LETTER — Communicated by Javier Movellan

Learning to Design Synergetic Computers with an Extended Symmetric Diffusion Network

Koji Okuhara
Shunji Osaki
Department of Industrial and Systems Engineering Faculty of Engineering, Hiroshima University, Higashi-Hiroshima-shi, 739-8527 Japan

Masaaki Kijima
Faculty of Economics, Tokyo Metropolitan University, Hachiohji, Tokyo, 192-0397 Japan

This article proposes an extended symmetric diffusion network that is applied to the design of synergetic computers. The state of a synergetic computer is translated to that of order parameters whose dynamics is described by a stochastic differential equation. The order parameter converges to the Boltzmann distribution, under some condition on the drift term, derived by the Fokker-Planck equation. The network can learn the dynamics of the order parameters from a nonlinear potential. This property is necessary to design the coefficient values of the synergetic computer. We propose a searching function for the image processing executed by the synergetic computer. It is shown that the image processing with the searching function is superior to the usual image-associative function of synergetic computation. The proposed network can be related, as a special case, to the discrete-state Boltzmann machine by some transformation. Finally, the extended symmetric diffusion network is applied to the estimation problem of an entire density function, as well as the proposed searching function for the image processing.

1 Introduction

Haken, Haas, and Banzhaf (1989) proposed a synergetic computer to achieve the image-associative function. The synergetic computer is one of complex systems consisting of many subsystems whose states change according to external information. The dynamics of each subsystem is described by a set of equations whose coefficients are assumed to be known a priori. However, when the coefficient values are not known explicitly before the system starts, it is natural to consider a model in which the coefficient values are acquired by learning. Little attention has been paid to the relationship between the image-associative function and the coefficient values.

Each subsystem is modeled by a single-state variable, and the state of

Neural Computation **11, 1475–1491** (1999)

the synergetic computer is given as a multidimensional vector consisting of them. The dimension of the state vector is usually very large, but Haken (1989) showed that it is transposed to a lower-dimensional vector, the component being called an *order parameter* (Haken 1989, 1990). The order parameter is defined by an inner product of the corresponding adjoint vector of the embedded pattern and the state vector of the synergetic computer. The dynamics of the whole system is thus determined by observing behaviors of the order parameters, whose number is substantially smaller than the number of subsystems. Moreover, it is known (see, e.g., Fuchs & Haken, 1988) that the image-associative function of synergetic computers is superior to the usual associative function. Thus, it is natural to formulate the image-associative function of synergetic computers in terms of the order parameters. In this article, we follow this line and propose a learning algorithm for the coefficient values of order parameter equations in order to analyze the image-associative function of synergetic computers.

The order parameters of a synergetic computer depend crucially on the coefficient values of governing equations and converge to a stationary solution of the equations as time goes by. Therefore, designing the coefficient values so as to let the order parameters converge to a desired stationary solution is important. The design method we propose is a version of supervised learning. Supervised learning is widely studied in the literature that includes the Boltzmann machine in neural networks. The Boltzmann machine has been applied to many problems (Gutzmann, 1987; Kohonen, Barna, & Chrisley, 1988; Lippmann, 1989) and is known to be suitable especially for complex systems. Moreover, for the case of continuous outputs, symmetric diffusion networks (Movellan & McClelland, 1993, 1994) are developed that extend the discrete-state Boltzmann machine. The symmetric diffusion networks are constructed specifically to learn an entire probability density function by using covariance statistics.

Our model can be regarded as an extended model of symmetric diffusion networks. The learning algorithm in our model can also be stated, as expected, in terms of covariance statistics. The main difference between our model and the ordinary symmetric diffusion networks is due to the definition of the drift term. The ordinary symmetric diffusion networks use Hopfield's drift (Movellan & McClelland, 1993), while in our model, the drift term is derived from a nonlinear potential. The transitions in the proposed model are governed by a stochastic differential equation (SDE), and the order parameter converges to a Boltzmann distribution, under some condition on the drift term, derived by a Fokker-Planck equation (Feidlin & Wentzell, 1984). We call the proposed model an *extended symmetric diffusion network* (ESDN). The ESDN can be translated, as a special case, to the discrete-state Boltzmann machine with ease.

We apply the ESDN to the estimation problem of an entire probability density function and a searching function for the image processing executed by a synergetic computer. For this purpose, it is essential to clarify the relationship between the weights of the ESDN and the coefficient values of governing equations of the synergetic computer. The coefficients for the desired function can then be designed by learning in the ESDN. The learning algorithm of the ESDN can be regarded as a version of the learning algorithm of synergetic computers.

In section 2, we describe the outline of the image-associative function of synergetic computers in order to introduce the notation and definitions necessary for what follows. Section 3 proposes a learning mechanism in the ESDN to design the coefficient values of synergetic computers. Extensive simulation experiments have been performed to show the usefulness of our model, and some results are reported in section 4, together with some considerations about them, where the ESDN is applied to the estimation problem of a probability density function and the proposed searching function. Section 5 concludes this article.

2 The Image-Associative Function of Synergetic Computers

In this section we introduce the dynamics of the image-associative function using synergetic computation together with the notation necessary for what follows. Suppose that the architecture of the system consists of N fully connected components. The number N is assumed to be very large. Let x_i denote the output of the ith component. The state vector of the system is denoted by $\mathbf{x} = [x_1, x_2, \ldots, x_N]^{\mathrm{T}}$ in $\Re^N$ where T denotes the transpose and $\Re^N$ the N-dimensional Euclidean space. The mth embedded pattern is encoded in the system by vector $\mathbf{v}_m = [v_1, v_2, \ldots, v_N]_m^{\mathrm{T}}$ in $\Re^N$. It is assumed that the number of embedded patterns is M, which is sufficiently smaller than N.

