PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Shift Invariance In Optical Associative Memories

Psaltis, D., Hong, J., Venkatesh, S.

D. Psaltis, J. Hong, S. Venkatesh, "Shift Invariance In Optical Associative Memories," Proc. SPIE 0625, Optical Computing, (9 June 1986); doi: 10.1117/12.963498

SPIE.

Event: O-E/LASE'86 Symposium, 1986, Los Angeles, CA, United States

Shift Invariance in Optical Associative Memories

D.Psaltis, J.Hong, and S.Venkatesh
California Institute of Technology
Department of Electrical Engineering MS 116-81
Pasadena, Calif. 91125

Abstract

Shift invariance in associative memories is discussed. Two optical systems which exhibit shift invariance are described in detail along with computer simulations showing their effectiveness. It is shown, however, that shift invariance cannot be achieved with systems that employ only linear interconnections to store the associations without an accompanying decrease in the storage capacity equal to the number of shifted versions that are recognizable.

Introduction

An associative memory system is programmed to store input-output pairs of data so that when a certain input is recognized as one of the stored items, then the output associated with it is retrieved. Another key feature that can be incorporated into such memories is error correction which allows even input items that are distorted versions of any of the stored data to be recognized. In reference to Fig.1, an associative memory can then be described as a system which can be programmed to perform many to one mappings of inputs $\underline{f}_m + \delta\underline{f}$ into outputs $\underline{g}_m$ for m=1,2,,,,M, where the input error $\delta\underline{f}$ is small in some sense.

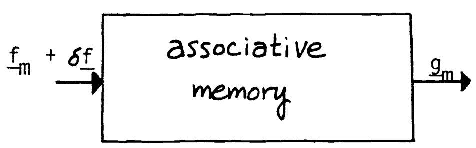


Fig.1 Associative Memory

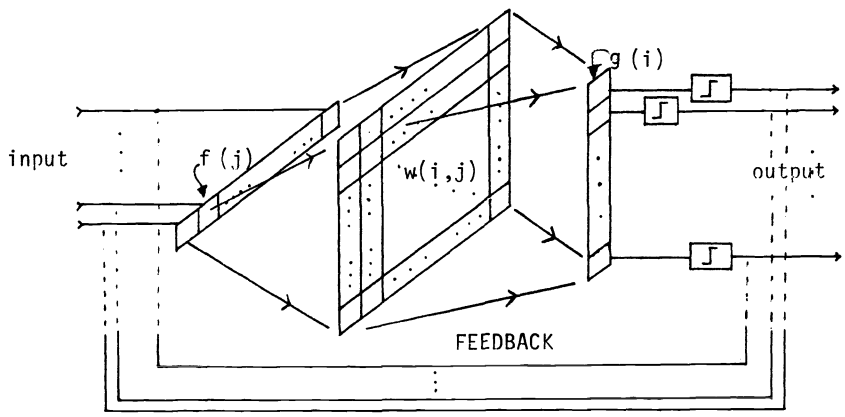


Fig.2 LDF Implementation of Associative Memory

If the $\underline{f}_m$ are real valued but the $\underline{g}_m$ binary, then the associative memory can be thought of as an array of discriminant functions where each output bit is the result of a binary decision based on the input vector. One particular implementation of such a memory system uses linear discriminant functions (LDF). The LDF of an N dimensional vector $\underline{f}$ is simply a weighted sum of the components of $\underline{f}$, and a pattern recognition machine based on such a function makes a binary decision based on this function in relation to a prescribed threshold θ. Specifically, the decision rule is given by

$$\sum_{i=1}^{N} w(i)f(i) - \theta \overset{?}{\gtrless} 0 \qquad (1)$$

An associative memory can be constructed that can store and recall $\underline{f}_m$ by producing an N-bit binary word $\underline{g}_m$ by configuring an array of N such decision machines. Shown in Fig.2 is the associative memory implemented in an optical vector matrix multiplier scheme[1]. For the special case of auto-association where an input $\underline{f}_m + \delta\underline{f}$ produces the corrected output $\underline{f}_m$, feedback can enhance the error correcting and robustness properties of ther system[2]. Since the weight matrix has N^2 adjustable coefficients w(i,j), the maximum number of bits that can be stored by the system is upper bounded by the capacity of N LDF's. By using the separating capacity of the LDF, the storage capacity, defined as the maximum number of pairs of N-bit vectors that can be associatively stored was shown to be ~ N[3].

