

# Image sequence correspondence via a Hopfield neural network

## Jyh-Yeong Chang

National Chiao Tung University  
Department of Control Engineering  
1001 Ta-Hsueh Road  
Hsinchu 300, Taiwan

## Shin-Wen Lee

Industrial Technology Research Institute  
Advanced Technology Center  
Computer and Communication Research  
Laboratory  
195 Chung-Hsing Road Section 4  
Chutung 310, Taiwan

## Mong-Fong Horng

National Chiao Tung University  
Department of Control Engineering  
1001 Ta-Hsueh Road  
Hsinchu 300, Taiwan

**Abstract.** A neural network approach to finding trajectories of feature points in a monocular image sequence is proposed. In conventional methods, this problem is formulated as an optimization problem and solved using heuristic algorithms. The problem usually involves lengthy computations, making it computationally difficult. We apply the Hopfield neural network to image sequence correspondence. The design and development of the Lyapunov function for this problem are discussed in detail. Furthermore, the neural-network-based image correspondence scheme is extended to the case of successive image frames, in which some feature points are allowed to be occluded. Examples and simulation results are presented to illustrate the design process and the convergence characteristics of the proposed neural network. By using the massive parallel-processing power of neural networks, a real-time and accurate solution can be obtained.

*Subject terms:* visual communication; image sequence correspondence; Hopfield neural model; image tracking; neural network structure.

*Optical Engineering* 32(7), 1531–1538 (July 1993).

## 1 Introduction

Finding trajectories of moving objects in a monocular image sequence is a vital technique in the field of motion analysis. A major step in finding such trajectories is to identify images of the same physical point in a sequence of frames; this step is usually called the correspondence problem. This technique, which successively refines the structure of the object as more frames are acquired, frees us from assumptions of rigidity and relies on natural assumptions about motion characteristics. Many correspondence algorithms<sup>1–4</sup> have been introduced in the past few years. Most research for establishing correspondence<sup>2,3</sup> uses only two frames of a sequence to solve this problem. However, two-frame algorithms require assumptions about the nature of the objects. Recently, a correspondence algorithm based on a sequence of frames was proposed by Sethi and Jain.<sup>1</sup> In their method, smoothness of motion is used to relax the need for assumptions about the rigidity of the object. The path coherence function is used to formulate the correspondence problem as an optimization problem. For instance, if we are given a sequence of  $m$  frames that have  $n$  feature points each, then basically there will be  $C(n^m, n)$  solutions of combinations, i.e.,  $n$  trajectories from

$n^m$  possible trajectories. After comparing measurements of motion smoothness and velocity continuity calculated from the defined coherence functions for each solution, optimal  $n$  trajectories can be obtained. However, a large amount of computation time is usually necessary to identify the optimal solution with  $n$  true paths from among all legal solutions. To expand the computation power applied to this problem, we shall introduce a Hopfield neural network algorithm<sup>5,6</sup> for finding trajectories of feature points in a monocular image sequence. The parallel nature of the neural network matches nicely the high-speed computation requirements of the image correspondence problem.

The Hopfield neural network has been used in a wide range of applications. Most engineering optimization problems, including image processing algorithms, can be solved using the minimum energy searching property of the Hopfield model.<sup>7–12</sup> When the neural network is used to solve an optimization problem, the problem is usually formulated as minimization of a constrained cost function, where all the constraints on the solution can be explicitly incorporated into the cost function. For example, an image restoration technique<sup>10</sup> has been developed where the cost function is minimized by the Hopfield network. Using the Hopfield network to match the subgraph features for object recognition<sup>11</sup> has also been discussed. Recently, the use of a Hopfield network for the feature point matching of left and right images<sup>12</sup> has also been reported. Note that this stereo vision

Paper VCI-17 received Dec. 2, 1992; revised manuscript received Mar. 1, 1993; accepted for publication Mar. 3, 1993.  
© 1993 Society of Photo-Optical Instrumentation Engineers. 0091-3286/93/\$2.00.

correspondence task is quite different from our image sequence correspondence, i.e., trajectory-finding, problem.

## 2 Smoothness of Motion

It is generally believed that, due to inertia, the motion of a physical object cannot be changed abruptly. If a frame sequence is acquired with enough sampling rate, the trajectories of moving objects in a sequence of images are smooth. Thus, in a real-world analysis of dynamic scenes, it seems reasonable to assume that the motion of an object is smooth at any instant of time.<sup>1</sup>

Let  $T_i$  represent one of the  $n^m$  possible trajectories and let  $X_{i,k}$  be the coordinate of the  $i_k$ 'th feature point on this chosen trajectory  $T_i$  in the  $k$ 'th frame of  $m$  frames. Then one possible trajectory  $T_i$  can be represented by

$$T_i = [X_{i_1,1}, X_{i_2,2}, \dots, X_{i_m,m}] \quad (1)$$

Let the deviation  $d_k$  in the path of the point in the  $k$ 'th frame be measured by the path coherence function  $\phi$ , as follows:

$$d_k = \phi(\overrightarrow{X_{i_{k-1},k-1} X_{i_k,k}}, \overrightarrow{X_{i_k,k} X_{i_{k+1},k+1}}), \quad (2)$$

where  $\overrightarrow{X_{i_{k-1},k-1} X_{i_k,k}}$  is the displacement vector of the  $i_k$ 'th feature point in the  $k$ 'th frame, and the path coherence function is given by

$$\phi(\overrightarrow{v_1}, \overrightarrow{v_2}) = w_1 \left( 1 - \frac{\overrightarrow{v_1} \cdot \overrightarrow{v_2}}{\|\overrightarrow{v_1}\| \|\overrightarrow{v_2}\|} \right) + w_2 \left( 1 - \frac{2[\|\overrightarrow{v_1}\| \|\overrightarrow{v_2}\|]^{1/2}}{\|\overrightarrow{v_1}\| + \|\overrightarrow{v_2}\|} \right), \quad (3)$$

where  $\overrightarrow{v_1} \cdot \overrightarrow{v_2}$  is the inner product of two vectors.