2.1 Dynamics. When an input pattern vector $\mathbf{x}(0)$ is offered to the system as the initial state vector, the dynamics of the system is described by an equation of the form

$$\frac{d\mathbf{x}}{dt} = \sum_m \alpha_m (\mathbf{v}_m^+ \mathbf{x}) \mathbf{v}_m - \beta \sum_m \sum_{m' \neq m} (\mathbf{v}_m^+ \mathbf{x})^2 (\mathbf{v}_{m'}^+ \mathbf{x}) \mathbf{v}_{m'} - \gamma |\mathbf{x}|^2 \mathbf{x}, \tag{2.1}$$

where the coefficients α_m $(m = 1, 2, \ldots, M)$, β, and γ are constants, and the vector $\mathbf{v}_m^+$ is the adjoint vector of the mth embedded pattern $\mathbf{v}_m$ and $|\mathbf{x}|^2 = \mathbf{x}^{\mathrm{T}}\mathbf{x}$ (Haken et al., 1989). The adjoint vector $\mathbf{v}_m^+$ is the mth row vector of the generalized inverse matrix $\mathbf{u}$ satisfying

$$\mathbf{uv} = \mathbf{I}_M, \qquad \mathbf{vu} = \mathbf{I}_N, \tag{2.2}$$

where $\mathbf{I}_n$ denotes the identity matrix of order n and $\mathbf{v} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M]$. More specifically, denoting the Kronecker delta by $\delta_{mm'}$, that is, $\delta_{mm'} = 1$ if $m = m'$ and $\delta_{mm'} = 0$ otherwise, the adjoint vectors have the property

$$\mathbf{v}_m^+ \mathbf{v}_{m'} = \delta_{mm'} \tag{2.3}$$

2.2 Order Parameters. It is known that synergetic computers can remove noise that lies inside the space spanned by the embedded patterns $\mathbf{v}_m$. Let $\mathbf{z} = [z_1, z_2, \ldots, z_N]^{\mathrm{T}}$ in $\mathfrak{R}^N$ be the noise vector orthogonal to the space spanned by the adjoint vectors $\mathbf{v}_m^+$. This means that

$$\mathbf{v}_m^+ \mathbf{z} = 0, \qquad m = 1, 2, \ldots, M. \tag{2.4}$$

For some $\mathbf{y}$ the state vector $\mathbf{x}$ can be written as

$$\mathbf{x} = \mathbf{v}\mathbf{y} + \mathbf{z}. \tag{2.5}$$

It follows from equations 2.2, 2.4, and 2.5 that the mth element of $\mathbf{y}$ is obtained as

$$y_m = \mathbf{v}_m^+ \mathbf{x}, \qquad m = 1, 2, \ldots, M. \tag{2.6}$$

The element y_m is called the mth order parameter and $\mathbf{y} = [y_1, y_2, \ldots, y_M]^{\mathrm{T}}$ in $\mathfrak{R}^M$, the order vector. Hence, the state vector $\mathbf{x}$ is translated to the order vector $\mathbf{y}$ having a less dimension via equation 2.6.

From equations 2.3, 2.5, and 2.6, we can rewrite equation 2.1 in the form of temporal change using the order parameter y_m and the noise vector $\mathbf{z}$:

$$\frac{dy_m}{dt} = \left\{ \alpha_m - \beta \sum_{m' \neq m} y_{m'}^2 - \gamma \left(\sum_m y_m^2 + |\mathbf{z}|^2 \right) \right\} y_m \tag{2.7}$$

$$\frac{d\mathbf{z}}{dt} = -\gamma \left(\sum_m y_m^2 + |\mathbf{z}|^2 \right) \mathbf{z}. \tag{2.8}$$

From equation 2.8, the noise vector $\mathbf{z}$ converges to the zero vector as time goes to infinity. This implies that the system can remove the noise vector $\mathbf{z}$ orthogonal to the space spanned by the adjoint vectors $\mathbf{v}_m^+$ from the input pattern $\mathbf{x}(0)$. Also, if the coefficient γ is large enough, then $|\mathbf{z}|^2$ in equation 2.7 can be dropped to yield

$$\frac{dy_m}{dt} = \left\{ \alpha_m - (\beta + \gamma) \sum_{m' \neq m} y_{m'}^2 - \gamma y_m^2 \right\} y_m. \tag{2.9}$$

It is clear from equation 2.9 that a suitable choice of coefficients α_m, β, and γ makes only one order parameter y_{m_0} ($m_0 \in \{1, 2, \ldots, M\}$) converge to unity while the other y_m ($m \neq m_0$) converges to zero as time goes to infinity. This means that the system can also remove the noise that lies inside the space spanned by the embedded patterns $\mathbf{v}_m$. Recall that this noise cannot be removed by the usual autoassociative function.

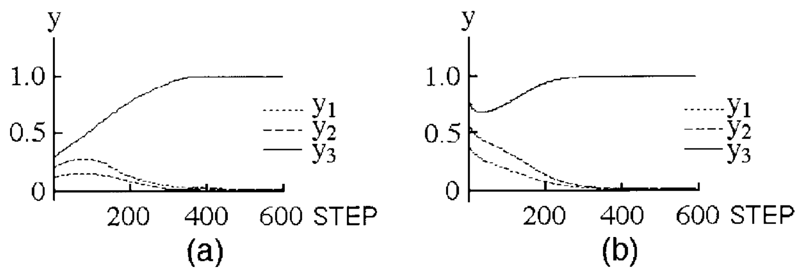

Figure 1: The transitions of order parameters y_m in the usual image-associative function with $\beta = 3$. (a) The initial state $\mathbf{y}(0) = [0.1, 0.2, 0.3]^{\mathrm{T}}$ is used. (b) $\mathbf{y}(0) = [0.4, 0.6, 0.8]^{\mathrm{T}}$ is used.

