Composite matched filtering with error correction

Robert J. Marks II and Les E. Atlas

Interactive Systems Design Laboratory, Department of Electrical Engineering, FT-10, University of Washington, Seattle, Washington 98195

Received March 24, 1986; accepted November 7, 1986

Information theory shows that one can communicate with arbitrarily low error probability over a noisy channel. By recognizing that some types of computations can be cast as communications problems, it may be possible to compute accurately with an inexact processor. Traditional analog optical processors, for example, offer advantages of parallelism and speed but suffer from significant inaccuracies. We propose an algorithm whereby the accuracy of matched filter processors can be improved significantly at the cost of a modest increase in computational resources. Errors that are due to noisy data and/or inexact computing can be detected and in some cases corrected.

Inaccuracy has been a major obstacle to the widespread adoption of optical processors. By trading dynamic range for accuracy, *digital* optical computing is a promising solution to this problem.[1,2] By using coding concepts from communications theory we propose a combination of analog processing and digital coding to increase accuracy with a modest sacrifice in throughput.

In his cornerstone papers on information theory, Shannon[3] demonstrated that one could communicate with arbitrarily low error probability over a noisy channel. In the same spirit, by using standard block coding techniques[4] it may be possible to compute accurately with an inexact processor. Indeed, we demonstrate that techniques used to communicate with a high degree of accuracy over a noisy channel can be applied directly to increasing the precision of an inexact composite matched filter[5] (CMF) processor.[6–8] In a CMF processor, an input, $\mathbf{g}$, represents one of N possible library elements. When CMF's are used, the processor output is a binary vector, which, when decoded (with error correction capability), will specify the appropriate library element. With 26 CMF's, we can recognize each of more than 10^6 library elements with single-error correction capability. The need for error correction and self-healing in optical processing has been addressed previously.[9,10]

To illustrate the CMF, assume that we have $N = 16$ real object vectors of length $L : \{\mathbf{f}_n | 0 \leq n < N = 16\}$. For the moment, assume that the elements are orthonormal. That is, $\mathbf{f}_n{}^T\mathbf{f}_m = \delta[n - m]$, where $\delta[p]$, the Kronecker delta function, is 1 for $p = 0$ and is otherwise 0. This orthonormality constraint will be removed later.

In Table 1 there are four rows of numbers from 1 to 15. The s_0 row contains numbers that, in binary, contain a 1 in the least significant bit. Row s_1 contains those numbers containing 1 in the second least significant bit, etc. Generalizing, row s_{q-1} contains those numbers that, in binary, have a 1 in the qth least significant bit. Thus $(9)_{10} = (1001)_2$ is contained in the first and last rows. Clearly, similar tables can be formulated for numbers between 0 and $2^Q - 1$, where Q is an arbitrary positive integer.

We formulate the CMF's ($\mathbf{h}_m$; $m = 0, 1, 2, 3$) as sums of vectors with indices in accordance with Table 1:

$$\begin{aligned}
\mathbf{h}_0 &= \mathbf{f}_1 + \mathbf{f}_3 + \mathbf{f}_5 + \mathbf{f}_7 + \mathbf{f}_9 + \mathbf{f}_{11} + \mathbf{f}_{13} + \mathbf{f}_{15},\\
\mathbf{h}_1 &= \mathbf{f}_2 + \mathbf{f}_3 + \mathbf{f}_6 + \mathbf{f}_7 + \mathbf{f}_{10} + \mathbf{f}_{11} + \mathbf{f}_{14} + \mathbf{f}_{15},\\
\mathbf{h}_2 &= \mathbf{f}_4 + \mathbf{f}_5 + \mathbf{f}_6 + \mathbf{f}_7 + \mathbf{f}_{12} + \mathbf{f}_{13} + \mathbf{f}_{14} + \mathbf{f}_{15},\\
\mathbf{h}_3 &= \mathbf{f}_8 + \mathbf{f}_9 + \mathbf{f}_{10} + \mathbf{f}_{11} + \mathbf{f}_{12} + \mathbf{f}_{13} + \mathbf{f}_{14} + \mathbf{f}_{15}. \quad (1)
\end{aligned}$$

Note that

$$\mathbf{f}_n{}^T\mathbf{h}_m = \hat{\Delta}_{np} \equiv \begin{cases} 1 & n \in s_p \\ 0 & n \notin s_p \end{cases}. \quad (2)$$

For example, if $\mathbf{f}_9$ is input to the processor, the inner products are $\mathbf{f}_9{}^T\mathbf{h}_0 = 1$, $\mathbf{f}_9{}^T\mathbf{h}_1 = 0$, $\mathbf{f}_9{}^T\mathbf{h}_2 = 0$, and $\mathbf{f}_9{}^T\mathbf{h}_3 = 1$. The result, then, is $(1001)_2 = (9)_{10}$, which is the index of the input. In general, for an input of $\mathbf{f}_n$ the inner products of the CMF's will yield the binary equivalent of n. Thus, instead of $N = 16$ matched filters, we require $\log_2 N = 4$ CMF filters.