Path coherence function  $\phi$  contains two terms: directional coherence and speed coherence, and  $w_1$  and  $w_2$  are the relative weights for these two terms, respectively. Note that function  $\phi$  accounts for the direction and speed continuity during the displacement of the feature point in an image sequence but does not take acceleration continuity into consideration. We are given  $n$  feature points in each of  $m$  successive frames, and  $n$  trajectories must be determined. The value  $D_t(i_1, i_2, \dots, i_m)$ , representing the measurement of continuity for any chosen trajectory  $t$ , is given by

$$D_t(i_1, i_2, \dots, i_m) = \sum_{k=2}^{m-1} d_k \quad (4)$$

In a scene, several objects may be undergoing random motion. If there are  $n$  points in a sequence of  $m$  frames,  $n$  trajectories will result, and the deviation of all  $n$  trajectories is given by

$$\Phi = \sum_{t=1}^n D_t(i_1, i_2, \dots, i_m), \quad (5)$$

where  $t$  is the index of the paths.

The problem we now face is how to determine the correspondence so that the total deviation of these resulting trajectories will be minimal. This has been formulated by Sethi and Jain<sup>1</sup> as an optimization problem, in which  $\Phi$  is mini-

mized by faster variants of conventional optimization algorithms. Below, we apply the Hopfield neural network model to solve this optimization problem. The following sections treat two cases: the first is processing of an image sequence without acquiring new frames and the second is finding correspondences with frames coming in continuously.

## 3 Finite Image Sequence Correspondence Using The Hopfield Model

To begin this section, we briefly review the Hopfield network. The Hopfield model is a recurrent neural network consisting of  $N$  mutually interconnected processing units, called *neurons*, whose states  $v_i$  are taken as their outputs. The  $i$ 'th neuron state of  $v_i$  is changed according to the following equation:

$$u_i = \sum_{j=1}^N W_{ij} v_j + I_i, \quad (6)$$

where  $W_{ij}$  is the connection weight from node  $j$  to node  $i$  and  $I_i$  is the bias input to the  $i$ 'th neuron. Furthermore,

$$v_i = f(u_i), \quad (7)$$

where the nonlinear function  $f(\cdot)$  is the so-called sigmoid function, which has the following form:

$$f(x) = \frac{1}{1 + e^{-\lambda x}}, \quad (8)$$

in which  $\lambda$  denotes the gain of the sigmoid function. This function takes the value one as  $x$  approaches  $+\infty$  and zero as  $x$  approaches  $-\infty$ . As  $\lambda$  increases,  $f(\cdot)$  approaches a step function. Consider the following energy function:

$$E = -\frac{1}{2} \sum_i \sum_j W_{ij} v_i v_j - \sum_i I_i v_i \quad (9)$$

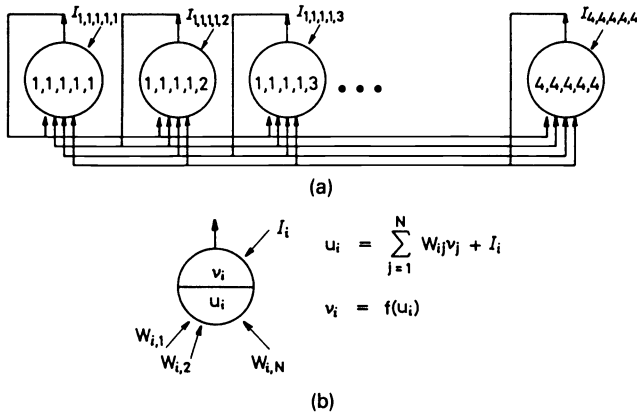
Minimal energy convergence has been proven<sup>5</sup> if  $W_{ij} = W_{ji}$  and  $W_{ii} = 0$  because

$$\Delta E = -\Delta v \left( \sum_i W_{ij} v_j + I_i \right) \leq 0 \quad (10)$$

Hence, the evolution of the dynamic system will seek the minima of the energy surface  $E$ . Moreover, if Eq. (9) is a local positive definite function around some neighborhood of the equilibrium points of the system, then Eq. (9) is a valid Lyapunov function to prove that the equilibrium point is stable.

We are now in a position to introduce the Hopfield network into the image sequence correspondence problems. If a dynamic scene is sampled at a proper sampling rate so as to capture all significant events in the scene and there are  $n$  points in each of  $m$  successive frames,  $n$  trajectories will result. For purposes of our application, we use a neural network containing  $n^m$  interconnected neurons, where  $n$  is the number of feature points in one frame and  $m$  is the number of image frames. Let  $V = \{v_{i_1, i_2, \dots, i_m} \mid 1 \leq i_1, i_2, \dots, i_m \leq n\}$  be a binary state set of the neural network denoting the state of the  $(i_1, i_2, \dots, i_m)$ 'th neuron.

Each possible path  $T_i = [X_{i_1,1}, X_{i_2,2}, \dots, X_{i_m,m}]$  is associated with an activation value of the neuron  $v_{i_1, i_2, \dots, i_m}$ . A value of



**Fig. 1** (a) Hopfield neural network for finding trajectories of four feature points with five image planes and (b) the function of each neuron.

one stands for a trajectory; zero means not a trajectory. A neural network of this type is shown in Fig. 1, for the case of  $n = 4, m = 5$ .