2.3 Functions. A system using synergetic computation can associate the initial state $\mathbf{x}(0)$ with the embedded pattern $\mathbf{v}_{m_0}$ ($m_0 \in \{1, 2, \ldots, M\}$) having the largest correlation to the initial state. This function is called the *image-associative function*. In order to execute this function, we set the coefficients α_m ($m = 1, 2, \ldots, M$) and γ in equation 2.1 to be $\alpha_m = 1$ and $\gamma = 1$, respectively. However, the coefficient β can be arbitrary as far as it is positive (Fuchs & Haken, 1988). The dynamics of the order parameter y_m is then given by

$$\frac{dy_m}{dt} = \left\{ 1 - \beta \sum_{m' \neq m} y_{m'}^2 - \sum_m y_m^2 \right\} y_m. \tag{2.10}$$

Figure 1 depicts the transitions of the order vector $\mathbf{y}$ in equation 2.10. It is observed that only one order parameter, y_{m_0} ($m_0 \in \{1, 2, \ldots, M\}$), converges to unity, and the other order parameters y_m ($m \neq m_0$) converge to zero. The *survival* order parameter y_{m_0} is related to embedded pattern $\mathbf{v}_{m_0}$, which has the largest initial value $y_{m_0}(0)$. The results in Figure 1 suggest that such systems execute the image-associative function depending not on the coefficients α_m, β, and γ, but on the initial order vector $\mathbf{y}(0)$.

Another function proposed in this article is a searching function for the image processing of complex systems. The searching function is executed by assuming the coefficients α_m and β in equation 2.1 to satisfy

$$\alpha_1 > \alpha_2 > \cdots > \alpha_M, \quad \beta = 1 - \gamma, \quad \gamma > 0.$$

From equation 2.11, the dynamics of the order parameters y_m is then given by

$$\frac{dy_m}{dt} = \left\{ \alpha_m - \sum_{m' \neq m} y_{m'}^2 - \gamma y_m^2 \right\} y_m \tag{2.11}$$

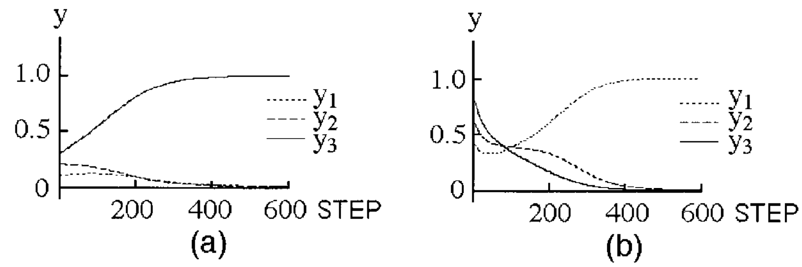

Figure 2: The transitions of order parameters y_m in the proposed searching function with $\gamma = 3$. (a) The parameters α_m are set to be [0.3, 0.1, 0.5] and the initial state is $\mathbf{y}(0) = [0.1, 0.2, 0.3]^{\mathrm{T}}$. (b) α_m are [0.5, 0.3, 0.1] and the initial state is $\mathbf{y}(0) = [0.4, 0.6, 0.8]^{\mathrm{T}}$.

Figure 2 shows the transitions of the order vector $\mathbf{y}$ in equation 2.12. Observe that the order parameter y_{m_0} survived there is related to the largest coefficient α_{m_0}, not to the largest initial value $y_{m_0}(0)$ as in Figure 1. The results reveal that the proposed searching function executes the image-associative function depending not only on the initial order vector $\mathbf{y}(0)$ but also on the coefficients α_m and γ. The problem of how to determine these coefficients to execute the desired function then needs to be solved. In the next section, we propose a learning mechanism to solve this problem.

3 The Learning Mechanism

3.1 Foundations of the Proposed Network. Suppose that a system consists of M components. Later we consider the order parameters as the system. The transition of the state of the system is governed by a stochastic mechanism. Let $\mathbf{X}(t) = [X_1(t), X_2(t), \ldots, X_M(t)]^{\mathrm{T}}$ in $\Re^M$ be the state vector of the system at time t, where $X_i(t)$ denotes a random variable representing the state of the ith component at time t. We assume that the dynamics of the system is given by the following stochastic differential equation (SDE),

$$d\mathbf{X}(t) = \mathbf{D}^{(r)}(\mathbf{X}(t))dt + \sigma(\mathbf{X}(t))d\mathbf{W}(t), \tag{3.1}$$

where $\mathbf{D}^{(r)}(\mathbf{x}) = [D_1^{(r)}(\mathbf{x}), D_2^{(r)}(\mathbf{x}), \ldots, D_M^{(r)}(\mathbf{x})]^{\mathrm{T}}$ in $\Re^M$ is the drift vector with polynomials $D_m^{(r)}(\mathbf{x})$ of order r, $\sigma(\mathbf{x})$ is a diffusion coefficient matrix, and $\mathbf{W}(t)$ is an M-dimensional standard Winner process. It is assumed that a unique strong solution to the SDE exists. The existence of such a solution is guaranteed under some condition on the drift vector and the diffusion matrix (Karatzas & Shreve, 1988).

Let $p(\mathbf{x}, t \mid \mathbf{x}_0, t_0)$ be the temporal change of the conditional probability density. We will denote

$$p_t(\mathbf{x}) = p(\mathbf{x}, t \mid \mathbf{x}_0, t_0).$$

Then the probability density function $p_t(\mathbf{x})$ satisfies the Fokker-Planck equation,

$$\frac{\partial p_t(\mathbf{x})}{\partial t} = -\sum_{i=1}^{M} \frac{\partial}{\partial x_i} \left\{ D_i^{(r)}(\mathbf{x}) p_t(\mathbf{x}) \right\} + \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} \frac{\partial^2}{\partial x_i \partial x_j} \{K_{ij}(\mathbf{x}) p_t(\mathbf{x})\}, \quad (3.2)$$

where

$$K_{ij}(\mathbf{x}) = \sum_k \sigma_{ik}(\mathbf{x}) \sigma_{jk}(\mathbf{x}).$$

Here $\sigma_{ij}(\mathbf{x})$ denotes the ijth component of the diffusion coefficient matrix $\sigma(\mathbf{x})$.