Of the various effective methods for choosing the weight values[2,4], the outer product method is the simplest and can be easily implemented optically. The weight matrix w(i,j) for auto-association that results from this method is simply the sum of the outer products of the vectors to be stored. The weight matrix for storing M N-bit, binary, bipolar vectors f_m is given by

$$w(i,j) = \begin{cases} \sum_{m=1}^{M} f_m(i)\, f_m(j) & i \neq j \\ 0 & i = j \end{cases} \qquad (2)$$

The output just before the threshold gates is simply the product

$$g(i) = \sum_{j=0}^{N-1} w(i,j) f_{m0}(j) = \sum_{m=1}^{M} f_m(i) \sum_{j=0}^{N-1} f_m(j) f_{m0}(j) \qquad (3)$$

where, upon expansion and interchange of summations, the output response to an input f_{m0} is seen to be simply a weighted sum of the stored vectors, the weights being the inner products of the stored vectors with the input. As inner products can easily be computed optically, eqn.3 suggests implementations different from that shown in Fig.2[1]. Shown in Fig.3 is one such implementation. The first portion of the system is a one dimensional multi-channel correlator, and the slit samples the correlation functions at the zero-shift position to obtain the inner products. The latter portion uses these weights to illuminate appropriate portions of the second mask which contains the stored vectors in order to obtain the weighted sum of eqn.3. While the output threshold gates are the only nonlinearities found in the outer product scheme, the expanded system of Fig.3 allows the possibility of placing nonlinearities also in the correlation plane. It will be shown in later sections that this is indeed necessary if shift invariant (SI) operation is required.

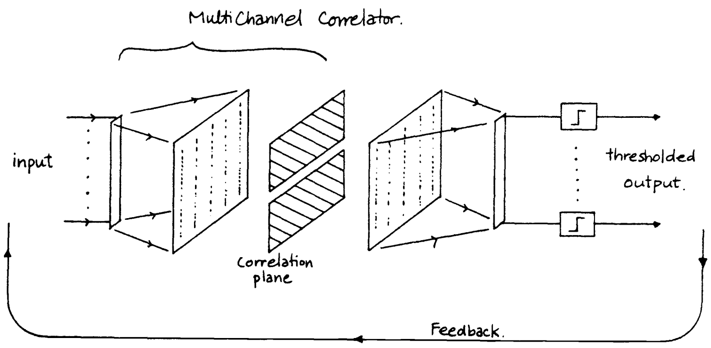


Fig.3 Optical Implementation of LDF Memory

Shift Invariance

In relation to associative memories, two types of shift invariance may be considered, as illustrated in Fig.4. The goal in both cases is to store a set of signals and retrieve them in response to any shifted version of those signals. The first type is the usual notion of shift invariance where the response to a shifted input is also shifted by a proportional amount. Note that in both cases, error correction is illustrated. The output of the second type always remains centered regardless of the shift occuring at the input.