We construct an error function, Eq. (11), for our problem as follows:

$$\begin{aligned}
 E = & \left( \sum_{i_1=1}^n \sum_{i_2=1}^n \dots \sum_{i_m=1}^n v_{i_1, i_2, \dots, i_m} D_f(i_1, i_2, \dots, i_m) \right)^2 \\
 & + k \left[ \left( \sum_{f_1=1}^n \left( \sum_{i_2=1}^n \sum_{i_3=1}^n \dots \sum_{i_m=1}^n v_{f_1, i_2, \dots, i_m} - 1 \right) \right)^2 \right. \\
 & + \left( \sum_{f_2=1}^n \left( \sum_{i_1=1}^n \sum_{i_3=1}^n \dots \sum_{i_m=1}^n v_{i_1, f_2, \dots, i_m} - 1 \right) \right)^2 \\
 & \vdots \\
 & \left. + \left( \sum_{f_m=1}^n \left( \sum_{i_1=1}^n \sum_{i_2=1}^n \dots \sum_{i_{m-1}=1}^n v_{i_1, i_2, \dots, f_m} - 1 \right) \right)^2 \right] \\
 & + \sum_{i_1=1}^n \sum_{i_2=1}^n \dots \sum_{i_m=1}^n (mk + D_f^2(i_1, i_2, \dots, i_m)) \\
 & \times v_{i_1, i_2, \dots, i_m} (1 - v_{i_1, i_2, \dots, i_m}) . \tag{11}
 \end{aligned}$$

The purpose of the first term in Eq. (11) is to minimize the error of the set of all currently activated paths. To choose  $n$  correct trajectories, we make use of the terms in the bracket to inhibit illegal correspondences at each frame. Consider the first index  $f_1$  of the  $m$  neuron dimensions. The term

$$\left( \sum_{i_2=1}^n \sum_{i_3=1}^n \dots \sum_{i_m=1}^n v_{f_1, i_2, \dots, i_m} - 1 \right)^2 \tag{12}$$

for  $f_1 = 1, 2, \dots, n$  will be zero when the feature point  $f_1$  of frame 1 appears in only one of  $n$  trajectories; otherwise, this term will introduce a large error proportional to the constant  $k$ . In the other dimensions, similar lines of reasoning apply. Hence, when the energy of Eq. (11) reaches the minimum, each feature point will appear in one and only one trajectory

and then  $n$  trajectories can be obtained by adjusting the inhibiting constant  $k$  appropriately. It is easy to see that the terms in the bracket will be zero, the minimal value, only when all feature points in the same frame are assigned to distinct trajectories and no feature point is missing. The very last term of Eq. (11) is introduced to cancel the self-feedback connections for the consideration of stability; as stated in Eq. (10), the diagonal elements of the connection weighting matrix should be zero. In constructing the energy function for the Hopfield neural network, we can see that the energy function is a local positive definite function satisfying the requirement mentioned above. Therefore, the system dynamics will seek the minimum of the defined energy surface.<sup>5</sup>

Expanding Eq. (11), we obtain

$$\begin{aligned}
 E = & -\frac{1}{2} \sum_{i_1=1}^n \sum_{i_2=1}^n \dots \sum_{i_m=1}^n \sum_{j_1=1}^n \sum_{j_2=1}^n \dots \sum_{j_m=1}^n \\
 & \times (-2)(ck + D_f(i_1, i_2, \dots, i_m)) \\
 & \cdot D_f(j_1, j_2, \dots, j_m) \cdot v_{i_1, i_2, \dots, i_m} v_{j_1, j_2, \dots, j_m} \\
 & - \sum_{i_1=1}^n \sum_{i_2=1}^n \dots \sum_{i_m=1}^n (mk - D_f^2(i_1, i_2, \dots, i_m)) \cdot v_{i_1, i_2, \dots, i_m} \\
 & - \sum_{i_1=1}^n \sum_{i_2=1}^n \dots \sum_{i_m=1}^n (mk + D_f^2(i_1, i_2, \dots, i_m)) \cdot v_{i_1, i_2, \dots, i_m}^2 \\
 & + mnk , \tag{13}
 \end{aligned}$$