As a special case, suppose that the $K_{ij}(\mathbf{x})$ have the form

$$K_{ij}(\mathbf{x}) = K(\mathbf{x}) \delta_{ij}, \quad i, j = 1, 2, \ldots, M$$

for some function $K(\mathbf{x})$. Then equation 3.2 can be rewritten as

$$\frac{\partial p_t(\mathbf{x})}{\partial t} = -\sum_{i=1}^{M} \frac{\partial G_i(\mathbf{x})}{\partial x_i},$$

where $G_i(\mathbf{x})$ is defined by

$$G_i(\mathbf{x}) = D_i^{(r)}(\mathbf{x}) p_t(\mathbf{x}) - \frac{1}{2} \frac{\partial}{\partial x_i} \{K(\mathbf{x}) p_t(\mathbf{x})\}$$

and is called a probability current variable.

Let $p_{st}(\mathbf{x})$ denote the stationary probability density,

$$p_{st}(\mathbf{x}) = \lim_{t \to \infty} p_t(\mathbf{x}),$$

if it exists. Since the solution to the SDE 3.1 is Markovian, the stationary solution is unique if it exists (Movellan & McClelland, 1993). Suppose that the stationary probability density exists. Then it must be true that all the probability current variables $G_i(\mathbf{x})$ vanish, that is, the equations,

$$D_i^{(r)}(\mathbf{x}) p_{st}(\mathbf{x}) - \frac{1}{2} \frac{\partial}{\partial x_i} \{K(\mathbf{x}) p_{st}(\mathbf{x})\} = 0, \quad i = 1, 2, \ldots, M, \quad (3.3)$$

must hold. We expect that the stationary density $p_{st}(\mathbf{x})$ is of the form

$$p_{st}(\mathbf{x}) = \frac{C}{K(\mathbf{x})} \exp\{-V(\mathbf{x})\}, \quad (3.4)$$

where C is an additive constant and is determined from the normalization condition of the stationary probability density $p_{st}(\mathbf{x})$:

$$\int\int\cdots\int p_{st}(\mathbf{x})dx_1dx_2\cdots dx_M = 1.$$

The density function given by equation 3.4 is called the Boltzmann distribution and $V(\mathbf{x})$ the potential. Substituting equation 3.4 into 3.3 shows that the drift coefficients $D_i^{(r)}(\mathbf{x})$ must satisfy

$$D_i^{(r)}(\mathbf{x}) = -\frac{K(\mathbf{x})}{2}\frac{\partial V(\mathbf{x})}{\partial x_i}, \quad i = 1, 2, \ldots, M \tag{3.5}$$

The existence condition of such a potential $V(\mathbf{x})$ is given by

$$\frac{\partial}{\partial x_i}\left\{\frac{D_j^{(r)}(\mathbf{x})}{K(\mathbf{x})}\right\} = \frac{\partial}{\partial x_j}\left\{\frac{D_i^{(r)}(\mathbf{x})}{K(\mathbf{x})}\right\}, \quad i = 1, 2, \ldots, M. \tag{3.6}$$

If condition 3.6 is met for all i, then, from equation 3.5, the potential $V(\mathbf{x})$ can be obtained as

$$V(\mathbf{x}) = -\int_{\mathbf{a}}^{\mathbf{x}} \frac{1}{K(\mathbf{x}')}\sum_{i=1}^{M} D_i^{(r)}(\mathbf{x}')dx_i', \tag{3.7}$$

where $\mathbf{a}$ is a constant vector (Stratonovich, 1963).

3.2 An Extended Symmetric Diffusion Network. Consider a neural network consisting of M neurons whose dynamics is given by equation 3.1. We assume that the interactions among neurons inside the neural network are described by

$$D_i^{(r)}(\mathbf{x}) = w_i^{(1)} + \sum_{r'=2}^{r}\left\{\sum_{j_1}\sum_{j_2}\cdots\sum_{j_{(r'-1)}} w^{(r')}_{ij_1j_2,\ldots,j_{(r'-1)}} x_{j_1}x_{j_2},\ldots,x_{j_{(r'-1)}}\right\}, \tag{3.8}$$

where the weights $w^{(r)}_{k_1k_2,\ldots,k_r}$ are permutationally symmetric about $k_1, k_2, \ldots, k_r$. For example, the drift $D_i^{(r)}(\mathbf{x})$ with $r = 4$ is given by

$$\begin{aligned} D_i^{(4)}(\mathbf{x}) = w_i^{(1)} &+ \sum_{j_1} w_{ij_1}^{(2)}x_{j_1} + \sum_{j_1}\sum_{j_2} w_{ij_1j_2}^{(3)}x_{j_1}x_{j_2} \\ &+ \sum_{j_1}\sum_{j_2}\sum_{j_3} w_{ij_1j_2j_3}^{(4)}x_{j_1}x_{j_2}x_{j_3}. \end{aligned} \tag{3.9}$$

As for the external noise, we assume the independent gaussian noise, that is,

$$\sigma_{ij}(\mathbf{x}) = \sqrt{Q}\,\delta_{ij},$$

where Q is constant. It follows that the probability current variable is given by

$$G_i(\mathbf{x}) = D_i^{(r)}(\mathbf{x})p_t(\mathbf{x}) - \frac{Q}{2}\frac{\partial}{\partial x_i}p_t(\mathbf{x}), \quad i = 1, 2, \ldots, M.$$

We call this neural network an *extended symmetric diffusion network* (ESDN). The main difference between the ESDN and the ordinary symmetric diffusion networks is the definition of the drift $D_i^{(r)}(\mathbf{x})$.