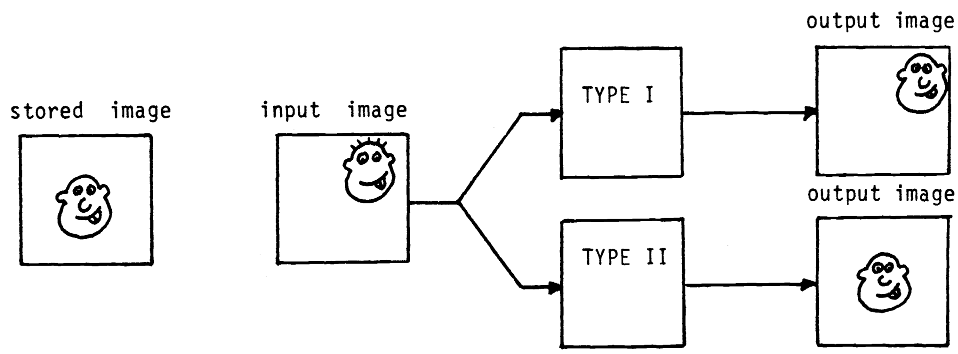


Fig.4 Shift Invariance

If each shifted version of the stored N-bit vectors is treated as a separate entity and shifts of up to N-bits are allowed at the input, then the storage capacity of such systems must be of the order of MN where M is the number of signals to be stored. The LDF memory described earlier, which uses linear interconnections to store information, has a maximum storage capacity of N' vectors where N' is the size of any input. Such a system can then store only 1 item with full shift invariance. That the storage capacity is 1 for both types of shift invariance can be shown with the following simple but fundamental arguments. Let h(x,x') be the interconnection kernel, and $\{f_m(x), g_m(x)\}$ be the pairs of data to be stored, with m=1,2,,,,M. The kernel must then satisfy the following set of M integral equations:

$$g_m(x) = \int h(x,x')f_m(x')dx' , \quad m=1,2,,,,M. \tag{4}$$

If shift invariance of type I is required, where the output shifts proportionately with the input, then the kernel must be a function only of the difference between its two arguments. With this, eqn.4 reduces to a set of convolution integrals which can be written in the Fourier domain as follows:

$$G_m(\omega) = H(\omega)F_m(\omega) , \quad m=1,2,,,M, \tag{5}$$

where $G_m(\omega)$, $H(\omega)$, and $F_m(\omega)$ are Fourier transforms of $g_m(x)$, $h(x)$, and $f_m(x)$, respectively. For arbitrary pairs $\{g_m, f_m\}$, $H(\omega)$ is completely determined by one pair of data, so that storage is limited to one association. Shift invariance of type II requires that the output $g_m(x)$ be the same for $f_m(x-\alpha)$, for all possible shifts α. Using eqn.3, this condition leads to an h(x,x') that satisfies:

$$\int h(x,x')f_m(x'-\alpha)dx' = \int h(x,x')f_m(x')dx'. \tag{6}$$

For arbitrary $f_m(x)$, $h(x,x'+\alpha) = h(x,x')$ for all α which can be true only if $h(x,x')=h(x)$. The kernel h(x) is completely determined by one pair of data resulting in a storage capacity of one association.

Optical Systems with Linear Interconnections

One of the first optical implementations of associative memories was proposed by Gabor[5] who pointed out that a hologram is really an association device that stores an object wave in relation to a reference wave and retrieves one wave in response to the other. Shown in Fig.5 are the recording and reconstruction of the holographic memory. The mutual interference pattern between the waves emanating from the objects 'A' and 'B' is recorded on a plane hologram in the Fresnel diffraction region. The developed holgram is illuminated by the object 'B' or a portion thereof, resulting in the reconstruction of the associated image 'A' at the output plane.

