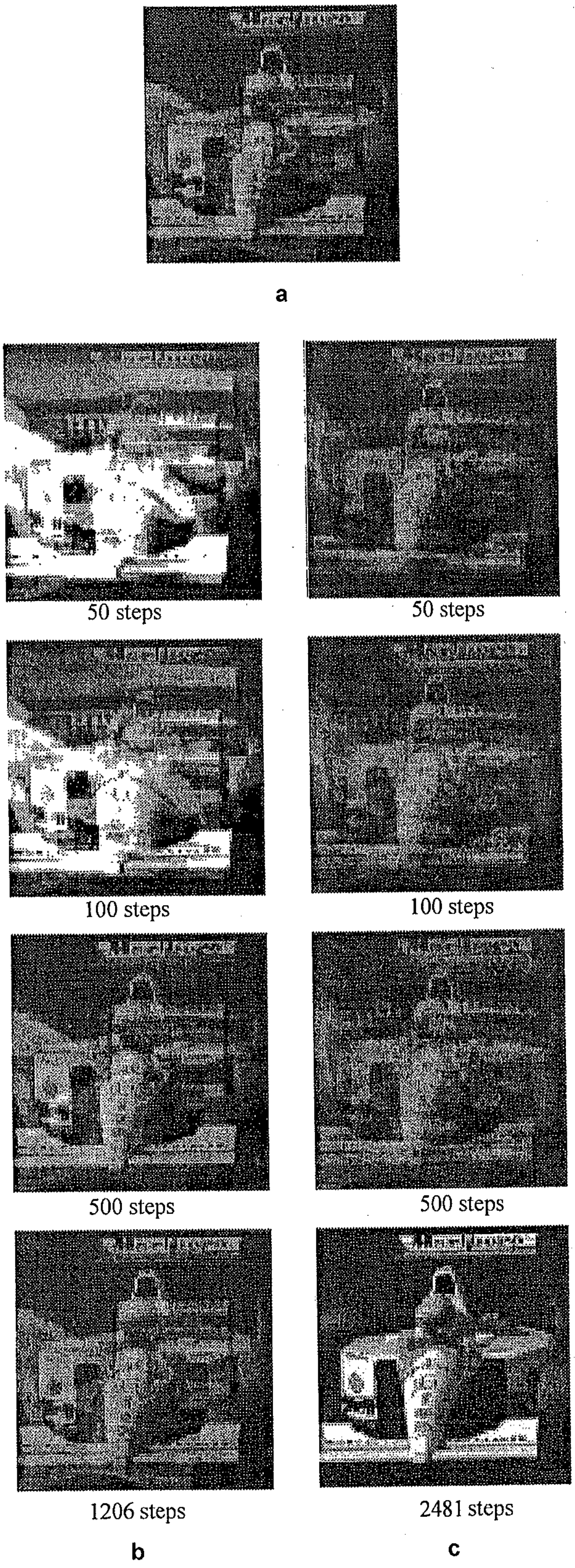
This section shows the simulation results using two methods: the diffusion equation only (13) or the reaction-diffusion equation (20). Five images (100×100 pixels with gray values ranging from 0 to 255) are stored in advance (Fig. 9). Three kinds of image are offered, and the time evaluation of output images is shown. For both methods, the diffusion coefficient D is set to 200.

3.1 Recalling from an image with high-frequency noise

Image 1 containing high-frequency noise is used as the input image. As shown in Fig. 10a, uniformly random noise (maximum amplitude 40 and average 0) is added to all pixels. Figure 10b and c show that the output image becomes the same as image 1, indicating that the high-frequency noise is removable in both methods.

3.2 Recalling from two mixed images

This simulation examines whether the noise existing in the subspace spanned by the stored images (Π in Fig. 1) is removable or not. Thus the input image is made by mixing images 2 and 5 at a ratio of 1 : 1 (Fig. 11a). In the case of diffusion only, the output becomes, as we expect, the same as the input image. This is because the diffusion equation is derived from the orthogonal projection. The result with the reaction-diffusion equation, on the other hand, converges to one of the memorized images, i.e., image 2 (Fig. 11c), implying that associative memory with the reaction-diffusion equation can get rid of the noise in the subspace spanned by the stored images. Since the proportions in the mixture were the same for each image, the output converged to one of them with the same probability. If the proportions are different (for example, the ratio 6 : 4), the image contributing the greater proportion will be retrieved.