The nonlinear drift $D_i^{(r)}(\mathbf{x})$ of the ESDN satisfies the existence condition 3.6 of potential $V(\mathbf{x})$ under the condition of permutational symmetry. It follows from equations 3.7 and 3.8 that the potential $V(\mathbf{x})$ can be written as

$$V(\mathbf{x}) = -\frac{2}{Q}\sum_{r'=1}^{r}\frac{1}{r'}\left\{\sum_{j_1}\sum_{j_2}\cdots\sum_{j_{r'}} w^{(r')}_{j_1 j_2,\ldots,j_{r'}} x_{j_1}x_{j_2},\ldots,x_{j_{r'}}\right\}, \qquad (3.10)$$

provided that the stationary probability density exists. Hence if the state vector $\mathbf{X}(t)$ evolved by equation 3.1 converges in distribution for given weights $w^{(r)}_{k_1k_2,\ldots,k_r}$ and a given constant Q, then the stationary probability density $p_{st}(\mathbf{x})$ is given by

$$p_{st}(\mathbf{x}) = Z_Q{}^{-1}\exp\{-V(\mathbf{x})\}, \qquad (3.11)$$

where Z_Q is the total partition function.

In order for $\mathbf{X}(t)$ to converge as $t \to \infty$, we need to assume a particular form of the drift function $D_i^{(r)}(\mathbf{x})$. Namely, we require $D_i^{(r)}(\mathbf{x})$ to satisfy some condition so that the state vector $\mathbf{X}(t)$ will not explode. This requirement holds, for example, if we assume that all the coefficients are zero other than $w_{ii}^{(2)}$ and $w_{iijj}^{(4)}$ and $w_{iijj}^{(4)}$ are negative in equation 3.9. In this case, we have

$$D_i^{(4)}(\mathbf{x}) = \left(w_{ii}^{(2)} + 3\sum_{j\neq i} w_{iijj}^{(4)}x_j^2 + w_{iiii}^{(4)}x_i^2\right)x_i, \qquad (3.12)$$

so that there exists some $B_1 > 0$ such that $D_i^{(4)}(\mathbf{x}) < 0$ for all $x_i > B_1$ while $D_i^{(4)}(\mathbf{x}) > 0$ for all $x_i < -B_1$. In the next section, we define

$$w_{ii}^{(2)} = \alpha_i, \quad w_{iijj}^{(4)} = -\frac{\beta+\gamma}{3} \quad (j \neq i), \quad w_{iiii}^{(4)} = -\gamma \qquad (3.13)$$

in order to relate the ESDN to the synergetic computer. Note that this definition of the coefficients makes the dynamics of the ESDN have the drift 3.12 being equal to equation 2.9.

When performing simulation experiments, we need to discretize the SDE 3.1 as follows:

$$\Delta X_i(t) = D_i^{(r)}(\mathbf{X}(t))\Delta t + \sqrt{Q}\,\Delta W_i(t), \tag{3.14}$$

where Δt denotes the length of the time step. This discretization is valid if the SDE has the unique strong solution. Note that in order to guarantee this mathematically, it is enough to assume that the drift vector is uniformly bounded. This is done if we modify the drift term $D_i^{(4)}(\mathbf{x})$ as

$$\widehat{D}_i^{(4)}(\mathbf{x}) = \max\left\{D_i^{(4)}(\mathbf{x}), K_i\right\}$$

for $\mathbf{x}$ outside the region $\mathbf{B}_2$ for some $K_i < 0$, where

$$\mathbf{B}_2 = \{\mathbf{x} = [x_1, x_2, \ldots, x_M]^{\mathrm{T}}\colon\ |x_i| \le B_2\}$$

for some $B_2 > B_1 > 0$. But if B_2 is sufficiently large, this modification does not contribute to the dynamics of the state vector $\mathbf{X}(t)$. Hence, for example, if the weights are defined as in equation 3.14, then the state vector $\mathbf{X}(t)$ satisfies all the properties we need. Now let $\xi_i(t)$ denote independent standard gaussian random variables. By assumption,

$$\Delta W_i(t) = \xi_i(t)\sqrt{\Delta t}.$$

It then follows that

$$X_i(t+\Delta t) = X_i(t) + D_i^{(r)}(\mathbf{X}(t))\Delta t + \sqrt{Q\Delta t}\,\xi_i(t). \tag{3.15}$$

In actual computation, we use equation 3.15 for the ESDN.

Before closing this section, we show the relationship between the ESDN with $r = 2$ and the discrete-state Boltzmann machine. Suppose that the Boltzmann machine consists of M neurons. We denote the state vector of the discrete-state Boltzmann machine by $\mathbf{u} = [u_1, u_2, \ldots, u_M]^{\mathrm{T}}$ where the outputs u_i take on values 0 or 1. The weight from the jth neuron to the ith neuron is denoted by w_{ij}; however, $w_{ii} = 0$ is assumed. The ith threshold is denoted by θ_i. The stationary probability mass function of the discrete-state Boltzmann machine with a positive temperature T is given by

$$p_T(\mathbf{u}) = {Z_T}^{-1}\exp\left(-\frac{E(\mathbf{u})}{T}\right),$$

where Z_T denotes the total partition function, and its energy function $E(\mathbf{u})$ is

$$E(\mathbf{u}) = -\frac{1}{2}\sum_i\sum_j w_{ij}u_iu_j + \sum_i \theta_i u_i.$$

Let $r = 2$ and $w_i^{(1)} = 0$ $(i = 1, 2, \ldots, M)$ in equation 3.10. For a vector $\mu = [\mu_1, \mu_2, \ldots, \mu_M]^{\mathrm{T}}$ in $\Re^M$, we substitute $\mathbf{x}-\mu$ in the place of $\mathbf{x}$ in equation 3.11. Then, defining the matrix $\mathbf{C}^{-1} = (C_{ij}^{-1})$ where

$$C_{ij}^{-1} = \frac{2w_{ij}^{(2)}}{Q}, \quad i, j = 1, 2, \ldots, M,$$

it follows that

$$p_{st}(\mathbf{x}) = {Z_Q}^{-1} \exp\{-(\mathbf{x}-\mu)^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{x}-\mu)\}.$$

