VDO in general. Specifically, the VDO cannot be used to compute $(v_i - v_j)$ when i and j belong to different recurrent subchains. Thus our algorithm in place of the VDO in Howard's algorithm has this advantage. Finally, we note that replacing the VDO requires only a negligible increase in computational cost, since each cycle of the algorithm given by Theorems 1 and 2 when used to compute $\boldsymbol{g}$ and $\boldsymbol{v}$ requires only $2N^2 + 2(N - k)$ storage locations.

IV. Conclusion

We have presented a new algorithm to compute $(\boldsymbol{I} - z\boldsymbol{P})^{-1}$ in a factored form for a finite stochastic matrix $\boldsymbol{P}$. This algorithm involves only the computation of constant matrices and can be easily implemented on a digital computer. An immediate application of this algorithm in Markov decision processes is to replace the value determining operation in Howard's policy iteration algorithm with the advantage of being able to compute the exact asymptotic performance of an optimal stationary policy.

References

[1] L. A. Zadeh and C. A. Desoer, *Linear System Theory*. New York: McGraw-Hill, 1963.
[2] R. A. Howard, *Dynamic Programming and Markov Processes*. New York: Wiley, 1960.
[3] P. G. Hoel, S. C. Port, and C. J. Stone, *Introduction to Stochastic Processes*. Boston: Houghton Mifflin, 1972.
[4] V. I. Romanovsky, *Discrete Markov Chains*. The Netherlands: Wolters-Noordhoff, 1970.
[5] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*. Princeton, N.J.: Van Nostrand, 1960.
[6] R. A. Howard, *Dynamic Probabilistic Systems*, vols. I and II. New York: Wiley, 1971.
[7] C. Derman, *Finite State Markovian Decision Processes*. New York: Academic, 1970.
[8] H. Mine and S. Osaki, *Markovian Decision Processes*. New York: American Elsevier, 1970.
[9] U. U. J. Leverrier, "Sur les variations séculaires des élements elliptiques des sept planètes principales," *J. Math. Pures Appl.*, vol. 5, pp. 220–254, 1840.
[10] A. S. Householder, *The Theory of Matrices in Numerical Analysis*. New York: Blaisdell, 1964.

Distributed Associative Memory for Patterns

YOH-HAN PAO, SENIOR MEMBER, IEEE, AND FRANK L. MERAT

***Abstract*—A technique for implementing distributed associative memories for the storage and recognition of patterns is described. The implementation may either be in terms of a mathematical structure, i.e., software, or in terms of an actual operable device suited to the memorization and recognition of specific types of patterns, i.e., hardware. The patterns are represented by sequences of binary values, and each pattern is associated with a reference, a Walsh function in the present case. The products of patterns and references are summed and stored. Recognition is achieved by regeneration of the reference, albeit in distorted form, and recognition of it by means of a discriminant function. Variations of this scheme can be implemented to suit different circumstances.**

I. Introduction

This paper presents a technique for implementing distributed associative memories. The implementation may either be in terms of a mathematical structure, i.e., software, or in terms of an actual operable device suited to the memorization and recognition of specific types of patterns, i.e., hardware.

Manuscript received November 7, 1974; revised May 12, 1975. This work was initiated during a visit by Y-H. Pao to the Bionics Research Laboratory, School of Artificial Intelligence, University of Edinburgh, Edinburgh, Scotland, on a NATO Senior Science Fellowship.
The authors are with the Department of Engineering, Case Western Reserve University, Cleveland, Ohio 44106.

In the present context, the meaning of pattern recognition is not limited to the act of recognizing actual pictorial patterns. Ensembles of symbols and arrays of parameter values, in fact, all constitute patterns. Such patterns are encountered in medical diagnostics, linguistics, economics, and management, in fact, wherever complex systems need to be characterized. In addition, we note that the power of any information processing system is closely related to its ability to recognize patterns.

Although the history of pattern recognition activities is relatively brief, considerable progress has been attained in a number of types of problems. This is borne out, for example, by the papers and extensive bibliography in the October 1972 Proceedings of the IEEE (Special Issue on Digital Pattern Recognition) and by the companion July 1972 Proceedings of the IEEE (Special Issue on Digital Picture Processing).

The implementation presented in this paper differs from previous work in advocating an extreme version of parallel storage and parallel processing, resulting in much decreased requirements in storage capacity and in processing time. In addition, the discriminant function used to decide which pattern had indeed been presented for recognition may be structured to emphasize contexts and/or features.

A statement of the underlying ideas is contained in Section II. Experimental results for a computer simulation of a software implementation of such a memory, with a capacity of 64 storage sites, are presented in Section III.

