Optical implementation of the Hopfield model

Nabil H. Farhat, Demetri Psaltis, Aluizio Prata, and Eung Paek

Optical implementation of content addressable associative memory based on the Hopfield model for neural networks and on the addition of nonlinear iterative feedback to a vector–matrix multiplier is described. Numerical and experimental results presented show that the approach is capable of introducing accuracy and robustness to optical processing while maintaining the traditional advantages of optics, namely, parallelism and massive interconnection capability. Moreover a potentially useful link between neural processing and optics that can be of interest in pattern recognition and machine vision is established.

I. Introduction

It is well known that neural networks in the eye–brain system process information in parallel with the aid of large numbers of simple interconnected processing elements, the neurons. It is also known that the system is very adept at recognition and recall from partial information and has remarkable error correction capabilities.

Recently Hopfield described a simple model[1] for the operation of neural networks. The action of individual neurons is modeled as a thresholding operation and information is stored in the interconnections among the neurons. Computation is performed by setting the state (on or off) of some of the neurons according to an external stimulus and, with the interconnections set according to the recipe that Hopfield prescribed, the state of all neurons that are interconnected to those that are externally stimulated spontaneously converges to the stored pattern that is most similar to the external input. The basic operation performed is a nearest-neighbor search, a fundamental operation for pattern recognition, associative memory, and error correction. A remarkable property of the model is that powerful global computation is performed with very simple, identical logic elements (the neurons). The interconnections provide the computation power to these simple logic elements and also enhance dramatically the stor-age capacity; approximately $N/4 \ln N$ bits/neuron can be stored in a network in which each neuron is connected to N others.[2] Another important feature is that synchronization among the parallel computing elements is not required, making concurrent, distributed processing feasible in a massively parallel structure. Finally, the model is insensitive to local imperfections such as variations in the threshold level of individual neurons or the weights of the interconnections.

Given these characteristics we were motivated to investigate the feasibility of implementing optical information processing and storage systems that are based on this and other similar models of associative memory.[3,4] Optical techniques offer an effective means for the implementation of programmable global interconnections of very large numbers of identical parallel logic elements. In addition, emerging optical technologies such as 2-D spatial light modulators, optical bistability, and thin-film optical amplifiers appear to be very well suited for performing the thresholding operation that is necessary for the implementation of the model.

The principle of the Hopfield model and its implications in optical information processing have been discussed earlier.[5,6] Here we review briefly the main features of the model, give as an example the results of a numerical simulation, describe schemes for its optical implementation, then present experimental results obtained with one of the schemes and discuss their implications as a content addressable associative memory (CAM).

II. Hopfield Model

Given a set of **M** bipolar, binary $(1,-1)$ vectors $\mathbf{v}_i^{(m)}$, $i = 1,2,3 \ldots N$, $m = 1,2,3 \ldots \mathbf{M}$, these are stored in a synaptic matrix in accordance with the recipe

$$T_{ij} = \sum_{m=1}^{M} \mathbf{v}_i^{(m)} \mathbf{v}_j^{(m)}, \qquad i,j = 1,2,3 \ldots N, \qquad T_{ii} = 0, \qquad (1)$$

$\mathbf{v}_i^{(m)}$ are referred to as the nominal state vectors of the

Nabil Farhat is with University of Pennsylvania, Moore School of Electrical Engineering, Philadelphia, Pennsylvania 19104; the other authors are with California Institute of Technology, Electrical Engineering Department, Pasadena, California 91125.

© 1985 Optical Society of America.

memory. If the memory is addressed by multiplying the matrix T_{ij} with one of the state vectors, say $v_i^{(mo)}$, it yields the estimate

$$\hat{v}_i^{(mo)} = \sum_j^N T_{ij} v_j^{(mo)} \qquad (2)$$

$$= \sum_{j \neq i}^N \sum_m^M v_i^{(m)} v_j^{(m)} v_j^{(mo)}$$

$$= (N-1)v_i^{(mo)} + \sum_{m \neq m_0} \alpha_{m,mo} v_i^{(m)}, \qquad (3)$$

where

$$\alpha_{m,mo} = \sum_j^N v_j^{(mo)} v_j^{(m)}.$$

$\hat{v}_i^{(mo)}$ consists of the sum of two terms: the first is the input vector amplified by $(N-1)$; the second is a linear combination of the remaining stored vectors and it represents an unwanted cross-talk term. The value of the coefficients $\alpha_{m,mo}$ is equal to $\sqrt{N-1}$ on the average (the standard deviation of the sum of $N-1$ random bits), and since $(M-1)$ such coefficients are randomly added, the value of the second term will on the average be equal to $\sqrt{(M-1)(N-1)}$. If N is sufficiently larger than M, with high probability the elements of the vector $\hat{v}_i^{(mo)}$ will be positive if the corresponding elements of $v_i^{(mo)}$ are equal to $+1$ and negative otherwise. Thresholding of $\hat{v}_i^{(mo)}$ will therefore yield $v_i^{(mo)}$:

$$v_i^{(mo)} = \text{sgn}[\hat{v}_i^{(mo)}] = \begin{cases} +1 \text{ if } \hat{v}_i^{(mo)} > 0 \\ -1 \text{ otherwise.} \end{cases} \qquad (4)$$

When the memory is addressed with a binary valued vector that is not one of the stored words, the vector–matrix multiplication and thresholding operation yield an output binary valued vector which, in general, is an approximation of the stored word that is at the shortest Hamming distance from the input vector. If this output vector is fed back and used as the input to the memory, the new output is generally a more accurate version of the stored word and continued iteration converges to the correct vector.

The insertion and readout of memories described above are depicted schematically in Fig. 1. Note that in Fig. 1(b) the estimate $\hat{v}_i^{(mo)}$ can be viewed as the weighted projection of T_{ij}. Recognition of an input vector that corresponds to one of the state vectors of the memory or is close to it (in the Hamming sense) is manifested by a stable state of the system. In practice unipolar binary (0,1) vectors or words $b_i^{(m)}$ of bit length N may be of interest. The above equations are then applicable with $[2b_i^{(m)} - 1]$ replacing $v_i^{(m)}$ in Eq. (1) and $b_i^{(mo)}$ replacing $v_i^{(mo)}$ in Eq. (2). For such vectors the SNR of the estimate $\hat{v}_i^{(mo)}$ can be shown to be lower by a factor of $\sqrt{2}$.[1]

An example of the T_{ij} matrix formed from four binary unipolar vectors, each being $N = 20$ bits long, is given in Fig. 2 along with the result of a numerical simulation of the process of initializing the memory matrix with a partial version of $b_i^{(4)}$ in which the first eight digits of $b_i^{(4)}$ are retained and the remainder set to zero. The Hamming distance between the initializing vector and $b_i^{(4)}$ is 6 bits and it is 9 or more bits for the other three
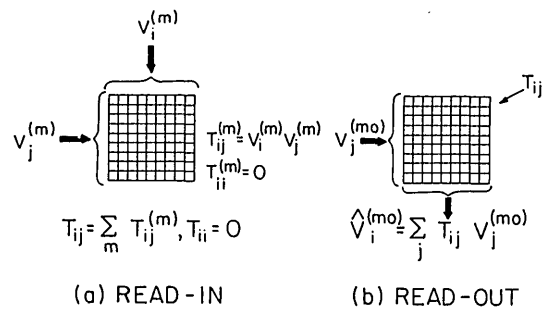

Fig. 1. (a) Insertion and (b) readout of memories.

stored vectors. It is seen that the partial input is recognized as $b_i^{(4)}$ in the third iteration and the output remains stable as $b_i^{(4)}$ thereafter. This convergence to a stable state generally persists even when the T_{ij} matrix is binarized or clipped by replacing negative elements by minus ones and positive elements by plus ones evidencing the robustness of the CAM. A binary synaptic matrix has the practical advantage of being more readily implementable with fast programmable spatial light modulators (SLM) with storage capability such as the Litton Lightmod.[7] Such a binary matrix, implemented photographically, is utilized in the optical implementation described in Sec. III and evaluated in Sec. IV of this paper.

Several schemes for optical implementation of a CAM based on the Hopfield model have been described earlier.[5] In one of the implementations an array of light emitting diodes (LEDs) is used to represent the logic elements or neurons of the network. Their state (on or off) can represent unipolar binary vectors such as the state vectors $b_i^{(m)}$ that are stored in the memory matrix T_{ij}. Global interconnection of the elements is realized as shown in Fig. 3(a) through the addition of nonlinear feedback (thresholding, gain, and feedback) to a conventional optical vector–matrix multiplier[8] in which the array of LEDs represents the input vector and an array of photodiodes (PDs) is used to detect the output vector. The output is thresholded and fed back in parallel to drive the corresponding elements of the LED array. Multiplication of the input vector by the T_{ij} matrix is achieved by horizontal imaging and vertical smearing of the input vector that is displayed by the LEDs on the plane of the T_{ij} mask [by means of an anamorphic lens system omitted from Fig. 3(a) for simplicity]. A second anamorphic lens system (also not shown) is used to collect the light emerging from each row of the T_{ij} mask on individual photosites of the PD array. A bipolar T_{ij} matrix is realized in incoherent light by dividing each row of the T_{ij} matrix into two subrows, one for positive and one for negative values and bringing the light emerging from each subrow to focus on two adjacent photosites of the PD array that are electrically connected in opposition as depicted in Fig. 3(b). In the system shown in Fig. 3(a), feedback is achieved by electronic wiring. It is possible and preferable to dispose of electronic wiring altogether and replace it by optical feedback. This can be achieved by combining the PD and LED arrays in a single compact hybrid or