where

$$c = eql(i_1, j_1) + eql(i_2, j_2) + \dots + eql(i_m, j_m) , \tag{14}$$

$$eql(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases} .$$

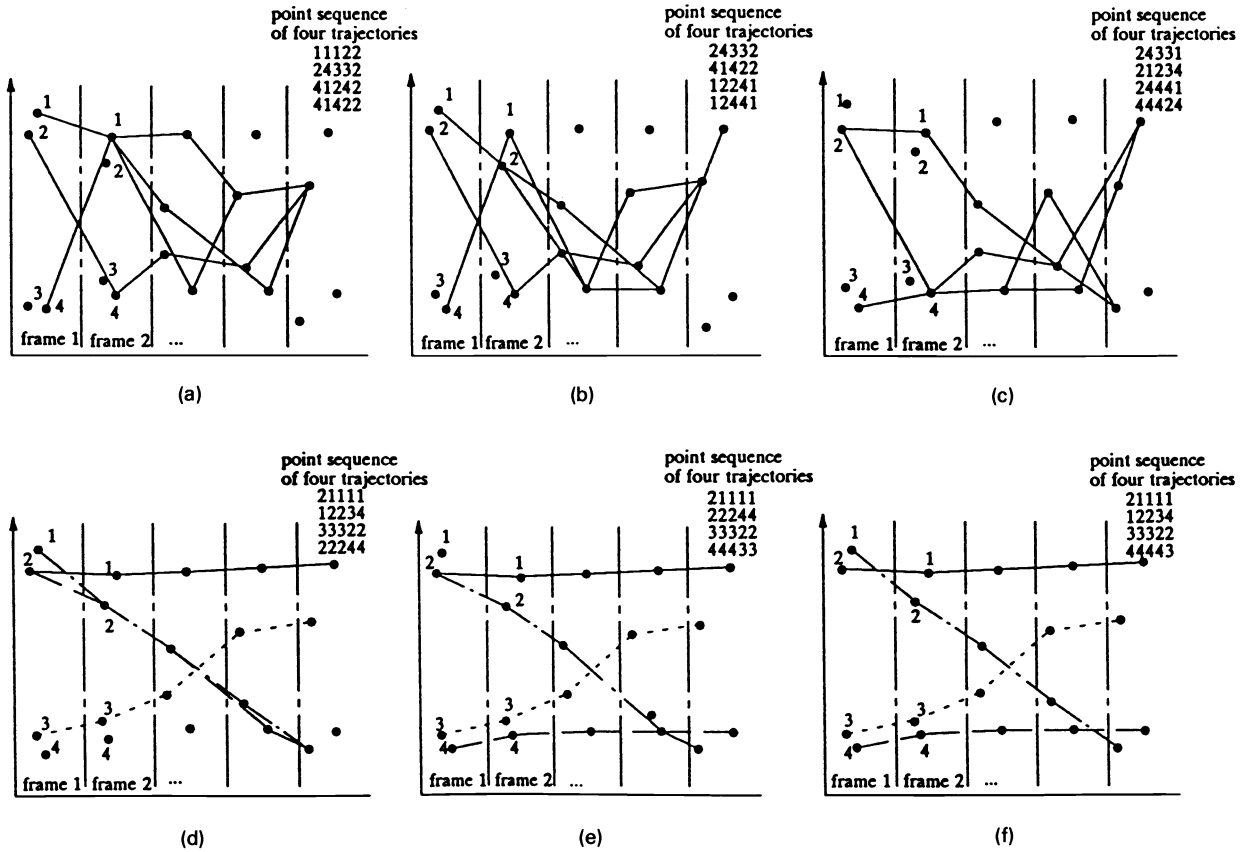
Comparing Eq. (13) with Eq. (9) and ignoring the constant term, we have

$$\begin{aligned}
 W_{i_1, i_2, \dots, i_m; j_1, j_2, \dots, j_m} = & \\
 \begin{cases} -2(ck + D_f(i_1, i_2, \dots, i_m) D_f(j_1, j_2, \dots, j_m)) & \text{if } (i_1, i_2, \dots, i_m) \neq (j_1, j_2, \dots, j_m) \\ 0 & \text{if } (i_1, i_2, \dots, i_m) = (j_1, j_2, \dots, j_m) \end{cases} \tag{15}
 \end{aligned}$$

and

$$I_{i_1, i_2, \dots, i_m} = mk - D_f^2(i_1, i_2, \dots, i_m) . \tag{16}$$

From Eq. (15), one can see that the interconnection weights are determined by the path coherence function and the constant  $k$ . Hence,  $T_{i_1, i_2, \dots, i_m; j_1, j_2, \dots, j_m}$  and  $I_{i_1, i_2, \dots, i_m}$  can be computed without error from the coordinates of objects within a frame. The state of the neural network will be stable after a certain number of evolutionary iterations, according to Eq. (6). Then  $n$  neurons with an activation value of one will be obtained. The proposed algorithm is illustrated by the following example. Note that feature point correspondence of synthetic and real images by minimizing the path coherence function has been tested and justified.<sup>1</sup> In light of this success, only synthetic images are analyzed in this paper.



**Fig. 2** Searching evolution of the neural network: (a) initial trajectories, (b) trajectories after 500 iterations, (c) trajectories after 1000 iterations, (d) trajectories after 1500 iterations, (e) trajectories after 2000 iterations, and (f) the completed correspondence.

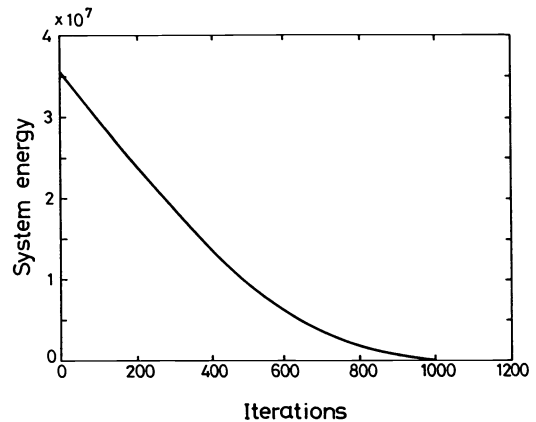
*Finite image frame example.* We tried to find the trajectories of four moving targets from five successive frames. According to Eqs. (15) and (16), the weights of interconnection and the external inputs corresponding to each neuron were obtained as follows:

$$W_{i_1, i_2, \dots, i_5; j_1, j_2, \dots, j_5} = \begin{cases} -2(ck + D_r(i_1, i_2, \dots, i_5)D_r(j_1, j_2, \dots, j_5)) & \text{if } (i_1, i_2, \dots, i_5) \neq (j_1, j_2, \dots, j_5) \\ 0 & \text{if } (i_1, i_2, \dots, i_5) = (j_1, j_2, \dots, j_5) \end{cases} \quad (17)$$