The matrix $\mathbf{C} = (C_{ij})$ is the covariance matrix if the inverse of $\mathbf{C}^{-1}$ exists. Thus

$$p_T(\mathbf{u}) = {Z'}^{-1} F_{\mathbf{u}}\{p_{st}(\mathbf{x}+\mu)\},$$

where Z' denotes the total partition function and $F_{\mathbf{u}}\{\cdot\}$ is an operator to calculate the Fourier transformation,

$$F_{\mathbf{u}}\{p_{st}(\mathbf{x}+\mu)\} = \int \exp\{-j\mathbf{x}\cdot\mathbf{u}\}\, p_{st}(\mathbf{x}+\mu)d\mathbf{x} = \exp\left\{-\frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{C}\mathbf{u}\right\}.$$

This means that the weight w_{ij} and the threshold θ_i of the Boltzmann machine are proportional to the weight $w_{ij}^{(2)}$ of the ESDN, that is,

$$w_{ij} = C_{ij}, \qquad \theta_i = 2C_{ii}.$$

Hence, we have shown that the ESDN with $r = 2$ is reduced to the discrete-state Boltzmann machine.

4 Simulation Results

In this section, we demonstrate two applications of the ESDN: one for the estimation problem of an entire probability density function and the other for the searching function described in section 2.3 of the image processing executed by a synergetic computer.

4.1 Identification. We show that the ESDN has the ability to reconstruct the unknown potential. This ability is essential to design the coefficient values of the governing equations of a synergetic computer. We will use the ESDN with $r = 4$ for this purpose; we shall employ the definition 3.12. Then the potential (see equation 3.10) is given by

$$V(\mathbf{x}) = -\frac{2}{Q}\sum_{i=1}^{2}\left(w_i^{(1)}x_i + \frac{1}{2}w_{ii}^{(2)}x_i^2 + \frac{1}{4}\sum_{j=1}^{2} w_{iijj}^{(4)}x_i^2x_j^2\right),$$

and the data are generated on the basis of equation 3.11. In actual computation, we take the time step Δt equal to 0.05 in equation 3.15. The iteration

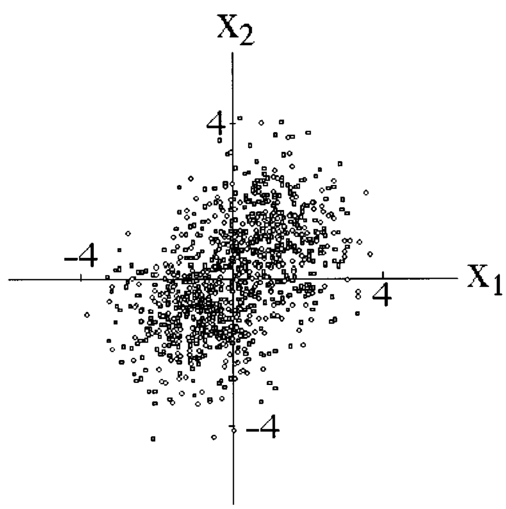

Figure 3: The probability density function as the learning data.

of learning is terminated if $\Delta X_i(t) < 0.01$ for all i. Figure 3 shows the probability density function to be learned. The density function possesses two clusters that are plotted on the two-dimensional plane $\Re^2$. The ESDN consisting of two neurons is used to reconstruct the unknown potential. An output of neuron is denoted by $\mathbf{x} = [x_1, x_2]^{\mathrm{T}}$. Figure 4a shows the outputs of the ESDN before learning. It is observed that the ESDN generates different outputs from the training data. In contrast, as shown in Figure 4b, the outputs of the ESDN after learning are quite similar to the presented data. The fitness to the unknown potential is confirmed by calculating the moments of the probability density function. Tables 1 and 2 list the mean vector, covariance matrix, and some higher moments of both the outputs of the ESDN and the presented data, where μ in $\Re^2$ is the mean vector and $m^{(r)}_{j_1 j_2,\ldots,j_r}$ is the rth moment. It is verified that the two distributions possess at least similar moments. The result reveals that the ESDN can acquire the external probability density function by learning. If the polynomial degree r becomes larger, some system combined with an ESDN may be able to produce a better approximation to the external density function. Such a problem is of interest and in progress. At this point, we note that the probability neural network (Streit & Luginbuhl, 1994) using a backpropagation-type learning algorithm, can also approximate the external probability density function. The learning algorithm used in such neural networks is similar to an expectation-maximizing algorithm (Dempster, Laird, & Rubin, 1977). It is known that the probability neural network has the capability of recon-

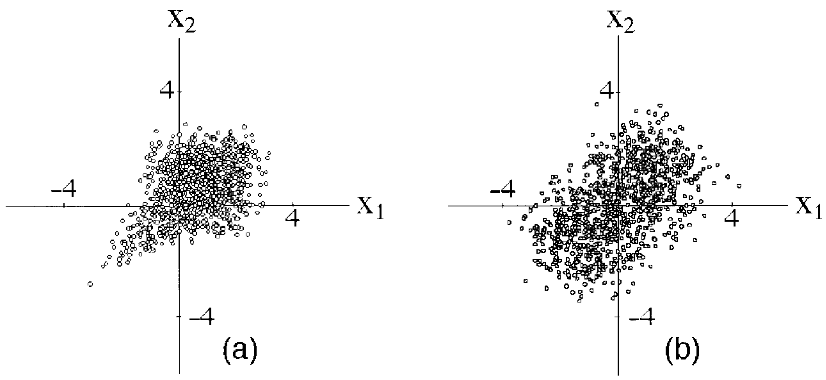

Figure 4: (a) Outputs of the ESDN before learning. (b) Outputs of the ESDN after learning.

Table 1: The Mean Vector and Covariance Matrix.