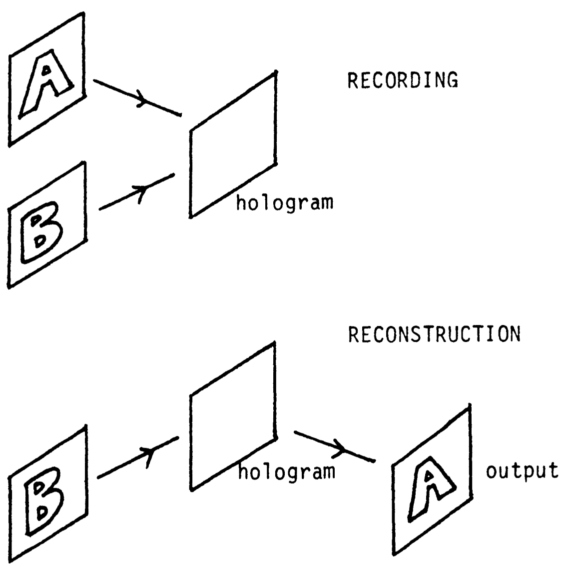


Fig.5 Gabor's Associative Hologram

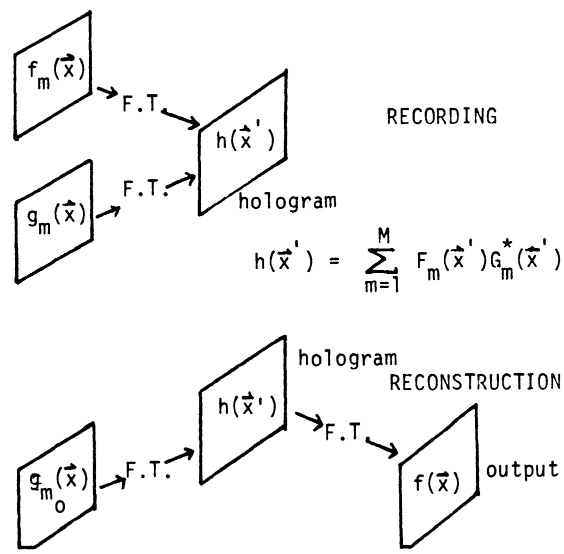


Fig.6 Shift Invariant Optical Memory (ineffective for M > 1)

An important issue is whether many such associations can be made using the same hologram. Shown in Fig.6 is a candidate implementation which is also SI. The hologram is prepared by superimposing the products of the Fourier transformed image pairs, $\{F_m(\omega), G_m(\omega)\}$. When one of the stored images $g_{m0}(x)$ is used as the input, the resulting output can be described by the following sum of triple product correlation-convolutions:

$$f(x) = \sum_{m=1}^{M} (g_{m0}(x) \circledast g_m(x)) * f_m(x) \qquad \circledast \text{ denotes correlation}$$
$$= (g_{m0} \circledast g_{m0})*f_{m0} + \sum_{m=1}^{M} (g_{m0} \circledast g_m)*f_m \qquad * \text{ denotes convolution}$$
$$\cong f_{m0} + \sum_{m=1}^{M} (g_{m0} \circledast g_m)*f_m, \qquad (7)$$

where the last step is justified if the autocorrelation function of $g_{m0}(x)$ is sharply peaked. Since the overall operation is a composition of correlation and convolution, the system is shift invariant and from eqn.7, we see that the output contains the correct recall term but with cross correlation noise. For analysis, it is useful to discretize the problem by dividing the input, hologram, and output planes each into N pixels where N is the space bandwidth product(SBP) of the hologram. For binary, bipolar signals whose pixel values are statistically independent, the signal to noise ratio (SNR) is calculated to be

$$SNR = \{E^2[f_{m0}(x)]/E[\alpha^2(x)]\}^{1/2} \cong (1/2M)^{1/2}. \qquad (8)$$

The poor SNR derives from the fact that the number of degrees of freedom available in the hologram, N, is insufficient to store more than 1 shifted version of 1 pair of images. It has been shown[6] that the SBP of the hologram must be multiplexed in some way among the associations to be stored for such systems to be effective; a particular scheme is to use volume holograms to multiplex the data, in effect, in the propagation direction.