and

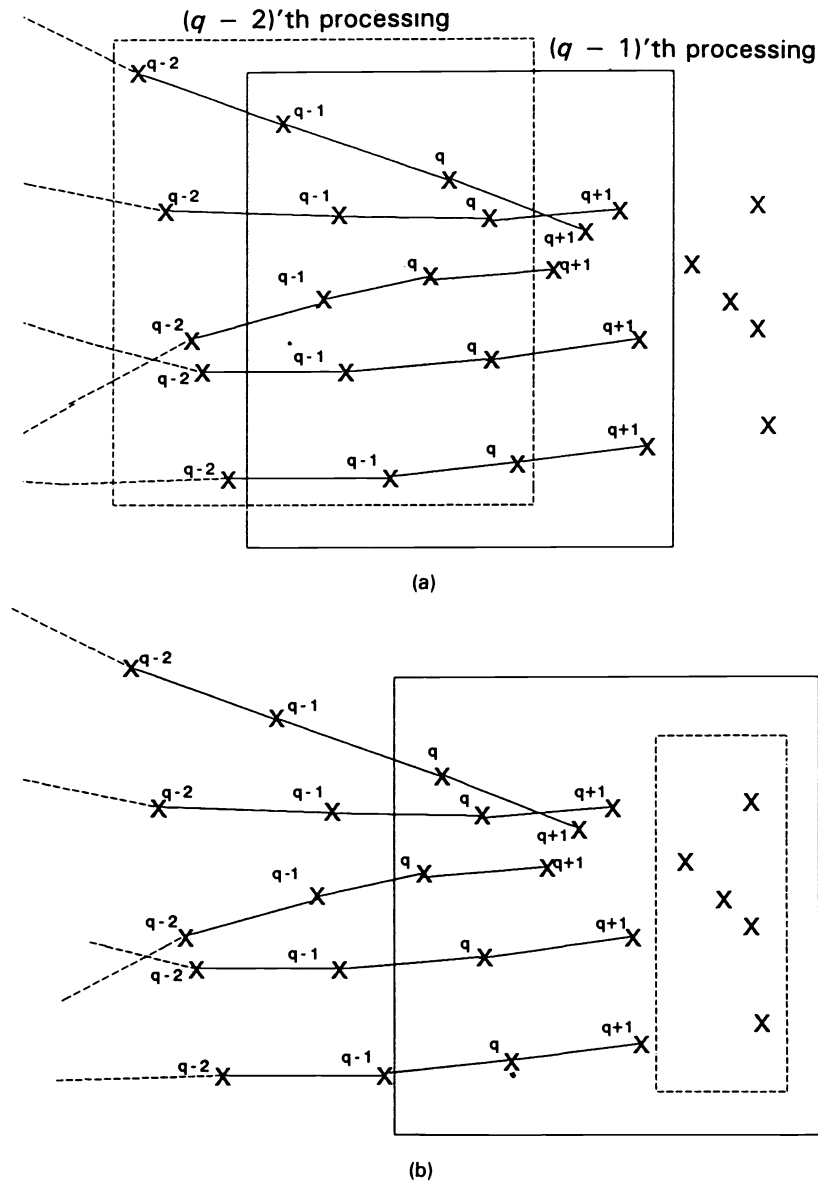
$$I_{i_1, i_2, \dots, i_5} = 5k - D_r^2(i_1, i_2, i_3, i_4, i_5) \quad (18)$$

Since the network constructed was symmetric and zero diagonal, the algorithm could be expected to be stable. The four nodes of the stabilized network that have maximum value will be chosen as the optimal solution. By adjusting the inhibiting constant  $k$  of Eq. (11), we were able to obtain the correct solution. In our experiment, we generated four feature points of objects in space. The objects were exhibiting different rational motions. The experiment was carried out under the following parameter assignments: the sigmoid function constant  $\lambda = 5$ , the inhibiting constant  $k = 15$ , and



**Fig. 3** The evolution of monotone decreasing energy.

the coherence function constant  $w_1 = 70$  and  $w_2 = 30$ . The initial trajectories, i.e., the active states of the neural network, were given randomly, as shown in Fig. 2(a). After  $\sim 2500$  iterations of updating the nodes, the trajectories of the solution were found. The evolution of the searching is shown in Fig. 2. The evolution of monotone decreasing energy is shown in Fig. 3.



**Fig. 4** (a) An image sequence is divided into sets of successive frames to be processed and the result of the previous frames is used to guide the correspondence of the current frame, and (b) the newly acquired points and the point set to be processed are indicated in the dash-line box and the solid-line box, respectively.

#### 4 Correspondence of Image Sequence with Successive Input Frames

So far, the proposed correspondence algorithm has worked successfully. That the whole system will possess  $n^m$  neurons and  $(n^m)^2 - n^m$  interconnections in the case of  $n$  trajectories in  $m$  frames is shown. Thus, the components and computation of iterations will increase exponentially as the number of image frames  $m$  increases. This makes it troublesome to exploit the above algorithm directly.

In this section we propose a modified image sequence correspondence algorithm, which will lead to a faster and more flexible solution for the case of successive input frames. On receiving the first three frames, the algorithm establishes the correspondences of these three frames, as described in

the previous section. When the fourth image frame is received, the algorithm takes advantage of the correspondence relations obtained in the previous image frames by deleting the first frame, shifting the second and third frames to be the first and second frames of a new set of frames, and taking the image just received to be the third frame of the new set. The correspondence is complete if the correspondence relation for the second and third images of the set is constructed, which determines the execution cycle of the approach. This operation is like an *execution window* that covers three frames at a time instant, as shown in Fig. 4(a). Proceeding in this fashion, every time we have a new image frame, we will invoke an execution cycle. In this way, during each time instant the neural network will deal with the correspondences

between the new second and third frames of three image sequences. Hence, the network components and computational cost can be reduced dramatically.

Let us now explain this concept in more detail. As mentioned above, the first three frames will be processed by the algorithm in Sec. 3. Once the initial correspondence of the first three frames is constructed, we will have the coordinates of  $n$  constructed trajectories, as follows:

$$\begin{aligned} T_1 &= [X'_{1,1}, X'_{1,2}, X'_{1,3}] \\ T_2 &= [X'_{2,1}, X'_{2,2}, X'_{2,3}] \\ &\vdots \\ T_n &= [X'_{n,1}, X'_{n,2}, X'_{n,3}] \end{aligned} \quad (19)$$

where  $X'_{p,k}$  is the computed coordinate of the feature point of the  $k$ 'th frame on the  $p$ 'th trajectory.