a

50 steps 50 steps

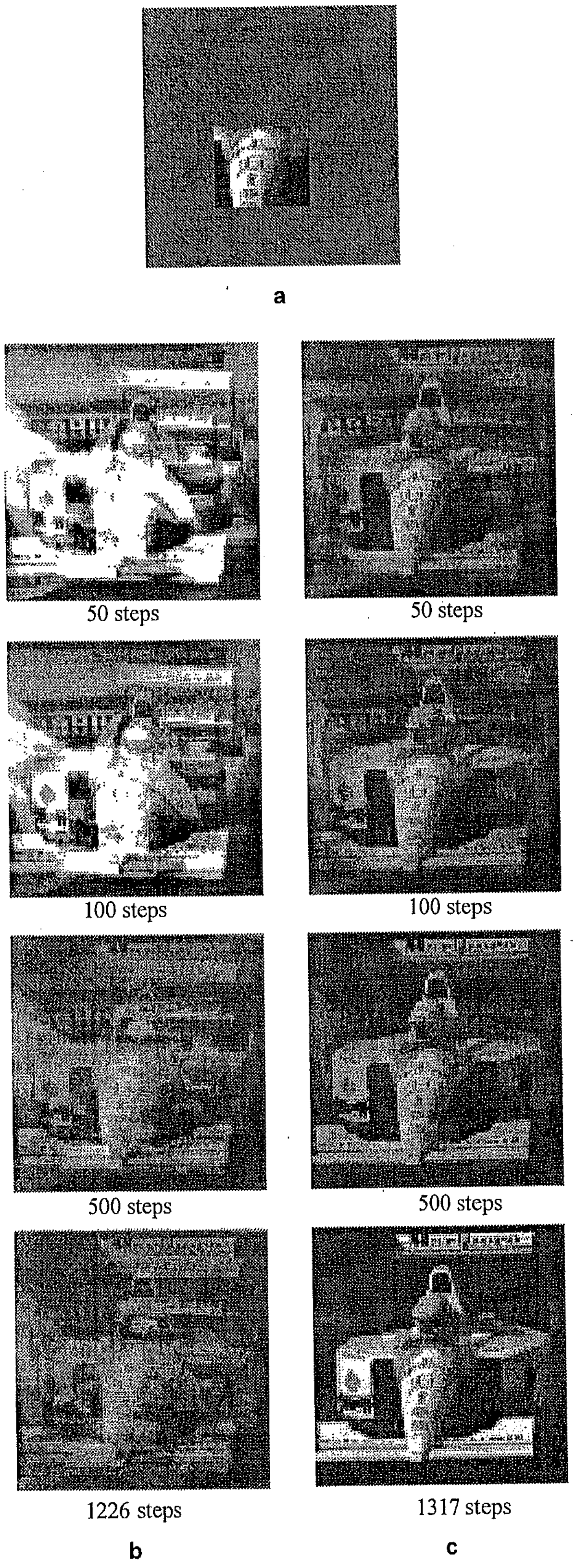
100 steps 100 steps

500 steps 500 steps

1206 steps 2481 steps

b c

Fig. 11a–c. Simulation result for the input in which two images are mixed. **a** Input image. **b** Output with the diffusion equation. From the *top*: 50; 100, 500, 1206 steps. **c** Output with the reaction-diffusion equation. From the *top*: 50, 100, 500, 2481 steps



a

50 steps 50 steps

100 steps 100 steps

500 steps 500 steps

1226 steps 1317 steps

b c

Fig. 12a–c. Simulation result for part of an image. **a** Input image. **b** Output with the diffusion equation. From the *top*: 50, 100, 500, 1226 steps. **c** Output with the reaction-diffusion equation. From the *top*: 50, 100, 500, 1317 steps

3.3 Recalling from a part of the image

This simulation examines the ability of association by inputting a part of image 2 (Fig. 12a). The output image for the diffusion equation contains components other than image 2 (Fig. 12b) while the output for the reaction-diffusion system converged exactly to image 2 (Fig. 12c).

4 Conclusion

This paper has proposed an associative memory based on local parallel processing. This has three advantages: Firstly, parallel processing makes performance time short. Secondly, the local connecting architecture greatly reduces the number of connections. Lastly, the type of connection conserves the topography of pixels in images, which reflects the original property of natural images that neighboring pixels have a higher correlation than distant pixels.

Schmutz and Banzhaf (1992) have proposed the 'diffusive Haken model which combines the original Haken model with a diffusive process. Our idea is the same as theirs, i.e., diffusion transfers local information throughout the whole system. The role of Haken's potential function, however, is essentially different. The diffusive Haken model defines the diffusive process in the same space where Haken's potential function works for the purpose of the robustness, which is achieved by sharing the information for the winner with its neighbors by diffusion. In our associative memory, on the other hand, the workspace of the two types of interaction is structurally separated. Our model explicitly uses the winner-take-all function of the Haken model to select the image that is recalled.