II. Theory

Let $\boldsymbol{x} = (x_1,\cdots,x_N)$ and $\boldsymbol{y} = (y_1,\cdots,y_N)$ be two N-component vectors. We now define a vector product of the two vectors $\boldsymbol{z} = \boldsymbol{xy}$ by

$$z = (x_1y_1x_2y_2 \cdots x_iy_i \cdots x_Ny_N). \tag{1}$$

That is, the ith component of the product vector is the product of the ith component of the factors. Note that this is associative, so that $\boldsymbol{u}(\boldsymbol{vw}) = (\boldsymbol{uv})\boldsymbol{w}$.

Also define a scalar product $\langle \boldsymbol{z} \rangle = \langle \boldsymbol{xy} \rangle$ by

$$\langle z \rangle = \langle \boldsymbol{xy} \rangle \sum_{i=1}^{n} x_i y_i = \sum_{i=1}^{N} z_i. \tag{2}$$

In particular, if the elements of $\boldsymbol{x}$ are either $+1$ or -1, it follows that

$$\boldsymbol{xx} = (1111 \cdots 1 \cdots 1) \tag{3}$$

the vector of all ones and

$$\langle \boldsymbol{xx} \rangle = N. \tag{4}$$

Let $\boldsymbol{x}_p$, $p = 1,2,\cdots,p$, represent *patterns* and let $\boldsymbol{y}_n$, $n = 1,2,\cdots,N$ represent a special subgroup of patterns which are called *references* and which have the additional properties that

$$\langle \boldsymbol{y}_n \rangle = \sum_{j=1}^{N} (y_j)_n = 0, \qquad \text{for all } n.$$

$(1/N) \langle \boldsymbol{y}_m \boldsymbol{y}_n \rangle = \delta_{mn}$ and $\boldsymbol{y}_m \boldsymbol{y}_n = \boldsymbol{y}_k$, where $\boldsymbol{y}_k$ is also a member of this same subgroup of patterns.

Let $\boldsymbol{x}$ be a pattern, $\boldsymbol{y}$ a reference, and $\boldsymbol{m} = \boldsymbol{xy}$ the memory. Then, given a pattern $\boldsymbol{x}$ and a memory $\boldsymbol{m}$, one forms

$$\boldsymbol{xm} = \boldsymbol{x}(\boldsymbol{xy}) = (\boldsymbol{xx}) = \boldsymbol{y} \tag{5}$$

to retrieve the reference $\boldsymbol{y}$.

If we can unmistakably (and with little effort) recognize that $\boldsymbol{y}$ has been retrieved, then pattern $\boldsymbol{x}$ has, in fact, been recognized.

One of the underlying ideas of this present technique is to store many such individual memory vectors $\boldsymbol{m}_i$ in the same storage space without regard for the fact that the space had already been used in a certain sense by other m vectors.

That is, when one has P patterns x_p and references y_p, (the x and y vectors are associated in a specific and possibly trainable way—this is a second way in which this memory is an "associative" memory) the individual memories are

$$m_p = x_p y_p$$

and the overall associative memory is

$$m = \sum_{p=1}^{P} x_p y_p. \tag{6}$$

Of course, now m is no longer binary; it is multivalued, and retrieval is no longer so clean. In particular, when we input x_q, we obtain

$$x_q \cdot m = y_q + \sum_{p \neq q} x_q x_p y_p \equiv r_q \tag{7}$$

where r_q is, in general, y_q with additional contributions from other y vectors.

A discriminant vector f is now chosen to emphasize a certain context or certain "features" or provide greatest resolution in certain regions of the N-dimensional pattern space. In general,

$$f = \sum a_k y_k \tag{8}$$

and

$$\langle r_q f \rangle = a_q + \sum_{p \neq q} \sum_{k} a_k \langle x_q x_p y_p y_k \rangle = D_q \tag{9}$$

a real number.

In order to be able to recognize that x_q was presented for recognition, it is only necessary that the association

$$x_q \Leftrightarrow D_q \tag{10}$$

be unique and that the difference between any pair of D_q and $D_{q'}$ be measureable with sufficient resolution.

To appreciate the merits and disadvantages of this type of memory, we compare this technique with that of straightforward template matching.[1] In particular, consider $P = 10^3$ patterns, each with $n = 10^6$ elements. In the template matching technique, $NP = 10^9$ binary value storage sites, and about 10^{10} processing steps are required for determining which pattern had been presented for recognition. These overwhelming requirements in one form or another constitute the very real limitations to the recognition of complex patterns in real time.

In contrast to that, the present technique requires only 10^6 multivalued storage sites and only 2×10^6 processing steps, representing a decrease of two orders of magnitude in storage capacity and about four orders of magnitude in processing time. These savings in storage space and processing time are due to the much less stringent demands of the "recognition" act required of this technique as compared with the detailed comparison required of the template-matching technique.