A newly acquired frame together with the previous two images forms a new frame set, as shown in Fig. 4(b). We are required to determine how the feature points of the new frame should be assigned to the constructed trajectories. By using the correspondence relation of the two previous frames, we can reduce the complexity of the neural network. A neural network for dealing with the correspondences of three frames, in which the correspondences of the first two are known, is presented below.

We define a new binary state set of the neural network,  $V = \{v_{p,i} | 1 \leq p, i \leq n\}$ , where  $v_{p,i}$  denotes the state of the  $(p, i)$ 'th neuron. If the  $i$ 'th point in the new frame is associated with the  $p$ 'th path, then activation value one is assigned to neuron  $v_{p,i}$ ; otherwise, zero is assigned. Thus, we can formulate the correspondence problem between the new frame and the previous two frames as an optimization problem similar to that examined earlier. For the currently processed frame set numbered  $(q - 2, q - 1, q)$ , we will have

$$\begin{aligned} E &= \left( \sum_{p=1}^n \sum_{i=1}^n v_{p,i} D'_i(p, i, q) \right)^2 \\ &+ k \left[ \sum_{f_1=1}^n \left( \sum_{i=1}^n v_{f_1,i} - 1 \right)^2 + \sum_{f_2=1}^n \left( \sum_{i=1}^n v_{f_2,i} - 1 \right)^2 \right] \\ &+ \sum_{p=1}^n \sum_{i=1}^n (2k + D_i'^2(p, i, q)) v_{p,i} (1 - v_{p,i}) \end{aligned} \quad (20)$$

where

$$D'_i(p, i, q) = \Phi(\overrightarrow{X'_{p,q-2} X'_{p,q-1}}, \overrightarrow{X'_{p,q-1} X_{i,q}}) \quad (21)$$

for  $q = 4, 5, \dots$

Expanding Eq. (20), we obtain

$$\begin{aligned} E &= -\frac{1}{2} \sum_{p_1=1}^n \sum_{i_1=1}^n \sum_{p_2=1}^n \sum_{i_2=1}^n (-2)(c'k + D'_i(p_1, i_1, q) \\ &\cdot D'_i(p_2, i_2, q)) v_{p_1, i_1} v_{p_2, i_2} \\ &- \sum_{p_1=1}^n \sum_{i_1=1}^n (2k - D_i'^2(p_1, i_1, q)) v_{p_1, i_1} \end{aligned}$$

$$- \sum_{p_1=1}^n \sum_{i_1=1}^n (2k + D_i'^2(p_1, i_1, q)) \cdot v_{p_1, i_1}^2 + 2nk \quad (22)$$

and

$$c' = eql(p_1, p_2) + eql(i_1, i_2) \quad (23)$$

Then the weights and the bias inputs can be obtained as follows:

$$\begin{aligned} W_{p_1, i_1; p_2, i_2} &= \\ \begin{cases} (-2)(c'k + D'_i(p_1, i_1, q) D'_i(p_2, i_2, q)) & \text{if } (p_1, i_1) \neq (p_2, i_2) \\ 0 & \text{if } (p_1, i_1) = (p_2, i_2) \end{cases} \end{aligned} \quad (24)$$

and

$$I_{p_1, i_1} = 2k - D_i'^2(p_1, i_1, q) \quad (25)$$

Hence, we can find the correspondences between frames 1, 2, and 3 using the algorithm in Sec. 3 and then find those between frames 3 and 4, frames 4 and 5, ..., etc., by shifting the execution window of the present algorithm frame by frame. The performance of this algorithm will not depend on the length of the image sequence, and the coherence of the path can be retained. Furthermore, there are only  $n^2$  neurons and  $n^4 - n^2$  interconnections in this algorithm. An illustrative example follows.

**Successive input frames example.** The algorithm was given three frames with four feature points to begin with, and it received seven other frames with the same feature points successively. The resulting correspondences of these ten successive frames are shown in Fig. 5. The total time required to complete the experiment was less than 0.2 s on a DEC 3100 workstation.

## 5 Occlusion Problem

Occlusion refers to objects disappearing partially or totally during an image sequence. Occlusion, which may be caused by an object rotating out of view or being covered by other objects, is an inherent problem in the analysis of moving objects. Chow and Aggarwal<sup>13</sup> presented a method for coping with occlusion by matching predictions of the movement of objects between consecutive frames. Their method assumes that the speed and shape of each object are constant during the time the object is occluded. This requirement, however, makes the proposed algorithm difficult to apply in a real world. Yachida, Asada, and Tsuji<sup>14</sup> used temporal information from the previous frame to produce an algorithm that did not require these assumptions, but their method made it necessary to backtrack objects that were marked as misinterpreted. The hypothesize-and-test algorithm,<sup>1</sup> which predicts the occluded points using established correspondences between all frames and excludes occluded frames, requires a great amount of time to estimate and verify.

In this section, we describe an algorithm that directly establishes the correspondence without backtracking or verifications when occlusion occurs. Since our goal here is to

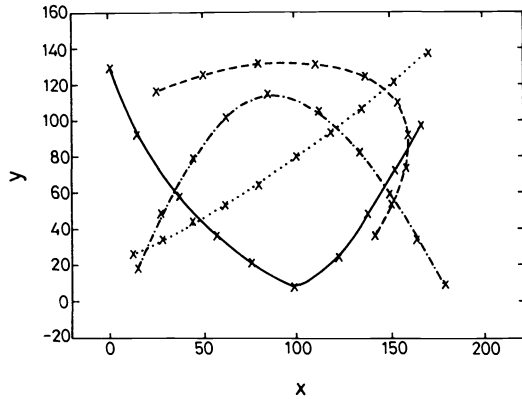


Fig. 5 The completed correspondence of successive frames.

find the overall trajectories of moving objects rather than to find the positions of missing feature points in an image sequence, we do not have to estimate the coordinates of missing points. Thus, our strategy will find the correspondence between the feature points in the current frame and the feature points of the established trajectories in the last two frames, with some feature points being occluded. Thus, we only evaluate the correspondence for these points and ignore the occluded points. The details of our method are described below.