	$[\mu_1\ \mu_2]^T$	$\begin{bmatrix} m^{(2)}_{11} & m^{(2)}_{12} \\ m^{(2)}_{21} & m^{(2)}_{22} \end{bmatrix}$
Training data	$[0.00\ 0.01]^T$	$\begin{bmatrix} 1.90 & 0.94 \\ 0.94 & 2.02 \end{bmatrix}$
Outputs of the ESDN before learning	$[-1.42\ -1.42]^T$	$\begin{bmatrix} 1.01 & 0.43 \\ 0.43 & 1.02 \end{bmatrix}$
Outputs of the ESDN after learning	$[0.02\ 0.02]^T$	$\begin{bmatrix} 1.92 & 0.97 \\ 0.97 & 1.98 \end{bmatrix}$

structing the unknown potential; however, it is difficult to determine the number of necessary components. The ESDN can avoid this problem because the minimum number of necessary components is determined by the number of probability variables as a prior knowledge.

Table 2: Higher Moments.

	$m^{(3)}_{111}$	$m^{(3)}_{112}$	$m^{(3)}_{122}$	$m^{(3)}_{222}$	$m^{(4)}_{1111}$	$m^{(4)}_{1112}$	$m^{(4)}_{1122}$	$m^{(4)}_{1222}$	$m^{(4)}_{2222}$
Training data	0.10	0.03	0.15	0.31	9.32	3.50	3.68	3.99	11.11
Outputs before learning	−0.28	−0.41	−0.42	−0.26	2.66	1.50	1.79	1.45	2.56
Outputs after learning	0.07	0.02	0.16	0.35	9.39	3.57	3.63	3.89	10.41

Table 3: Values of the Degree of Impression λ_m.

Parameters	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7
Values	1.0	0.9	0.8	0.7	0.6	0.5	0.4

4.2 Searching Functions. In this subsection, we consider the image processing that associates the initial image to different embedded patterns by adjusting the coefficients of equation 2.1. For this purpose, we employ the ESDN consisting of three neurons with $r = 4$ and the coefficients defined in equation 3.13. Then the synergetic computer satisfies the proposed equation 2.11 for the searching function, and the coefficients α_m and γ are designed by learning of the ESDN. However, the condition for the coefficient β to achieve the searching function implies that the weight $w_{iijj}^{(4)}$ with $i \neq j$ is 1/3. Three fingerprints are used as the embedded patterns ($\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$). To associate the embedded pattern $\mathbf{v}_1$, for example, we give the vector $[1, 0, 0]^{\mathrm{T}}$ as the training signal. The ESDN learns the parameters α_m ($m = 1, 2, 3$) and γ of equation 2.1 for each pattern. Figure 5 shows that the three different embedded patterns are associated with the same random initial image. Such image processing cannot be executed by the usual associative function, which depends only on the initial state. The embedded patterns consist of 256×256 pixels, each pixel being painted by classifying 256 levels from 0 (white) to 1 (black). The noise vector $\mathbf{z}$, which lies orthogonal to the space spanned by the embedded patterns, changes according to time. Figure 6 shows the behavior of the noise vector $\mathbf{z}$ when the embedded pattern $\mathbf{v}_2$ is associated by the synergetic computer.

Finally, we provide another searching function. Figure 7 shows the embedded pattern that combines seven embedded patterns $\mathbf{v}_m$ ($m = 1, 2, \ldots, 7$) linearly. Each embedded pattern is represented by one person. The embedded patterns consist of 256×256 pixels, and each pixel takes on values between 0 and 256. The values 0 and 256 describe white and black, respectively. Let λ_m denote the degree of impression for person m. The impression values are listed in Table 3. The degree of impression to the person corresponding to the embedded pattern $\mathbf{v}_7$ is weaker than one-half of that to the person corresponding to the embedded pattern $\mathbf{v}_1$. The difference in the degrees of impression ($\lambda_1 > \lambda_2 > \cdots > \lambda_7$) means the difference in the information of the hierarchy among the embedded patterns. The coefficient β can be set to be $1 - \gamma$. In order to execute the searching function, the coefficient γ is given by 1.6. Figure 8a shows the associative pattern of the synergetic computer for $\gamma = 1.6$, and Figure 8b shows that for $\gamma = 0.5$. The input pattern is offered by a noisy pattern in both cases. In this demonstration, it is shown that the searching function of the synergetic computer can associate the initial pattern with the embedded patterns, depending on the degree of impression.

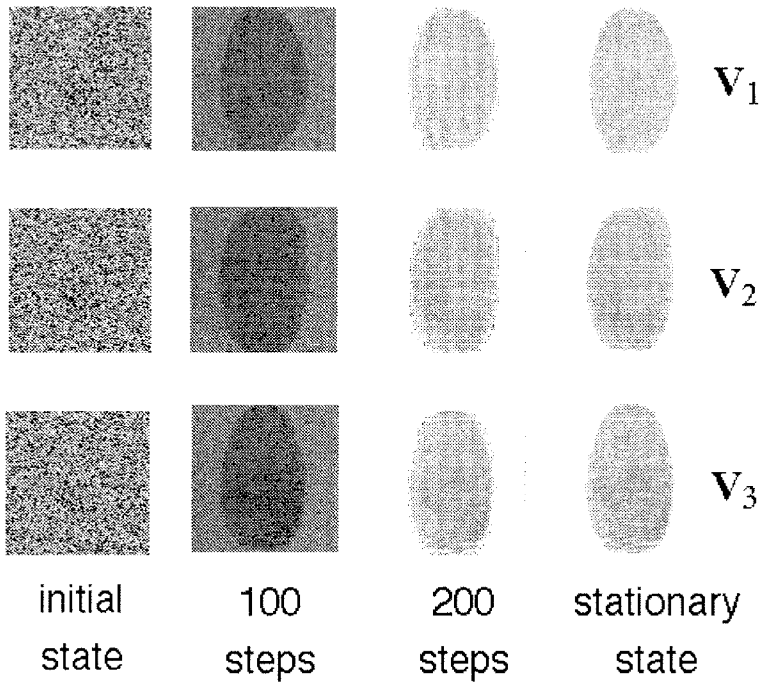

Figure 5: Associative processes of the embedded patterns. The initial state is the same random pattern.