In practice, the uniqueness of the association (10) need not be determined theoretically *a priori*; it can be determined experimentally as the vector f is chosen. However, in the use of technique as an analytical tool for the study of the time evolution of complex systems (modeled in terms of patterns of values of state variables), accidental degeneracies might be troublesome and need to be investigated further.

In prior work, there are occasional instances of the idea of combining many patterns in a distributed memory. Examples of these are a paper by C. R. Rosen [7] and some papers by D. G. Willis [2], D. J. Willshaw [3], [4], and Longuet-Higgins [5]. Although the Rosen and the Willis papers are preliminary discussions and the Willshaw and Longuet-Higgins ideas tend to emphasize optical holography, they pursue essentially the same goals as those of the present paper.

The characteristics of a 64-component distributed associative memory have been explored experimentally (i.e., by computer simulation), and the results are presented and discussed in the next section. At the present time, memories with significantly larger capacity would have to be implemented in a hardware manner with video discs, laser scanning, and printing for input and output. All other methods would be too slow.

[1] The template-matching technique is cited primarily to present a reference for comparison. Other better techniques do exist for certain specific situations, and a number of these are discussed in a review article by L. D. Harmon [1].

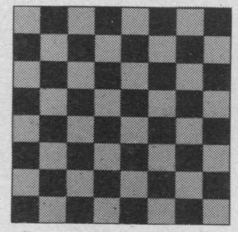

Fig. 1. Number of different patterns which may be displayed using an 8 × 8 array of picture elements (each picture element being either +1 or −1).

III. Characteristics of a 64-Element Memory

The memory selected for detailed study contains 64 storage sites and is suitable for storage of patterns made up of 64 resolution cells.

For pictorial purposes, these 64 resolution cells are arranged in the form of an eight by eight array of squares. For the purposes of this particular study, the different patterns are formed by assigning either a value of +1 or −1 to each and every square. This technique itself is not limited to this particular class of patterns, and even for 64 component patterns, each of the elements may be a real number rather than +1 or −1.

Even in this binary case, the pattern space is not trivial nor of limited interest. An eight by eight display such as that shown in Fig. 1 can, in fact, be used to display any one of 2^{64} different patterns, that is, any one of 18446744073709551616 different patterns.

Once the quantization procedure of dividing the display space into 64 discrete cells has been employed, the property of analyticity is lost, and in this particular instance, there is no advantage in maintaining the two-dimensional nature of the pattern. In what follows we represent such patterns by a sequence of 64 elements, i.e., each pattern by a 64 component vector as described in Section II.

The 2^{64} different vectors which can be drawn in this pattern space represent the 2^{64} different patterns which can be displayed by an eight by eight array. Of these 2^{64} different patterns, there are 64 patterns which are distinct from the rest, and they are, in fact, one set of orthonormal basis vectors which span this space. Many equivalent sets may be obtained by suitable transformation, but at least one set is provided by the Walsh functions of sequencies from zero to 63 which when reassembled in an 8 × 8 array, have the forms exhibited in Fig. 2.

This present technique can now be demonstrated graphically. For a single pair of pattern and associated reference, the recording procedure and the perfect regeneration of the associated reference are exhibited in Figs. 3 and 4.