Assume an occluded frame  $q$  with  $n_c (< n)$  feature points being received, in which there should be  $n$  feature points. We ignore the missing points and apply the algorithm presented in the previous section with a new binary state set of neurons, defined as  $V = \{v_{i_1, i_2} | 1 \leq i_1 \leq n, 1 \leq i_2 \leq n_c\}$ , to determine the most probable correspondence for the  $n_c$  visible points as usual. When processing the relationship between the found trajectories and the next frame,  $(q + 1)$ , we take the feature points of the constructed trajectories for the last two frames and the points of the newly acquired normal frame as a set of feature points to be processed. For example, we have constructed the correspondence of an image sequence, up to the  $(q - 1)$ 'th frame. The newly acquired frame  $q$ , however, has an occluded point, as shown in Fig. 6(a). The circle in Fig. 6(a) indicates the occluded feature point. First, we establish a correspondence between the trajectories up to the  $(q - 1)$ 'th frame and the three available points of the  $q$ 'th frame. The result is shown in Fig. 6(b). When the  $(q + 1)$ 'th frame is received, we take the points marked with a circled cross in Fig. 6(c) as a set of feature points to be processed.

Note that the speed continuity of some of the evaluated trajectories will be distorted greatly when we use the next normal frame to determine the trajectory in a previous frame with occluded feature points, because in ignoring the missing points in the occluded frame, the displacement vector used to calculate the speed coherence has been almost doubled. Thus, it follows that the weighting coefficient  $w_2$  of Eq. (3) should be relatively reduced.

*Image correspondence example with occlusion.* We shall now present two examples to demonstrate the use of the proposed algorithm. First, Fig. 7 indicates the feature points of an image sequence, in which two points are occluded and represented by circled crosses. The completed correspondence is shown in Fig. 7. As for Fig. 8, two consecutive feature points are occluded, represented likewise by circled

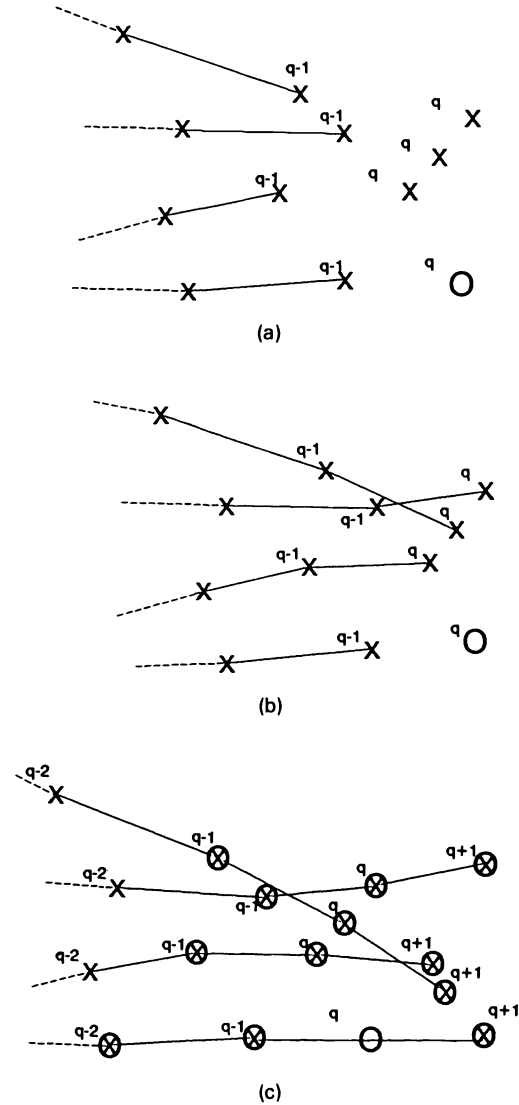


Fig. 6 (a) A frame with an occluded point is acquired—the circle indicates the occluded point in the new frame, (b) the correspondence ignoring the occluded point is evaluated first, and (c) the circled crosses indicate the point set being processed to overcome the occlusion.

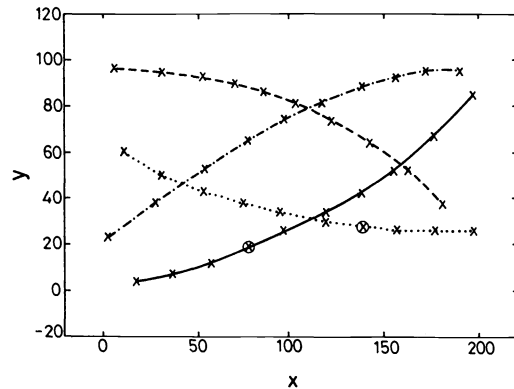
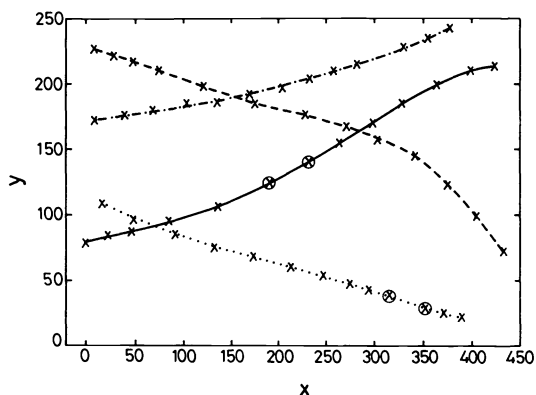


Fig. 7 The image sequence correspondence with single occlusion in two trajectories.