Distributed Nonlinearities

Although the linearly interconnected systems are ineffective when shift invariance is required, the same is not necessarily true when nonlinearities are introduced into the interconnections themselves. Shown in Fig.7 is an architecture designed to achieve nonlinear interconnections, in this case for auto-association although it can be applied also for hetero-associations as well. The input signal is correlated with the stored memories $\{f_m(i)\}$, m=1,2,,,M. The resulting correlation functions are then processed using point nonlinearities. The next stage convolves the processed correlation functions with the corresponding memory signals, and the resulting outputs are then summed and thresholded. Since the overall operation is a composition of correlations and convolutions with uniform point nonlinearities, the system is SI and can be described by the following equation:

$$f_{out}(x) = \sum_{m=1}^{M} N.L.[fm(x) \circledast f_{m0}(x)] * fm(x), \qquad (9)$$

where N.L.[] indicates the nonlinear operation. If the nonlinearities in the correlation plane are bypassed, then the system becomes a linearly interconnected, shift invariant machine which is ineffective because the processing gain acquired through the correlators is distributed over the output by the convolution operations, resulting in output cross correlation noise that is as strong as the correctly recalled component. The purpose of the nonlinearities is then to enhance the processing gain that is present at the correlation plane.

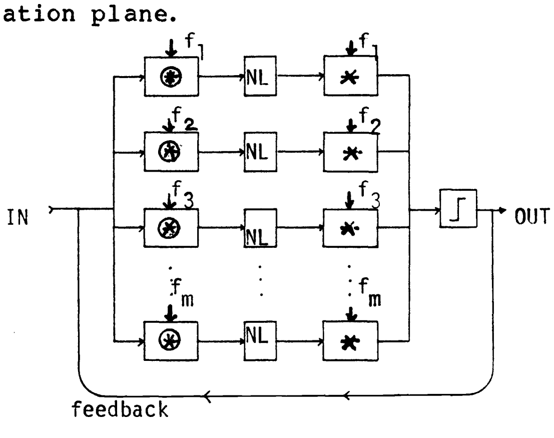


Fig.7 Distributed Nonlinearities System

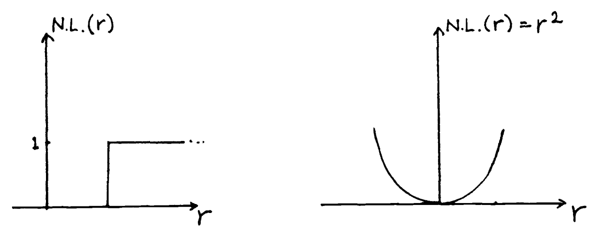


Fig.8 Threshold and Square-law N.L.

We consider two such nonlinearities as shown in Fig.8. If a threshold function is used, then for proper operation of the system, the threshold level must be automatically adjusted so as to always pick out only the maximum peak among the various correlations. If such an automatic threshold were used, the output threshold gates along with the feedback becomes unnecessary as only 1 of the stored memories would be read out. If however, the threshold level were fixed and several peaks exceed that level, the output signal term will be an equally weighted sum of several memory signals, resulting in a totally ambiguous recall. A nonlinearity that need not be adjusted adaptively for effective usage is the square-law. This is of special interest since the squaring operation is one that is achievable optically. The square-law nonlinearity increases the difference between the maximum of the autocorrelation term ($f_{m0} \circledast f_{m0}$) and the cross correlation terms. If M N-bit binary, bipolar signals $f_m(k)$ are stored and the input window allows 2N shifts (N bits in each direction), then the input-output equation can be written as

$$f(k) = \sum_{m=1}^{M} \sum_{j=1-N}^{N-1} R_m^2(j) f_m(k-j), \tag{10}$$

where $R_m(j) = \sum_{i=0}^{N-1} f_m(i) f_{m0}(i+j)$ is the correlation function. Eqn. 10 can be expanded to yield

$$f(k) = R_{m0}^2(0) f_{m0}(k) \quad + \quad \alpha(k)$$

$$= N^2 f_{m0}(k) \quad + \quad \alpha(k)$$

$$\alpha(k) = \sum_{\substack{j=1-N \\ j \neq 0}}^{N-1} R_{m0}^2(j) f_{m0}(k-j) \quad + \quad \sum_{\substack{m=1 \\ m \neq m_0}}^{M} \sum_{j=1-N}^{N-1} R_m^2(j) f_m(k-j). \tag{11}$$