This associative memory can be regarded as a dynamical system, the dynamics of which are described by the reaction-diffusion equation. The diffusion term acts to evaluate the overall similarity of each stored image to the input image. The similarity is computed at the each pixel, but diffusion averages it in each plane of active elements. The reaction term, on the other hands, works as a competition among stored images, with the result that the associative memory retrieves only the image that wins the competition. Due to the reaction term, the retrieved image contains no components of other stored images.

As for complete image retrieval, the energy function has been commonly used (Hopfield 1982; Haken 1988). Each minimum of the energy function corresponds to a stored image. Thus over time the state settles into one of the minima, implying that the image has become exactly the same as one of images in store. Although our associative memory utilizes this concept, it differs from traditional methods in the number of energy functions: the energy function is assigned to each of the element sets consisting of elements at the same position in every image, where the time evolution is executed in parallel. From an engineering point of view, such a distributed architecture is expected to possess flexibility, adaptability, and fault-tolerance (Ito 1995).

Another property of this associative memory is that the structure can make the most use of the topography which includes the spatial relations in images. The importance of topography has been asserted in the Self-Organizing Map (Kohonen 1989), neural gas model (Martinetz and Schulten 1991) and growing cell structure (Fritzke 1994), but these differ from our approach in the following way: all these models have focused on how a self-organizing system acquires a topology-preserving structure, while our approach is mainly concerned with how we can take advantage of the topographical relation for image processing. It should be very efficient for image processing, because images are formed from pixel connections and thus the order of pixels provides enough information for what is drawn in images.

Finally, we consider future works. Our system is weak in recognizing transformed images, i.e., translated, rotated or scaled images. This is because the activity of elements is initialized by comparing the pixel values at the same coordinate in the input and each stored image, and transformations break this one-to-one continuous mapping between pixels and elements. For transformed image retrieval, the invariant to transformations should be extracted. One method has been proposed (Fuchs and Haken 1988) in which a combination of both Fourier transform and transformation to a polar coordinate can detect the invariant with respect to translation, rotation, and scaling.

Acknowledgements. We thank Mr. Seiji Toyohara of NKK Corporation for his cooperation in this research. Part of the work presented here was supported by a science research fund from the Ministry of Education, Science and Culture of Japan under grant #04218209,06452253.

References

Arbib MA, Hanson AR (eds) (1987) Vision, brain, and cooperative computation. MIT Press, Harvard, Mass

Fukushima K (1980) Neocognitron: a self organizing neural network for a mechanism of pattern recognition unaffected by shift in position. Biol Cybern 36:193-202

Fukushima K (1988) Neocognitron: a hierarchical neural network capable of visual pattern recognition. Neural Networks 1:119-130

Fritzke B (1994) Growing cell structures: A self-organizing network for unsupervised and supervised learning. Neural Networks 7:1141-1460

Fuchs A, Haken H (1988) Pattern recognition and associative memory as a dynamical processes in a synergetic system. Biol Cybern 60:17-22,107-109

Haken H (1988) Neural and synergetic computers. Springer; Berlin Heidelberg New York

Haken H (1990) Synergetic computers and cognition. Springer; Berlin Heidelberg New York

Haken H (1991) Synergetic computers and cognition: a top-down approach to neural nets. Springer; Berlin Heidelberg New York

Hopfield JJ (1982) Neural network and physical system with emergent collective computational abilities. Proc Natl Acad Sci USA 79:2554-2558

Ito M (eds) (1995) The declaration of autonomous decentralized systems. Ohm-sha co, Ltd. (in Japanese)

Kanizsa G (1979) Organization in vision: essays on gestalt perception. Praeger, New York

Kohonen T (1973) Representation of associated pairs by matrix operators. IEEE Trans C-22:701-702

Kohonen T(1977) Associative memory: a system-theoretical approach. Springer; Berlin Heidelberg New York

Kohonen T (1989) Self-organization and associative memory, 3rd edn. Springer; Berlin Heidelberg New York

Marr D (1982) Vision: a computational investigation into the human representation and processing of visual information. WH Freeman, San Francisco

Martinez T, Schulten K (1991) A 'Neural-gas' network learns topologies. Artif Neural Networks, North Holland, I:397-402

Matsuoka K (1989) An associative network with cross inhibitory connections. Biol Cybern 61:393-399

Matsuoka K (1990) On various structures of orthogonal projection type of associative network. IEICE Trans J73-D-II:641-647

Mikhailov AS (1990) Foundation of synergetics I. Springer; Berlin Heidelberg New York

Shimutz M, Banzhaf W (1992) Robust competitive networks. Phys Rev A 45:4132-4145

Poggio T, Torre V, Koch C (1985) Computational vision and regularization theory. Nature 317:314-319

Yuasa H, Toyohara S, Ito M (1992) Spatial pattern formation through diffusive process with application to pattern recognition (in Japanese). Proc 2nd SICE Symp Decentralized Autonomous Syst, SICE, pp 47–52