(1) (2) (3) (4)
b_i b_i b_i b_i

T_{ij}

(a)   (b)

01110111000000000000 PARTIAL INPUT ($b_i^{(4)}$)

-2 -4 6 4 -8 4 6 -4 2 -2 10 -2 -10 -2 0 -8 8 2 2 -2 1st ESTIMATE   } 1st
                                                                 } ITERATION
00110110101010000001110 1st THRESHOLDING

00110110101010000101100 2nd ITERATION

01110110101010001011110 3rd ITERATION   STABLE

01110110101010001011110 4th ITERATION

(c)

Fig. 2. Numerical example of recovery from partial input; $N = 20$, $M = 4$. (a) Stored vectors, (b) memory or (synaptic) matrix, (c) results of initializing with a partial version of $b_i^{(4)}$.

$V_i = \sum_i T_{ij} V_j$, $T_{ii} = 0$

THRESHOLDING, FEEDBACK, AND GAIN.

PD ARRAY

LED ARRAY (LD, LED) T_{ij} MEMORY MASK

LED   MASK   PD

FEEDBACK, GAIN, AND THRESHOLDING

(a)

ONE ROW OF T_{ij} MASK $\{^{+1}_{-1}$

T_{ij} MASK

$V_0 = R(I_2 - I_1)$

(b)

Fig. 3. Concept for optical implementation of a content addressable memory based on the Hopfield model. (a) Matrix–vector multiplier incorporating nonlinear electronic feedback. (b) Scheme for realizing a binary bipolar memory mask transmittance in incoherent light.

monolithic structure that can also be made to contain all ICs for thresholding, amplification, and driving of LEDs. Optical feedback becomes even more attractive when we consider that arrays of nonlinear optical light amplifiers with internal feedback[9] or optical bistability

devices (OBDs)[10] can be used to replace the PD/LED arrays. This can lead to simple compact CAM structures that may be interconnected to perform higher-order computations than the nearest-neighbor search performed by a single CAM.

We have assembled a simple optical system that is a variation of the scheme presented in Fig. 3(a) to simulate a network of $N = 32$ neurons. The system, details of which are given in Figs. 5–8, was constructed with an array of thirty-two LEDs and two multichannel silicon PD arrays, each consisting of thirty-two elements. Twice as many PD elements as LEDs are needed in order to implement a bipolar memory mask transmittance in incoherent light in accordance with the scheme of Fig. 3(b). A bipolar binary T_{ij} mask was prepared for $M = 3$ binary state vectors. The three vectors or words chosen, their Hamming distances from each other, and the resulting T_{ij} memory matrix are shown in Fig. 4. The mean Hamming distance between the three vectors is 16. A binary photographic transparency of 32×64 square pixels was computer generated from the T_{ij} matrix by assigning the positive values in any given row of T_{ij} to transparent pixels in one subrow of the mask and the negative values to transparent pixels in the adjacent subrow. To insure that the image of the input LED array is uniformly smeared over the memory mask it was found convenient to split the mask in two halves, as shown in Fig. 5, and to use the resulting submasks in two identical optical arms as shown in Fig. 6. The size of the subrows of the memory submasks was made exactly equal to the element size of the PD arrays in the vertical direction which were placed in register

Stored words:

```
Word 1 : 1 1 1 0 0 0 0 1 0 1 0 1 1 1 0 1 1 0 1 1 1 1 0 1 1 0 0 0 0 0 1 0
Word 2 : 0 1 1 0 0 0 0 0 0 1 0 0 1 0 1 0 1 0 0 1 1 1 1 0 1 0 1 1 0 1 0
Word 3 : 1 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 1 0 1 1 0 0 0 0 1 1 0 0 0 0
```

Hamming distance from word to word:

WORD	1	2	3
1	0	15	14
2	15	0	19
3	14	19	0

Clipped memory matrix:

```
 0 -1  1  1 -1 -1  1  1  1  1 -1  1  1  1  1 -1  1 -1  1  1  1  1 -1 -1  1 -1  1 -1 -1 -1 -1 -1
-1  0  1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1 -1  1  1  1  1 -1  1  1  1  1  1  1  1 -1 -1  1 -1  1 -1
 1  1  0 -1 -1 -1 -1  1 -1  1  1  1  1  1 -1  1 -1 -1  1 -1  1  1 -1  1 -1 -1 -1  1 -1 -1  1 -1
 1 -1 -1  0  1  1  1  1  1  1  1  1  1 -1  1 -1 -1 -1  1 -1 -1 -1 -1 -1 -1 -1  1  1 -1  1 -1  1
-1 -1 -1  1  0  1  1 -1  1 -1 -1 -1 -1 -1  1 -1  1  1 -1  1 -1 -1  1  1  1 -1  1  1 -1  1
-1 -1 -1  1  1  0  1 -1  1 -1 -1 -1 -1 -1  1 -1  1  1 -1  1 -1 -1  1  1  1  1 -1  1  1 -1  1
 1 -1 -1  1  1  1  0  1  1  1  1  1  1 -1  1 -1 -1 -1  1 -1 -1 -1 -1 -1 -1 -1  1  1 -1  1 -1  1
 1 -1  1  1 -1 -1  1  0  1 -1  1  1  1  1  1 -1  1 -1  1  1  1  1 -1 -1  1 -1  1 -1 -1 -1 -1 -1
 1 -1 -1  1  1  1  1  1  0  1  1 -1  1 -1  1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1  1 -1  1 -1  1 -1 -1
 1 -1  1  1 -1 -1  1  1  1  0 -1  1  1  1  1 -1  1 -1  1  1  1  1 -1 -1  1 -1  1 -1 -1 -1 -1 -1
-1 -1  1  1 -1 -1  1 -1  1 -1  0 -1 -1  1  1 -1 -1  1 -1 -1  1  1  1 -1 -1  1  1  1 -1 -1 -1
 1 -1  1  1 -1 -1  1  1  1  1 -1  0  1  1  1 -1  1 -1  1  1  1  1 -1 -1  1 -1  1 -1 -1 -1 -1 -1
 1 -1  1  1 -1 -1  1  1  1  1 -1  1  0  1  1 -1  1 -1  1  1  1  1 -1 -1  1 -1  1 -1 -1 -1 -1 -1
 1  1  1 -1 -1 -1 -1  1 -1  1  1  1  1  0 -1  1 -1 -1  1 -1  1  1 -1  1 -1 -1 -1  1 -1 -1  1 -1
 1 -1 -1  1  1  1  1  1  1  1  1  1  1 -1  0 -1 -1 -1  1 -1 -1 -1 -1 -1 -1 -1  1  1 -1  1 -1  1
-1  1  1  1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1 -1  0  1  1 -1  1  1  1  1  1  1  1  1 -1 -1  1 -1  1 -1
 1  1 -1 -1  1  1 -1  1 -1  1 -1 -1 -1  1 -1  1  0 -1  1  1 -1 -1  1 -1 -1 -1 -1  1 -1 -1  1  1
-1  1 -1 -1  1  1 -1 -1 -1 -1  1 -1 -1 -1 -1  1 -1  0 -1 -1 -1 -1  1  1 -1  1 -1  1  1  1  1  1
 1 -1  1  1 -1 -1  1  1  1  1 -1  1  1  1  1 -1  1 -1  0  1  1  1 -1 -1  1 -1  1 -1 -1 -1 -1 -1
 1  1 -1 -1  1  1 -1  1 -1  1 -1  1 -1  1 -1  1  1 -1  1  0 -1 -1  1  1 -1 -1 -1 -1  1  1  1
 1  1  1 -1 -1 -1 -1  1 -1  1  1  1  1  1 -1  1 -1 -1  1 -1  0  1 -1  1 -1 -1 -1  1 -1 -1  1 -1
 1  1  1 -1 -1 -1 -1  1 -1  1  1  1  1  1 -1  1 -1 -1  1 -1  1  0 -1  1 -1 -1 -1  1 -1 -1  1 -1
-1  1 -1 -1  1  1 -1 -1 -1 -1  1 -1 -1 -1 -1  1  1 -1 -1  1  0  1 -1  1  1 -1  1  1  1  1  1
-1  1  1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1 -1  1  1  1  1 -1  1  1  1  1  0  1  1 -1  1  1  1 -1
 1  1 -1 -1  1  1 -1  1 -1  1 -1  1 -1  1 -1  1 -1 -1  1 -1  1  1 -1 -1  1  0 -1 -1  1  1  1  1
-1  1 -1 -1  1  1 -1 -1 -1 -1  1 -1 -1 -1 -1  1  1 -1 -1  1  1 -1  0 -1  1  1  1  1
 1 -1 -1  1  1  1  1  1  1  1  1  1  1 -1  1 -1 -1 -1  1 -1 -1 -1 -1 -1  1 -1  0  1  1 -1  1  1
-1 -1  1  1 -1 -1  1 -1  1 -1  1 -1 -1  1  1 -1 -1  1 -1 -1  1  1  1 -1 -1  1  1  0 -1 -1 -1 -1
-1  1  1  1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1 -1  1  1  1  1 -1  1  1  1  1  1  1  1 -1  0  1 -1 -1
-1  1 -1 -1  1  1 -1  1 -1  1 -1  1 -1  1 -1  1  1 -1 -1  1  1 -1  1 -1  1  1  1  1  1  0  1  1
-1 -1 -1  1  1  1  1 -1  1 -1 -1 -1 -1 -1  1 -1  1  1 -1  1 -1 -1  1  1  1 -1  1  1 -1  1  0 -1
-1 -1 -1  1  1  1  1 -1  1 -1 -1 -1 -1 -1  1 -1  1  1 -1  1 -1 -1  1  1  1 -1  1 -1 -1  1  1 -1  0
```