The output SNR can then be computed to yield the result

$$SNR = N^2 / \{E[\alpha^2(k)]\}^{1/2} \cong (N/6M)^{1/2}. \tag{12}$$

Because of the squaring operation, we have gained a factor of $N^{1/2}$ in SNR over that of the linearly interconnected machine. For comparison, the SNR of the outer product memory which is not SI is $SNR=(N/M')^{1/2}$, where $M'_{max} \cong .15N$ is the maximum number of vectors that can be stored[2]. Comparing the two SNR expressions, it is clear that the capacity M of the SI system using the square-law is proportional to that of the outer product system. This is significant because the SI system actually stores M(2N-1) items if we count the shifted versions as separate entities. The operation of this system is illustrated by the simulation results shown in Figs.9a-9f. Two 32 bit, binary, bipolar vectors shown in the marked regions of Figs.9a and 9b were programmmed into the memory. The first line in each figure constitutes the particular input used and the subsequent lines contain the thresholded output vectors after each iteration; '+' indicates a +1 and '-' indicates a -1. Because of the inherent three valuedness of the data representation, namely +1, -1, and 0, a threshold with a 'dead zone' as shown in Fig.10 was used. The width of the dead zone was fixed for the simulations. Figs.9a and 9b show that the two programmed vectors are indeed fixed points of the system, and in particular, Fig.9b demonstrates the memory's shift invariance. Figs.9c-9f serve to illustrate the error correcting capability of the system and in each case the system converges to the correct result within 3 iterations. With a small decrease in performance (capacity and error correction), the system can be implemented with a strictly 2 level threshold with no dead zone by configuring the memory with a combination of unipolar and bipolar signals and using unipolar inputs. Also, the extension of this system to the storage of 2 dimensional images is straightforward and optically implementable; the discussion will appear elsewhere.

```
           VECTOR #1
0 )  -+++-++--++-+--+--++---+-++-+-+-000000000000000000000
1 )  -+++-++--++-+--+--++---+-++-+-+-000000000000000000000
2 )  -+++-++--++-+--+--++---+-++-+-+-000000000000000000000
3 )  -+++-++--++-+--+--++---+-++-+-+-000000000000000000000
4 )  -+++-++--++-+--+--++---+-++-+-+-000000000000000000000
5 )  -+++-++--++-+--+--++---+-++-+-+-000000000000000000000
```

a) input: #1 with no errors

```
                VECTOR #2
0 )  00000-+-+-++++-+-++-----+-++-++-++--+0000000000000000
1 )  00000-+-+-++++-+-++-----+-++-++-++--+0000000000000000
2 )  00000-+-+-++++-+-++-----+-++-++-++--+0000000000000000
3 )  00000-+-+-++++-+-++-----+-++-++-++--+0000000000000000
4 )  00000-+-+-++++-+-++-----+-++-++-++--+0000000000000000
5 )  00000-+-+-++++-+-++-----+-++-++-++--+0000000000000000
```

b) input: #2 with no errors, 5 bits shift

```
0 )  -++++++--++-+-++-+++---++++-+-+-000000000000000000000
1 )  -+++-0+--++000-+0-++---+-++-+-+-000000000000000000000
2 )  -+++-++--++-+--+--++---+-++-+-+-000000000000000000000
3 )  -+++-++--++-+--+--++---+-++-+-+-000000000000000000000
4 )  -+++-++--++-+--+--++---+-++-+-+-000000000000000000000
5 )  -+++-++--++-+--+--++---+-++-+-+-000000000000000000000
```

c) input: #1 with 4 errors

```
0 )  -+++-++--++-+--+--++---+-++-+-+-0-00000-0000+0000+000
1 )  -+++-++--++-+--+--++---+-++-+-+-000000000000000000000
2 )  -+++-++--++-+--+--++---+-++-+-+-000000000000000000000
3 )  -+++-++--++-+--+--++---+-++-+-+-000000000000000000000
4 )  -+++-++--++-+--+--++---+-++-+-+-000000000000000000000
5 )  -+++-++--++-+--+--++---+-++-+-+-000000000000000000000
```