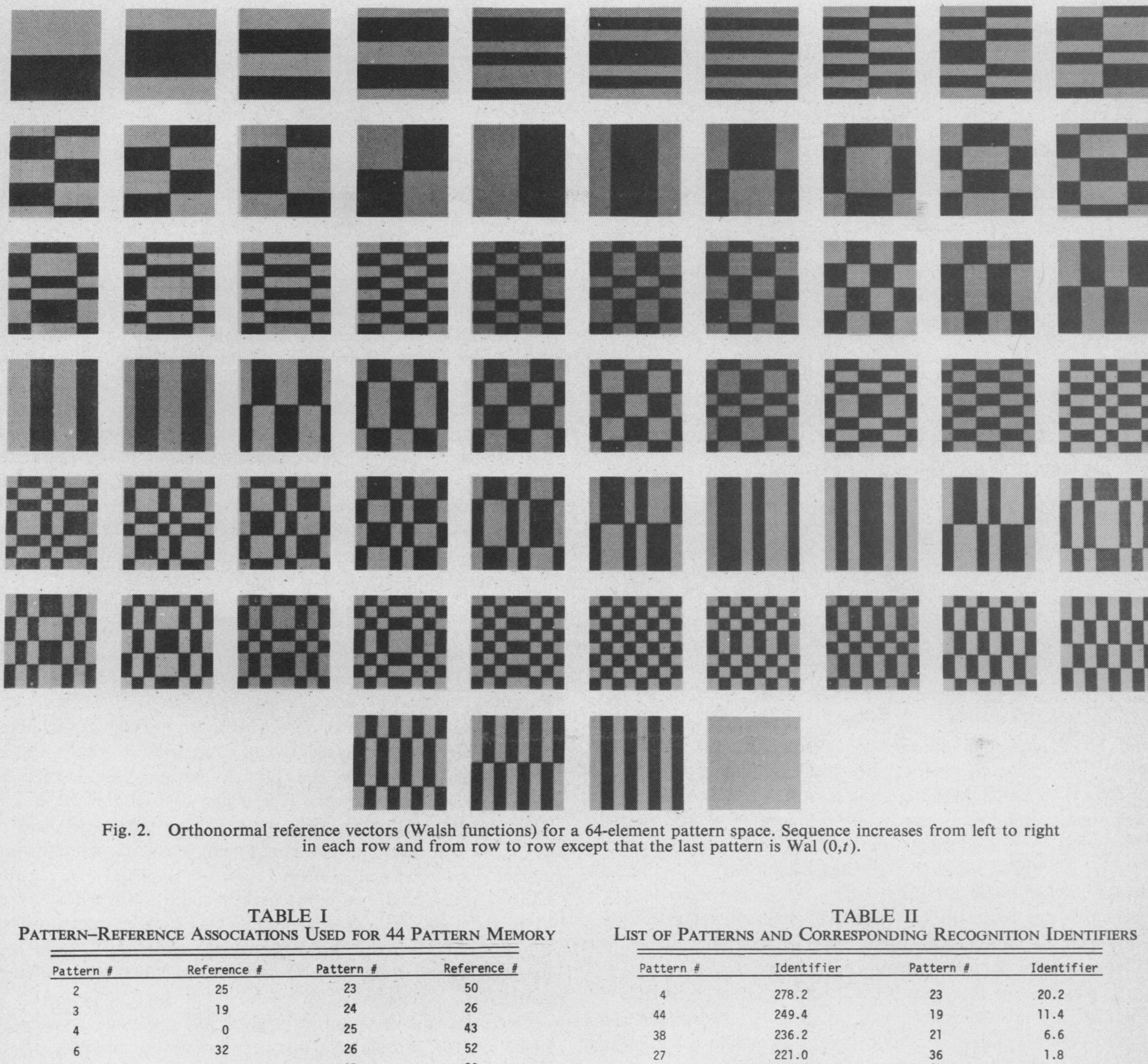

Fig. 2. Orthonormal reference vectors (Walsh functions) for a 64-element pattern space. Sequence increases from left to right in each row and from row to row except that the last pattern is Wal $(0,t)$.

TABLE I
PATTERN–REFERENCE ASSOCIATIONS USED FOR 44 PATTERN MEMORY

Pattern #	Reference #	Pattern #	Reference #
2	25	23	50
3	19	24	26
4	0	25	43
6	32	26	52
7	6	27	29
8	40	28	7
10	55	29	37
11	14	30	31
13	51	31	41
14	47	32	59
1	34	33	1
5	35	34	45
9	11	35	38
12	58	36	60
15	63	37	33
16	53	38	62
17	9	39	20
18	21	40	36
19	46	41	22
20	42	42	15
21	27	43	17
22	8	44	56

TABLE II
LIST OF PATTERNS AND CORRESPONDING RECOGNITION IDENTIFIERS

Pattern #	Identifier	Pattern #	Identifier
4	278.2	23	20.2
44	249.4	19	11.4
38	236.2	21	6.6
27	221.0	36	1.8
20	171.8	35	- 1.0
14	151.8	9	- 6.6
28	156.6	7	-11.4
8	143.8	17	-22.6
39	137.4	10	-23.8
30	118.2	37	-35.0
33	89.4	42	-40.2
31	79.8	12	-44.5
25,2	60.2	16	-47.4
13	59.0	43	-139.4
41	55.4	32	-142.2
22	52.6	29	-156.6
3,26	49.4	5	-171.8
40	48.6	6	-202.2
1	45.8	18	-221.0
24	45.0	15	-246.6
34	41.0		
11	37.0		

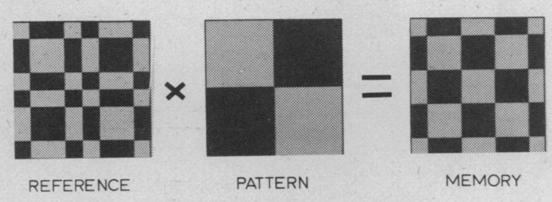

Fig. 3. Recording process.

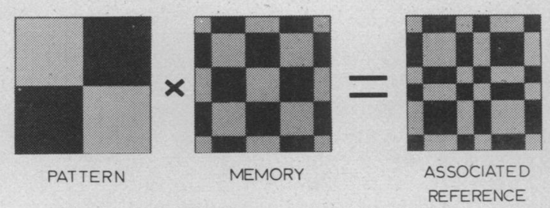

Fig. 4. Regeneration of associated reference.