Fig. 4. Stored words, their Hamming distances, and their clipped T_{ij} memory matrix.
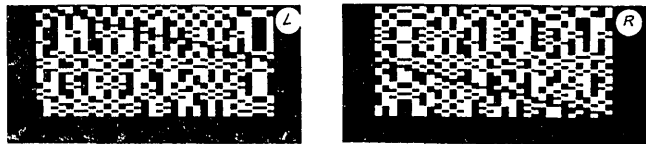

Fig. 5. Two halves of T_{ij} memory mask.

against the masks. Light emerging from each subrow of a memory submask was collected (spatially integrated) by one of the vertically oriented elements of the multichannel PD array. In this fashion the anamorphic optics required in the output part of Fig. 3(a) are disposed of, resulting in a more simple and compact system. Pictorial views of the input LED array and the two submask/PD array assemblies are shown in Figs. 7(a) and (b), respectively. In Fig. 7(b) the left memory submask/PD array assembly is shown with the submask removed to reveal the silicon PD array situated behind it. All electronic circuits (amplifiers, thresholding comparators, LED drivers, etc.) in the thirty-two parallel feedback channels are contained in the electronic amplification and thresholding box shown in Fig. 6(a) and in the boxes on which the LED array and the two submask/PD array assemblies are mounted (see Fig. 7). A pictorial view of a composing and display box is shown in Fig. 8. This contains an arrangement of thirty-two switches and a thirty-two element LED display panel whose elements are connected in parallel to the input LED array. The function of this box is to compose and
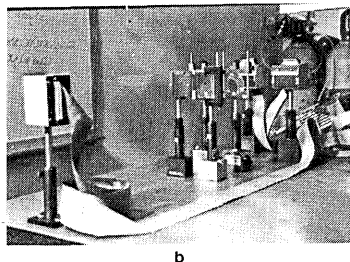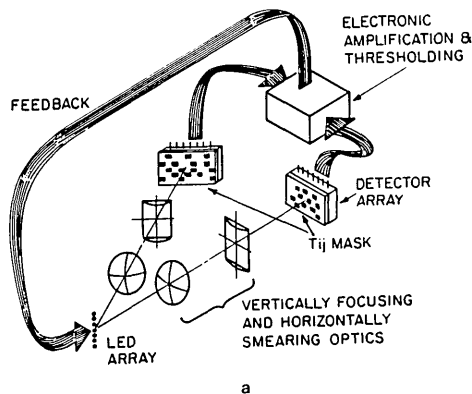
Fig. 6. Arrangement for optical implementation of the Hopfield model: (a) optoelectronic circuit diagram, (b) pictorial view.
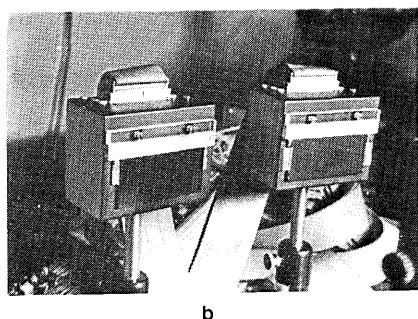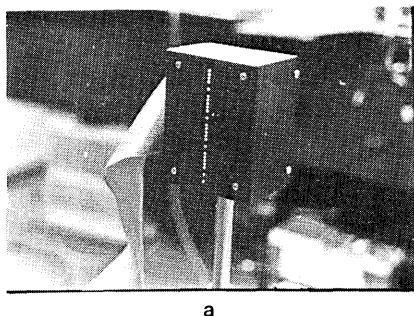