d) input: #1 with 4 errors

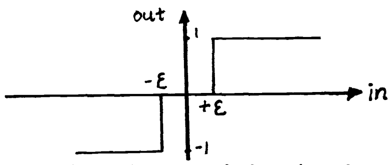
```
0 )  -+-+-++++-+-++-----+-++-++-++--+0-00000-0000+0000+000
1 )  -+-+-++++-+-++-----+-++-++-++--+000000000000000000000
2 )  -+-+-++++-+-++-----+-++-++-++--+000000000000000000000
3 )  -+-+-++++-+-++-----+-++-++-++--+000000000000000000000
4 )  -+-+-++++-+-++-----+-++-++-++--+000000000000000000000
5 )  -+-+-++++-+-++-----+-++-++-++--+000000000000000000000
```

e) input: #2 with 4 errors

```
0 )  -+++-++--++-++++---+---+-++---+-0-00000-0000+0000+000
1 )  -+++-++--++-+--+--++---+-++-00+-000000000000000000000
2 )  -+++-++--++-+--+--++---+-++-+-+-000000000000000000000
3 )  -+++-++--++-+--+--++---+-++-+-+-000000000000000000000
4 )  -+++-++--++-+--+--++---+-++-+-+-000000000000000000000
5 )  -+++-++--++-+--+--++---+-++-+-+-000000000000000000000
```

f) input: #1 with 8 errors

Fig.9a-f Computer Simulation Results for Square-Law System (input errors marked by ↓)



ε is fixed.

Fig.10 Threshold with Fixed 'Dead Zone'

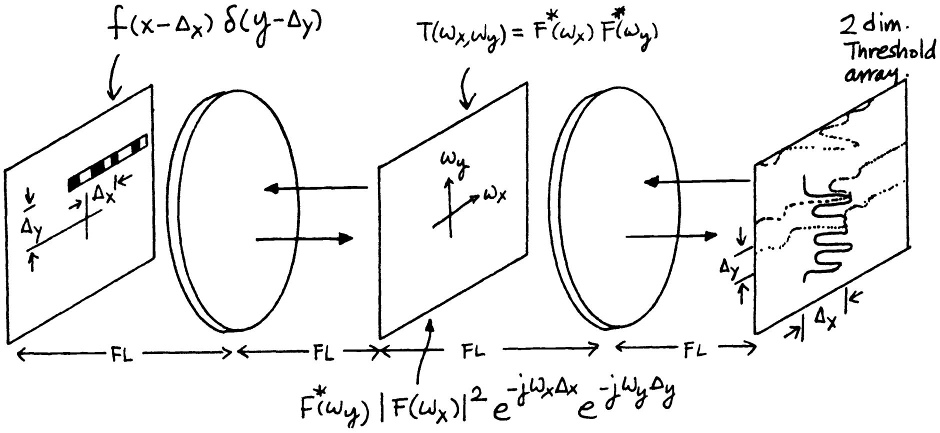


Fig.11 One Dim. Data Storage with Two Dim. Shift Invariance

Two-Dimensional Shift Invariance for One-Dimensional Data

We now discuss a novel method of storing and recalling 1 dimensional data with shift invariance in both dimensions. Shown in Fig.11 is the basic system and for illustration, the storage and recall of only 1 item is shown. The mask whose transmittance function is given by the outer product $T(\omega_x,\omega_y)=F^*(\omega_x)F^*(\omega_y)$ is placed in the Fourier transform plane with respect to both input and output planes of the memory vector. The input is a replica of the memory vector with shifts Δ_x and Δ_y in the x and y directions, respectively. The input expressed by $f(x-\Delta_x)\delta(y-\Delta_y)$ is Fourier transformed and multiplied by $T(\omega_x,\omega_y)$ to give

$$\tilde{F}(\omega_x,\omega_y) = F^*(\omega_y)|F(\omega_x)|^2\exp[-j(\omega_x\Delta_x + \omega_y\Delta_y)]. \tag{13}$$