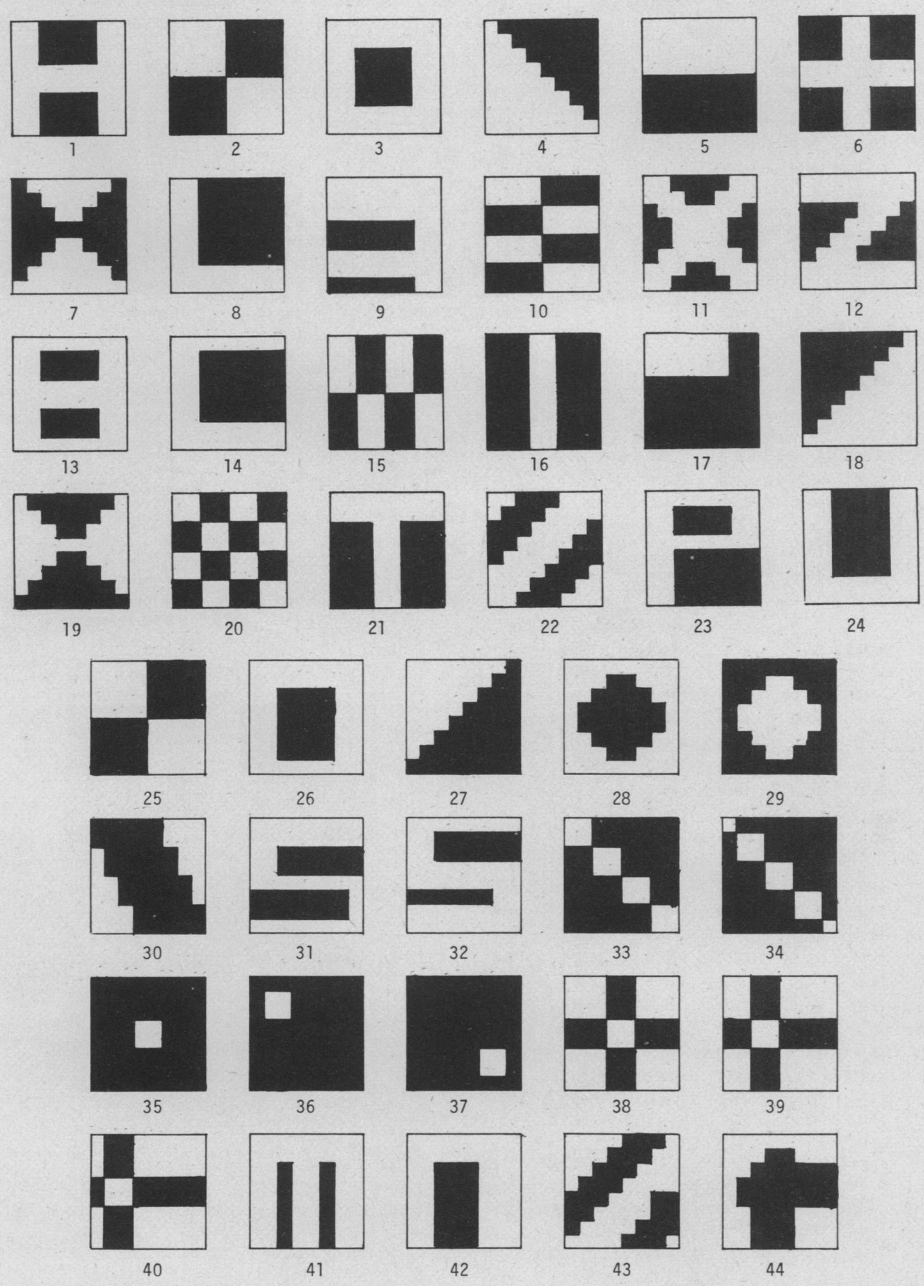

Fig. 5. Patterns used in computer simulation of associative memory.