Fig. 7. Views of (a) input LED array and (b) memory submask/PD array assemblies.

display the binary input word or vector that appears on the input LED array of the system shown in Fig. 7(a). Once an input vector is selected it appears displayed on the composing box and on the input LED box simultaneously. A single switch is then thrown to release the system into operation with the composed vector as the
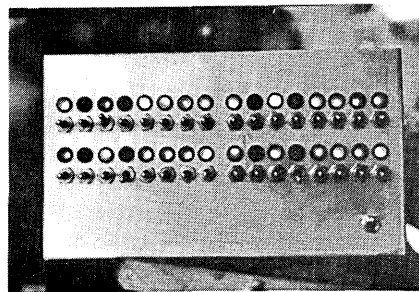
Fig. 8. Word composer and display box.

initializing vector. The final state of the system, the output, appears after a few iterations displayed on the input LED array and the display box simultaneously. The above procedure provides for convenient exercising of the system in order to study its response vs stimulus behavior. An input vector is composed and its Hamming distance from each of the nominal state vectors stored in the memory is noted. The vector is then used to initialize the CAM as described above and the output vector representing the final state of the CAM appearing, almost immediately, on the display box is noted. The response time of the electronic feedback channels as determined by the 3-dB roll-off of the amplifiers was ~60 msec. Speed of operation was not an issue in this study, and thus low response time was chosen to facilitate the experiment.

IV. Results

The results of exercising and evaluating the performance of the system we described in the preceding section are tabulated in Table I. The first run of initializing vectors used in exercising the system were error laden versions of the first word $b_i^{(1)}$. These were obtained from $b_i^{(1)}$ by successively altering (switching) the states of 1,2,3 . . . up to N of its digits starting from the Nth digit. In doing so the Hamming distance between the initializing vector and $b_i^{(1)}$ is increased linearly in unit steps as shown in the first column of Table I whereas, on the average, the Hamming distance between all these initializing vectors and the other two state vectors remained approximately the same, about $N/2 = 16$. The final states of the memory, i.e., the steady-state vectors displayed at the output of the system (the composing and display box) when the memory is prompted by the initializing vectors, are listed in column 2 of Table I. When the Hamming distance of the initializing vector from $b_i^{(1)}$ is <11, the input is always recognized correctly as $b_i^{(1)}$. The CAM is able therefore to recognize the input vector as $b_i^{(1)}$ even when up to 11 of its digits (37.5%) are wrong. This performance is identical to the results obtained with a digital simulation shown in parenthesis in column 2 for comparison. When the Hamming distance is increased further to values lying between 12 and 22, the CAM is confused and identifies erroneously other state vectors, mostly $b_i^{(3)}$, as the input. In this range, the Hamming distance of the initializing vectors from any of the stored vectors is approximately equal making it more difficult for the CAM to decide. Note that the performance of