**Fig. 8** The image sequence correspondence with two successive occlusions in two trajectories.

crosses, in each trajectory of two moving objects. Trajectories must be found of these feature points. With  $\lambda = 1$ ,  $w_1 = 60$ ,  $w_2 = 40$ , and  $k = 30$  being assigned, the four paths are found and shown in Fig. 8. The time required for this experiment is about 0.3 s on a DEC 3100 workstation. It can be seen that our algorithm establishes successfully the correspondence of feature points with occlusion.

## 6 Conclusion

A neural network approach to finding trajectories of feature points in a monocular image sequence is presented in this paper. Based on the Hopfield neural model and the assumption of the smoothness of motion, a neural network energy function for a finite image frame correspondence is first introduced. The neural network structure is then derived to deal with correspondence in a sequence of successive images. Moreover, image sequence correspondence with occlusion is also considered. With the first few frame correspondence results in hand, the proposed algorithm only needs to associate the image feature points with the trajectories found. Thus, the complexity and, hence, the hardware cost and computational effort of the algorithm is greatly reduced. In particular, the complexity is reduced to  $O(n^3)$ , where  $n$  is the object number, which does not depend on the length of the image sequence. In the examples presented we show that, due to the massive parallel processing power of neural networks, an optimal correspondence solution can be obtained in real time.

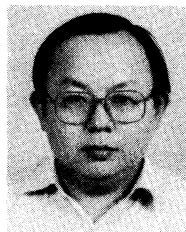
## Acknowledgment

This work was supported in part by the National Science Council of the Republic of China under grant CS79-0210-D009-23.

## References

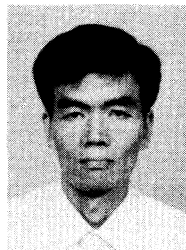
1. I. K. Sethi and R. Jain, "Finding trajectories of feature points in a monocular image sequence," *IEEE Trans. Patt. Anal. Mach. Intell.* **9**, 56–73 (1987).
2. S. T. Barnard and W. B. Thompson, "Disparity analysis of images," *IEEE Trans. Patt. Anal. Mach. Intell.* **2**, 333–340 (1980).

3. J. M. Prager and M. A. Arbib, "Computing the optic flow: The MATCH algorithm and prediction," *Comput. Vis. Graph. Image Process.* **24**, 271–304 (1983).
4. S. M. Haynes and R. Jain, "Event detection and correspondence," *Opt. Eng.* **25**, 387–393 (1986).
5. J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Natl. Acad. Sci. U.S.A.* **79**, 2554–2558 (1982).
6. J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Natl. Acad. Sci. U.S.A.* **81**, 3088–3092 (1984).
7. J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biol. Cybern.* **52**, 141–152 (1985).
8. Y. T. Zhou and R. Chellappa, "Neural network algorithms for motion stereo," in *Proc. IEEE Int. Joint Conf. Neural Networks*, pp. 251–258 (1989).
9. J. J. Hopfield and D. W. Tank, "Simple neural optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit," *IEEE Trans. Circ. Syst.* **33**, 533–541 (1986).
10. Y. T. Zhou, R. Chellappa, A. Vaid, and B. K. Jenkins, "Image restoration using a neural network," *IEEE Trans. Acoust. Speech Sig. Process.* **36**, 1141–1151 (1988).
11. N. M. Nasrabadi and W. Li, "Object recognition by a Hopfield Neural Network," *IEEE Trans. Syst. Man Cyber.* **21**, 1523–1535 (1991).
12. N. M. Nasrabadi and C. Y. Choo, "Hopfield network for stereo vision correspondence," *IEEE Trans. Neural Net.* **3**, 5–12 (1992).
13. W. K. Chow and J. K. Aggarwal, "Computer analysis of planar curvilinear moving images," *IEEE Trans. Comput.* **26**, 179–185 (1977).
14. M. Yachida, M. Asada, and S. Tsuji, "Automatic analysis of moving images," *IEEE Trans. Patt. Anal. Mach. Intell.* **3**, 12–20 (1981).



**Jyh-Yeong Chang** received his BS degree in control engineering in 1976 and MS degree in electronic engineering in 1980, both from National Chiao Tung University, Taiwan. From 1976 to 1978 and 1980 to 1982, he was a research fellow at the Chung Shan Institute of Science and Technology, Taiwan. He received a PhD in electrical engineering from North Carolina State University in 1987. Since 1987 he has been an associate professor of the

Department of Control Engineering at National Chiao Tung University. His research interests include multidimensional signal processing, pattern recognition, robot control, and neural network application.



**Shin-Wen Lee** received the BS and MS degrees in electrical engineering from National Cheng Kung University, Taiwan, in 1973 and 1975, respectively. He received a PhD in electrical engineering from the National Tsing Hua University, Taiwan, in 1992. In 1979 he joined the Industrial Technology Research Institute (ITRI) in Chutung, Taiwan. Currently, he is a researcher at the Advanced Technology Center at ITRI. His research interests include image signal processing, computer vision, computer architecture, and parallel algorithms.

Department of Control Engineering at National Chiao Tung University. His research interests include multidimensional signal processing, pattern recognition, robot control, and neural network application.



**Mong-Fong Horng** received his BS degree in 1989 and MS degree in 1991 in control engineering from National Chiao Tung University, Taiwan. He is now an officer serving in the army. His current research interests include image processing, neural networks, and control engineering.