The resultant product $\tilde{F}(.)$ is then Fourier transformed to give at the output plane

$$f_{out}(x,y) = R(x+\Delta_x)f(y+\Delta_y),$$

$$R(x) = \int f(\alpha)f(\alpha-x)d\alpha. \tag{14}$$

$R(x)$ is the autocorrelation function of $f(x)$. This output is thresholded and the resulting 2-dimensional binary signal is fed back into the system by reflection. Since the mask function is symmetric with respect to an interchange of its two arguments, the result of the feedback operation places the resulting data centered at the position where the input data originally appeared, as illustrated in Fig.11. In the forward operation, the x-shift is encoded into a horizontal shift of the autocorrelation function while the y-shift appears as a vertical shift in the reconstructed signal. Because the shifts have been encoded in this way, many such associations can be made by simply adding the corresponding terms to the mask function $T(\omega_x,\omega_y)$, which becomes

$$T(\omega_x,\omega_y) = \sum_{m=1}^{M} F_m^*(\omega_x)F_m^*(\omega_y). \tag{15}$$

If the system is interrogated with an input $f_{in}(x,y)=f_{m0}(x-\Delta_x)\delta(y-\Delta_y)$, then the output can be written as a sum of the correct recall term with the proper shifts and a cross correlation noise term as follows:

$$f_{out}(x,y) = R_{m0}(x+\Delta_x)f_{m0}(y+\Delta_y) + \sum_{m=1}^{M} R_m(x+\Delta_x)f_m(y+\Delta_y),$$

$$R_m(x) = \int f_m(\alpha)f_{m0}(\alpha+x)d\alpha. \tag{16}$$

Considering a system that stores M N-bit, binary, bipolar vectors whose individual bits are statistically independent, the output SNR can be calculated to be

$$SNR \cong N/(NM+(M-1)^2)^{1/2} \sim (N/M)^{1/2}, \qquad (17)$$

where the last approximation is valid for very large N. Eqn.17 shows that the capacity will approach that of the conventional outer product scheme while maintaining shift invariance. The useful capacity is gained at the cost of requiring a 2-dimensional array of threshold gates at the output plane.

Conclusion

Shift invariance in relation to associative memories was discussed. Without special encoding techniques, associative memories which use linear interconnections were shown to be ineffective for shift invariant operation. Two optical systems were introduced, one using nonlinear interconnections through the use of a square law nonlinearity in the correlation plane and the other using a novel encoding scheme. Both systems are shift invariant and their performances were shown to approach that of the outer product scheme which is not shift invariant.

Acknowledgements

We wish to thank our colleagues Dr. E. Paek, Mr. C.H. Park, Mr. F. Mok, and Mr. J. Yu with whom we have had helpful discussions. This work was supported by DARPA. One of the authors, Mr. Hong, was supported in part by a fellowship from Caltech's Program in Advanced Technologies, supported by Aerojet General, General Motors, GTE, and TRW.

References

1. D. Psaltis and N. Farhat, 'Optical information processing based on an associative-memory model of neural nets with thresholding and feedback,' Optics Letters, vol.10, pp.98-100. Jan.1985.

2. J.J. Hopfield, 'Neural networks and physical systems with emergent collective computational abilities,' Proc.Natl.Acad.Sci., USA, vol.79, pp.2554-2558, April 1982.

3. Y.S. Abu-Mostafa and J.M. St.Jacques, 'Information capacity of the hopfield model,' IEEE Trans.Info.Th., vol.IT-31, no.4, pp.461-464, July 1985.

4. T. Kohonen, Associative Memory: A System-Theoretical Approach, Springer Verlag, Berlin, 1977

5. D. Gabor, 'Associative holographic memory,' IBM J.Res.Devel., vol.13, pp.156-159, 1969.

6. D. Psaltis and J. Hong, in preparation.