Table I. Optical CAM Performance

Hamming distance of initializing vector from $b_i^{(m)}$	Recognized vector ($m = 1$)	Recognized vector ($m = 2$)	Recognized vector ($m = 3$)
0	1 (1)	2 (2)	3 (3)
1	1 (1)	2 (2)	3 (3)
2	1 (1)	2 (2)	3 (3)
3	1 (1)	2 (2)	3 (3)
4	1 (1)	2 (2)	3 (3)
5	1 (1)	2 (2)	3 (3)
6	1 (1)	2 (2)	3 (3)
7	1 (1)	2 (2)	3 (3)
8	1 (1)	2 (2)	3 (3)
9	1 (1)	2 (2)	3 (3)
10	1 (1)	1 (1)	3 (3)
11	1 (1)	2 (2)	3 (3)
12	3 (3)	$\bar{3}$,2 ($\bar{3}$)	3 (3)
13	3 (3)	$\bar{3}$ ($\bar{3}$)	3 ($\bar{2}$)
14	3 (3)	1,$\bar{3}$ (1)	3 ($\bar{2}$)
15	1 (OSC)	1 (1)	2,3 ($\bar{2}$)
16	3 (OSC)	1 (1)	$\bar{2}$ ($\bar{2}$)
17	3 (OSC)	1 (OSC)	$\bar{2}$ ($\bar{2}$)
18	3 (3)	1 ($\bar{2}$)	3 (OSC)
19	3 ($\bar{2}$)	$\bar{2}$ ($\bar{2}$)	$\bar{2}$ ($\bar{2}$)
20	3 ($\bar{1}$)	$\bar{2}$ ($\bar{2}$)	$\bar{2}$ (OSC)
21	1,2 ($\bar{1}$)	$\bar{2}$ ($\bar{2}$)	$\bar{3}$ (OSC)
22	3 ($\bar{1}$)	$\bar{2}$ ($\bar{2}$)	$\bar{3}$ (OSC)
23	$\bar{1}$ ($\bar{1}$)	$\bar{2}$ ($\bar{2}$)	$\bar{3}$ (OSC)
24	$\bar{1}$ ($\bar{1}$)	$\bar{2}$ ($\bar{2}$)	$\bar{3}$ ($\bar{3}$)
25	$\bar{1}$ ($\bar{1}$)	$\bar{2}$ ($\bar{2}$)	$\bar{3}$ ($\bar{3}$)
26	$\bar{1}$ ($\bar{1}$)	$\bar{2}$ ($\bar{2}$)	$\bar{3}$ ($\bar{3}$)
27	$\bar{1}$ ($\bar{1}$)	$\bar{2}$ ($\bar{2}$)	$\bar{3}$ ($\bar{3}$)
28	$\bar{1}$ ($\bar{1}$)	$\bar{2}$ ($\bar{2}$)	$\bar{3}$ ($\bar{3}$)
29	$\bar{1}$ ($\bar{1}$)	$\bar{2}$ ($\bar{2}$)	$\bar{3}$ ($\bar{3}$)
30	$\bar{1}$ ($\bar{1}$)	$\bar{2}$ ($\bar{2}$)	$\bar{3}$ ($\bar{3}$)
31	$\bar{1}$ ($\bar{1}$)	$\bar{2}$ ($\bar{2}$)	$\bar{3}$ ($\bar{3}$)
32	$\bar{1}$ ($\bar{1}$)	$\bar{2}$ ($\bar{2}$)	$\bar{3}$ ($\bar{3}$)

the CAM and results of digital simulation in this range of Hamming distance are comparable except for the appearance of oscillations (designated by OSC) in the digital simulation when the outcome oscillated between several vectors that were not the nominal state vectors of the CAM. Beyond a Hamming distance of 22 both the optical system and the digital simulation identified the initializing vectors as the complement $\overline{b}_i^{(1)}$ of $b_i^{(1)}$. This is expected because it can be shown using Eq. (1) that the T_{ij} matrix formed from a set of vectors $b_i^{(m)}$ is identical to that formed by the complementary set $\overline{b}_i^{(m)}$. The complementary vector can be viewed as a contrast reversed version of the original vector in which zeros and ones are interchanged. Recognition of a complementary state vector by the CAM is analogous to our recognizing a photographic image from the negative.

Similar results of initializing the CAM with error laden versions of $b_i^{(2)}$ and $b_i^{(3)}$ were also obtained. These are presented in columns 2 and 3 of Table I. Here again we see when the Hamming distance of the initializing vector from $b_i^{(3)}$, for example, ranged between 1 and 14, the CAM recognized the input correctly as $b_i^{(3)}$ as shown in column 3 of the table and as such it did slightly better than the results of digital simulation. Oscillatory behavior is also observed here in the digital simulation when the range of Hamming distance between the ini-

tializing vector from all stored vectors approached the mean Hamming distance between the stored vectors. Beyond this range the memory recognizes the input as the complementary of $b_i^{(3)}$.

In studying the results presented in Table I several observations can be made: The optically implemented CAM is working as accurately as the digital simulations and perhaps better if we consider the absence of oscillations. These are believed to be suppressed in the system because of the nonsharp thresholding performed by the smoothly varying nonlinear transfer function of electronic circuits compared with the sharp thresholding in digital comptations. The smooth nonlinear transfer function and the finite time constant of the optical system provide a relaxation mechanism that substitutes for the role of asynchronous switching required by the Hopfield model. Generally the system was able to conduct successful nearest-neighbor search when the inputs to the system are versions of the nominal state vectors containing up to ~30% error in their digits. It is worth noting that this performance is achieved in a system built from off-the-shelf electronic and optical components and with relatively little effort in optimizing and fine tuning the system for improved accuracy, thereby confirming the fact that accurate global computation can be performed with relatively inaccurate individual components.