1 Pattern Stored

	1/25
1/25	64.0

2 Patterns Stored

	1/25	3/19
1/25	64.0	16.0
3/19	16.0	64.0

3 Patterns Stored

	1/25	3/19	4/ 0
1/25	68.0	-4.0	4.0
3/19	8.0	80.0	0.0
4/ 0	4.0	4.0	84.0

4 Patterns Stored

	1/25	3/19	4/ 0	6/32
1/25	68.0	-4.0	-20.0	-4.0
3/19	8.0	80.0	8.0	-24.0
4/ 0	0.0	8.0	88.0	0.0
6/32	-12.0	-20.0	-4.0	68.0

5 Patterns Stored

	1/25	3/19	4/ 0	6/32	7/ 6
1/25	68.0	4.0	-12.0	-12.0	-52.0
3/19	8.0	80.0	8.0	-24.0	-8.0
4/ 0	-16.0	0.0	88.0	0.0	24.0
6/32	-12.0	-12.0	12.0	68.0	-4.0
7/ 6	-40.0	-8.0	40.0	0.0	64.0

6 Patterns Stored

	1/25	3/19	4/ 0	6/32	7/ 6	8/51
1/25	68.0	4.0	-12.0	-36.0	-52.0	4.0
3/19	8.0	80.0	8.0	-16.0	-8.0	16.0
4/ 0	-16.0	0.0	96.0	8.0	24.0	48.0
6/32	-12.0	-20.0	28.0	76.0	-4.0	4.0
7/ 6	-44.0	-20.0	44.0	20.0	52.0	-4.0
8/51	-20.0	44.0	36.0	-12.0	20.0	68.0

7 Patterns Stored

	1/25	3/19	4/ 0	6/32	7/ 6	8/51	9/40
1/25	68.0	4.0	-12.0	-32.0	-52.0	-8.0	0.0
3/19	8.0	80.0	8.0	-12.0	-8.0	4.0	36.0
4/ 0	-12.0	-4.0	100.0	16.0	20.0	40.0	-16.0
6/32	-20.0	-20.0	28.0	72.0	-4.0	0.0	-16.0
7/ 6	-52.0	-20.0	44.0	24.0	52.0	0.0	16.0
8/51	-20.0	44.0	36.0	-8.0	20.0	56.0	-8.0
9/40	-4.0	12.0	-12.0	-20.0	-12.0	-20.0	52.0

8 Patterns Stored

	1/25	3/19	4/ 0	6/32	7/ 6	8/51	9/40	11/55
1/25	68.0	4.0	-12.0	-32.0	-52.0	8.0	0.0	-8.0
3/19	8.0	95.0	8.0	-12.0	-8.0	-12.0	36.0	-28.0
4/ 0	-12.0	4.0	108.0	8.0	20.0	40.0	-24.0	40.0
6/32	-20.0	-44.0	28.0	72.0	-4.0	8.0	-16.0	-16.0
7/ 6	-52.0	-20.0	28.0	24.0	52.0	0.0	16.0	8.0
8/51	-20.0	44.0	44.0	-16.0	12.0	56.0	-8.0	8.0
9/40	-4.0	28.0	-12.0	-8.0	-12.0	-40.0	56.0	0.0
11/55	24.0	-40.0	8.0	-20.0	16.0	-28.0	4.0	92.0

9 Patterns Stored

	1/25	3/19	4/ 0	6/32	7/ 6	8/51	9.40	11/55	12/14
1/25	68.0	4.0	-12.0	-24.0	-44.0	8.0	-8.0	-8.0	-4.0
3/19	8.0	96.0	16.0	-12.0	08.0	-12.0	36.0	-28.0	16.0
4/ 0	-24.0	32.0	112.0	12.0	16.0	52.0	-20.0	36.0	-16.0
6/32	-20.0	-44.0	28.0	64.0	4.0	8.0	-16.0	-16.0	-20.0
7/ 6	-56.0	-16.0	32.0	12.0	48.0	-4.0	2υ.0	12.0	16.0
8/51	-24.0	64.0	64.0	-20.0	16.0	60.0	-4.0	4.0	-16.0
9/40	-8.0	16.0	-16.0	-8.0	-24.0	-40.0	56.0	0.0	40.0
11/55	24.0	-40.0	-8.0	12.0	8.0	-28.0	12.0	92.0	40.0
12/14	-12.0	20.0	36.0	-8.0	20.0	0.0	8.0	40.0	60.0

10 Patterns Stored

	1/25	3/19	4/ 0	6/32	7/ 6	8/51	9/30	11/55	12/14	13/47
1/25	68.0	4.0	-12.0	-24.0	-44.0	8.0	-8.0	-8.0	-4.0	56.0
3/19	8.0	96.0	16.0	-12.0	-8.0	-12.0	36.0	-28.0	16.0	60.0
4/ 0	-20.0	36.0	116.0	40.0	20.0	56.0	-32.0	40.0	-12.0	8.0
6/32	-28.0	-44.0	28.0	64.0	4.0	8.0	-16.0	-16.0	-20.0	-16.0
7/ 6	-48.0	-16.0	32.0	12.0	48.0	-4.0	28.0	12.0	8.0	-20.0
8/51	-24.0	64.0	64.0	4.0	16.0	68.0	-20.0	4.0	-16.0	20.0
9/40	-8.0	16.0	-16.0	-20.0	-24.0	-36.0	68.0	4.0	40.0	12.0
11/55	24.0	-40.0	-8.0	-12.0	8.0	-28.0	12.0	92.0	40.0	-4.0
12/14	-12.0	20.0	36.0	-8.0	20.0	8.0	8.0	24.0	60.0	32.0
13/47	52.0	36.0	20.0	-8.0	12.0	-8.0	-8.0	-24.0	12.0	88.0

Fig. 8. Response matrices of computer simulated distributed associative memories.