5 Conclusion

In this article, we proposed a searching function of the image processing for complex systems using synergetic computation. For the implementation of the searching function, the coefficient values of governing equations of the system need to be determined. We propose for this purpose an extended symmetric diffusion network that can learn the dynamics of the system derived from the nonlinear potential. As a special case, the continuous-state ESDN can be translated to the discrete-state Boltzmann machine.

The purpose of this article is to show the application of the ESDN to a synergetic computer with the searching function in addition to the usual image-associative function. The weights of the ESDN and the coefficients of the synergetic computer are connected to achieve the desired property. In simulation results, the basic property of the ESDN is shown through the estimation problem of an entire probability density function. The result reveals that the ESDN can acquire the external probability density function success-

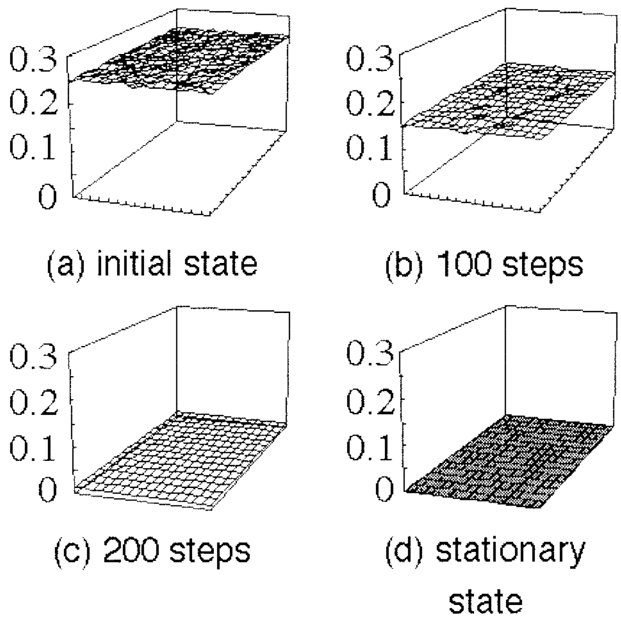

Figure 6: Transition of the noise in the associative processes. The noise is removed at the stationary state.

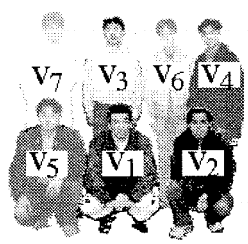

Figure 7: Embedded patterns and the hierarchy among them for the searching function.

fully. Finally, we demonstrated an application of the searching function to complex systems by using the proposed learning mechanism of the ESDN.

Acknowledgments

We thank an anonymous referee for many helpful suggestions and comments, which greatly improved the presentation of this study.

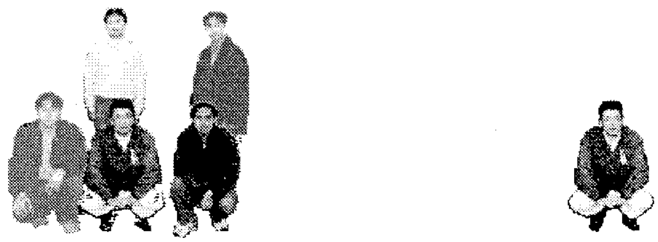

Figure 8: (a) Associative pattern ($\gamma = 1.6$). (b) Associative pattern ($\gamma = 0.5$).

References

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum-likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Soc. Ser., B39*, 1–38.

Feidlin, M. I., & Wentzell, A. D. (1984). *Random perturbations of dynamical systems*. Berlin: Springer-Verlag.

Fuchs, A., & Haken, H. (1988). Pattern recognition and associative memory as dynamical processes in a synergetic system I. *Biol. Cybern., 60*, 17–22.

Gutzmann, K. (1987). Combinatorial optimization using a continuous state Boltzmann machines. In *IEEE First International Conference on Neural Networks (San Diego 1987)* (Vol. 3, pp. 721–734). New York: IEEE.

Haken, H. (1989). *Synergetics: An introduction, nonequilibrium phase transitions and self-organization in physics, chemistry and biology*. Berlin: Springer-Verlag.

Haken, H. (1990). *Synergetic computers and cognition*. Berlin: Springer-Verlag.

Haken, H., Haas, R., & Banzhaf, W. (1989). A new learning algorithm for synergetic computers. *Biol. Cybern., 62*, 107–111.

Karatzas, I., & Shreve, S. E. (1988). *Brownian motion and stochastic calculus*. New York: Springer-Verlag.

Kohonen, T., Barna, G., & Chrisley, R. (1988). Statistical pattern recognition with neural networks: Benchmarking studies. In *IEEE International Conference on Neural Networks (San Diego 1988)* (Vol. 1, pp. 61–68). New York: IEEE.

Lippmann, R. P. (1989). Review of neural networks for speech recognition. *Neural Computation, 1*, 1–38.

Movellan, J. R., & McClelland, J. L. (1993). Learning continuous probability distributions with symmetric diffusion networks. *Cognitive Sci., 17*, 463–496.

Movellan, J. R., & McClelland, J. L. (1994). Covariance learning rules for stochastic neural networks. In *Proceedings of the World Congress on Neural Networks* (pp. 376–381).

Stratonovich, R. L. (1963). *Topics in the theory of random noise*. New York: Gordon and Breach.

Streit, R. L., & Luginbuhl, T. E. (1994). Maximum likelihood training of probabilistic neural networks. *IEEE Trans. NN, 5*, 764–783.

Received November 19, 1996; accepted April 28, 1998.