V. Discussion

The number M of state vectors of length N that can be stored at any time in the interconnection matrix T_{ij} is limited to a fraction of N. An estimate of $M \simeq 0.1N$ is indicated in simulations involving a hundred neurons or less[1] and a theoretical estimate of $M \simeq N/4 \ln N$ has recently been obtained.[2] It is worthwhile to consider the number of bits that can be stored per interconnection or per neuron. The number of pixels required to form the interconnection matrix is N^2. Since such a T_{ij} memory matrix can store up to $M \simeq N/4 \ln N$ (N-tuples), the number of bits stored is $MN = N^2/4 \ln N$. The number of bits stored per memory matrix element or interconnection is $MN/N^2 = (4 \ln N)^{-1}$, while the number of bits stored per neuron is $MN/N = M$.

The number of stored memories that can be searched for a given initializing input can be increased by using a dynamic memory mask that is rapidly addressed with different T_{ij} matrices each corresponding to different sets of M vectors. The advantage of programmable SLMs for realizing this goal are evident. For example, the Litton Lightmod (magnetooptic light modulator), which has nonvolatile storage capability and can provide high frame rates, could be used. A frame rate of 60 Hz is presently specified for commercially available units of 128 × 128 pixels which are serially addressed.[7] Units with 256 × 256 pixels are also likely to be available in the near future with the same frame rate capability. Assuming a memory mask is realized with a Litton Lightmod of 256 × 256 pixels we have $N = 256$, $M \simeq 0.1N \simeq 26$ and a total of 26 × 60 = 1560 vectors can be searched or compared per second against an initializing input vector. Speeding up the frame rate of the Litton

Lightmod to increase memory throughput beyond the above value by implementing parallel addressing schemes is also possible. Calculations show that the maximum frame rate possible for the device operating in reflection mode with its drive lines heat sunk is 10 kHz.[7] This means the memory throughput estimated above can be increased to search 2.6×10^5 vectors/sec, each being 256 bits long, or a total of 6.7×10^8 bits/sec. This is certainly a respectable figure, specially when we consider the error correcting capability and the associative addressing mode of the Hopfield model; i.e., useful computation is performed in addition to memory addressing.

The findings presented here show that the Hopfield model for neural networks and other similar models for content addressable and associative memory fit well the attributes of optics, namely, parallel processing and massive interconnection capabilities. These capabilities allow optical implementation of large neural networks based on the model. The availability of nonlinear or bistable optical light amplifiers with internal feedback, optical bistability devices, and nonvolatile high speed spatial light modulators could greatly simplify the construction of optical CAMs and result in compact modules that can be readily interconnected to perform more general computation than nearest-neighbor search. Such systems can find use in future generation computers, artificial intelligence, and machine vision.

References

1. J. J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," Proc. Natl. Acad. Sci. USA **79**, 2554 (1982).
2. R. J. McEliece, E. C. Posner, and S. Venkatesh, California Institute of Technology, Electrical Engineering Department; private communication.
3. G. E. Hinton and J. A. Anderson, *Parallel Models of Associative Memory* (LEA Publishers, Hillsdale, N.J., 1981).
4. T. Kohonen, *Content Addressable Memories* (Springer, New York, 1980).
5. D. Psaltis and N. Farhat, "A New Approach to Optical Information Processing Based On the Hopfield Model," in *Technical Digest, ICO-13 Conference, Sapporo* (1984), p. 24.
6. D. Psaltis and N. Farhat, "Optical Information Processing Based on an Associative-Memory Model of Neural Nets with Thresholding and Feedback," Opt. Lett. **10**, 98 (1985).
7. W. Ross, D. Psaltis, and R. Anderson, "Two-Dimensional Magneto-Optic Spatial Light Modulator For Signal Processing," Opt. Eng. **22**, 485 (1983).
8. J. W. Goodman, A. R. Dias, and L. M. Woody, "Fully Parallel, High-Speed Incoherent Optical Method for Performing Discrete Fourier Transforms," Opt. Lett. **2**, 1 (1978).
9. Z. Porada, "Thin Film Light Amplifier with Optical Feedback," Thin Solid Films **109**, 213 (1983).
10. H. M. Gibbs et al., "Optical Bistable Devices: The Basic Components of All-Optical Circuits," Proc. Soc. Photo-Opt. Instrum. Eng. **269**, 75 (1981).