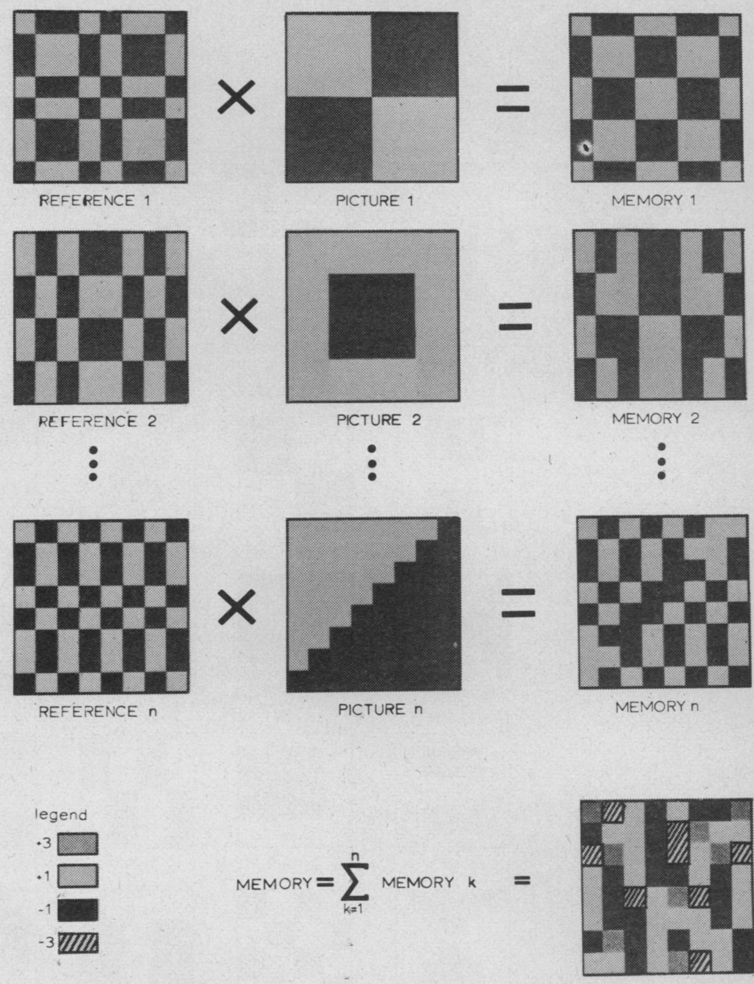

Fig. 6. Construction of memory containing many associated pairs of patterns and references.

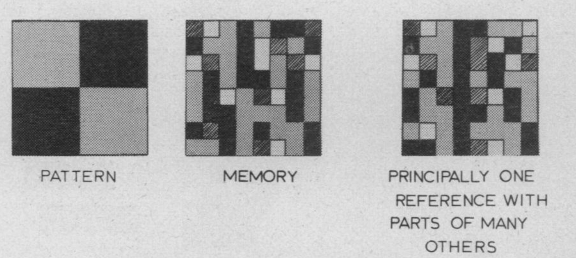

Fig. 7. Result of first stage of recognition act (for memory containing more than one pattern-reference pair).

In this study, 44 patterns were used, and these are shown in Fig. 5. The associations between the patterns and the references for these 44 patterns are listed in Table I. The successive formation of the individual memories and the total memory are depicted graphically in Fig. 6.

It is useful to emphasize that when a large number of patterns are stored, subsequent presentation of any one pattern for recognition does not regenerate the associated reference by itself, but does in fact yield a complex pattern which might be predominantly the associated reference. This situation is illustrated in Fig. 7. For any specific memory, these characteristics can be followed as the number of patterns stored is increased. The results for the 64 storage site memory in question are shown in matrix form in Fig. 8. The results are presented in a nonnormalized form. When five patterns are stored, and when pattern #6 (associated reference #32) is presented for recognition, the response is primarily the associated reference #2. As seen from the appropriate row entries, the response for the correct reference is 68 versus -12.0, -12.0, 12.0 and -4 for the other references. However, by the time nine patterns have been stored, presentation of pattern #8 (associated reference #51) results in significant values of the coefficients for reference #19 and for reference #0.

Examination of these characteristics indicates that if recognition depended on the associated pattern being regenerated predominantly, then this scheme would be of limited use indeed. We show however, that this is not the case. In fact, presentation of a pattern for recognition generates a "star" of reference vectors. It is not necessary that the star have the associated pattern as the predominant feature. It is only necessary that the stars regenerated in this manner are different from each other. The final act of recognition consists of taking the scalar product of a discriminant vector f and the star of reference vectors to yield a single number. To each pattern there is, then, a specific number, and this list of patterns and corresponding identifying number is the simple dictionary which is used in the recognition and retrieval of patterns.

Using the pattern-reference associations listed in Table I, an associative memory was prepared, and a discriminant vector consisting of a linear combination of the reference vectors was used for determining which pattern had been presented for recognition. This is to say that when x_q was presented for recognition,

$$f = 0.1y_1 + 0.2y_2 + 0.3y_3 \cdots 0.44y_{44}$$

was used to yield

$$\langle x_q m_f \rangle = D_q$$

a number. The 44 patterns and the numbers associated with them are listed in Table II.

IV. Significance of Results and Direction of Future Investigations

The results of the previous section indicate that an 8×8 memory can be used to store many (at least 44) patterns in parallel and that when any of these patterns is presented for recognition, only 3×44 processing steps are required to yield a number which then readily identifies which pattern had been presented for recognition. This is in contrast to the demands of the template-matching technique where 64×44 storage sites and about $3 \times 64 \times 44$ processing steps are required for recognition. These savings become very significant once memories of the order of 10^6 storage sites are contemplated.

This scheme has many interesting properties, some of which are now known and others need to be explored further. Of the patterns depicted in Fig. 5, we note that patterns #2 and #25, and #3 and #26 form identical pairs. Referring to Table II, we see that the identifying numbers also form pairs, even though different references were used. Thus this memory is effective in sorting out which patterns look alike, and this result is achieved regardless of the references used. Similarly, patterns #7 and #19, #18 and #27, and #28 and #29 form pairs which are "gray scale" inverted images of each other. It is interesting that the corresponding identifying numbers form pairs identical in absolute magnitude bit with opposite signs.

This proposed scheme either in software or hardware implementation may be used to serve as content addressable memories for patterns in major automated industrial systems. Presentation of part of a pattern would lead to recognition and subsequent automatic retrieval from storage all that is associated with the pattern and that needs to be recalled. The addressing is based on the content matter of the pattern so presented and not on any additional address. It would seem that this scheme has potential as an important element in all major information processing systems.

The discriminant vector f may be used to accentuate the importance of certain patterns or to group certain others together. The way to accomplish this is quite obvious when f is expressed as a linear expansion of the linearly independent reference vectors. Analogous techniques for circumstances when f is not a linear sum of the reference vectors and/or when the reference vectors do not form an orthonormal set of basis vectors need to be investigated further. There are indications that the characteristics of those memories are of substantial practical interest.

In addition to studies of software implementation of this scheme, a hardware implementation has also been initiated, and corresponding characteristics for memories of much larger storage capacity will become available in the future.

References

[1] L. D. Harmon, "Automatic recognition of print and script," *Proc. IEEE*, vol. 60, pp. 1165–1176, Oct. 1972.

[2] D. G. Willis, "Plastic neurons as memory elements," presented at 1959 Int. Conf. Information Processing, Paris, France.

[3] D. J. Willshaw, "Models of distributed associative memory," Ph.D. dissertation, University of Edinburgh, Edinburgh, Scotland, 1971.

[4] D. J. Willshaw, O. P. Buneman, and H. C. Longuet-Higgins, "Non-holographic associative memory," *Nature*, vol. 222, p. 960, 1969.

[5] H. C. Longuet-Higgins, D. J. Willshaw, and O. P. Buneman, "Theories of associative recall," *Quart. Rev., Biophys.*, vol. 3, p. 223, 1970.

[6] H. F. Harmuth, *Transmission of Information by Orthozonal Functions*. New York: Springer, 1969. (A background discussion of Walsh functions may be obtained from this reference.)

[7] C. R. Rosen, "An approach to a distributed memory," in *Principles of Self Organization*, Von Foerster and Zopf, Eds. New York: Pergamon, 1962.

Computer Editing of News Photos

DONALD E. TROXEL

Abstract—**A computer-based system for the editing of news photographs is described. Capabilities now implemented include cropping, enlarging, and the insertion of captions. A key element of this system is a display peripheral capable of posting a flicker-free display on a standard monochrome television monitor. A relatively small semiconductor memory of 2^{18} bits is used to store the picture information. The Roberts**

Manuscript received February 6, 1975; revised May 23, 1975. This work was supported by a Grant from the Associated Press.

The author is with the Research Laboratory of Electronics and the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Mass. 02139.