Independent of the number of objects, a single additional CMF will permit single-bit error detection. For our example, we choose an additional vector $\mathbf{d}_4$ so that the number of 1's at the output will always be even (i.e., even 1's parity). Since $(1)_{10} = (0001)_2$ contains an odd number of 1's, we must include $\mathbf{f}_1$ in $\mathbf{d}_4$. On the other hand, $(3)_{10} = (0011)_2$ already contains an even number of 1's. Thus $\mathbf{f}_3$ is not included in $\mathbf{d}_4$. In general, the CMF $\mathbf{d}_4$ is the sum of all object vectors whose indices, when represented in binary, contain an odd number of ones:

$$\mathbf{d}_4 = \mathbf{f}_1 + \mathbf{f}_2 + \mathbf{f}_4 + \mathbf{f}_7 + \mathbf{f}_8 + \mathbf{f}_{11} + \mathbf{f}_{13} + \mathbf{f}_{14}.$$

If, for example, we find that $\mathbf{f}_n{}^T\mathbf{h}_0 = 0$, $\mathbf{f}_n{}^T\mathbf{h}_1 = 1$, $\mathbf{f}_n{}^T\mathbf{h}_2 = 1$, $\mathbf{f}_n{}^T\mathbf{h}_3 = 0$, and $\mathbf{f}_n{}^T\mathbf{d}_4 = 1$, we know immediately that there has been a mistake made since the number

Table 1. Coding Map for Four CMF's

s_0	1	3	5	7	9	11	13	15
s_1	2	3	6	7	10	11	14	15
s_2	4	5	6	7	12	13	14	15
s_3	8	9	10	11	12	13	14	15

0146-9592/87/020135-03$2.00/0

of 1's is odd. Note that it is possible to have two-bit errors go undetected.

Using additional CMF's, we can detect and *correct* bit errors. The example presented here is based on Hamming error-correction coding.[4] To illustrate single-bit correction, we first reorganize the labeling of the four CMF's in Eq. (1):

$$\begin{aligned}
\tau_0 &= \mathbf{t}_1, \\
\tau_1 &= \mathbf{t}_2, \\
\mathbf{h}_0 &= \mathbf{t}_3 = \mathbf{f}_1 + \mathbf{f}_3 + \mathbf{f}_5 + \mathbf{f}_7 + \mathbf{f}_9 + \mathbf{f}_{11} + \mathbf{f}_{13} + \mathbf{f}_{15}, \\
\tau_2 &= \mathbf{t}_4, \\
\mathbf{h}_1 &= \mathbf{t}_5 = \mathbf{f}_2 + \mathbf{f}_3 + \mathbf{f}_6 + \mathbf{f}_7 + \mathbf{f}_{10} + \mathbf{f}_{11} + \mathbf{f}_{14} + \mathbf{f}_{15}, \\
\mathbf{h}_2 &= \mathbf{t}_6 = \mathbf{f}_4 + \mathbf{f}_5 + \mathbf{f}_6 + \mathbf{f}_7 + \mathbf{f}_{12} + \mathbf{f}_{13} + \mathbf{f}_{14} + \mathbf{f}_{15}, \\
\mathbf{h}_3 &= \mathbf{t}_7 = \mathbf{f}_8 + \mathbf{f}_9 + \mathbf{f}_{10} + \mathbf{f}_{11} + \mathbf{f}_{12} + \mathbf{f}_{13} + \mathbf{f}_{14} + \mathbf{f}_{15}.
\end{aligned} \tag{3}$$

The three additional vectors, $\mathbf{t}_1$, $\mathbf{t}_2$, and $\mathbf{t}_4$, are error-correcting CMF's and have yet to be specified.

To determine the error-correcting CMF's, we make use of Table 2, which is formulated exactly like Table 1 except that the maximum index number is 7. To determine the first error-correction CMF, $\tau_0 = \mathbf{t}_1$, we consider from the table the CMF's $\mathbf{t}_1$, $\mathbf{t}_3$, $\mathbf{t}_5$, and $\mathbf{t}_7$. We require that each $\mathbf{f}_n$ appear in these four CMF's an even number of times. $\mathbf{f}_1$, for example, already appears in $\mathbf{t}_3$ but not in $\mathbf{t}_5$ and $\mathbf{t}_7$. Thus, to make the number of appearances even, we require that $\mathbf{t}_1$ contain $\mathbf{f}_1$. $\mathbf{f}_7$, on the other hand, appears in $\mathbf{t}_3$ and $\mathbf{t}_5$ but not in $\mathbf{t}_7$. Since its number of appearances is already even, $\mathbf{f}_7$ is not included in $\mathbf{t}_1$.

The elements of the second error-correction CMF, $\tau_1 = \mathbf{t}_2$, are similarly determined, except that, as is given in Table 2, we consider only the CMF's $\mathbf{t}_2$, $\mathbf{t}_3$, $\mathbf{t}_6$, and $\mathbf{t}_7$. Continuing, our final result is

$$\begin{aligned}
\tau_0 = \mathbf{t}_1 &= \mathbf{f}_1 + \mathbf{f}_2 + \mathbf{f}_5 + \mathbf{f}_6 + \mathbf{f}_8 + \mathbf{f}_{11} + \mathbf{f}_{12} + \mathbf{f}_{15}, \\
\tau_1 = \mathbf{t}_2 &= \mathbf{f}_1 + \mathbf{f}_3 + \mathbf{f}_4 + \mathbf{f}_6 + \mathbf{f}_8 + \mathbf{f}_{10} + \mathbf{f}_{13} + \mathbf{f}_{15}, \\
\mathbf{t}_3 &= \mathbf{f}_1 + \mathbf{f}_3 + \mathbf{f}_5 + \mathbf{f}_7 + \mathbf{f}_9 + \mathbf{f}_{11} + \mathbf{f}_{13} + \mathbf{f}_{15}, \\
\tau_2 = \mathbf{t}_4 &= \mathbf{f}_2 + \mathbf{f}_3 + \mathbf{f}_4 + \mathbf{f}_5 + \mathbf{f}_8 + \mathbf{f}_9 + \mathbf{f}_{14} + \mathbf{f}_{15}, \\
\mathbf{t}_5 &= \mathbf{f}_2 + \mathbf{f}_3 + \mathbf{f}_6 + \mathbf{f}_7 + \mathbf{f}_{10} + \mathbf{f}_{11} + \mathbf{f}_{14} + \mathbf{f}_{15}, \\
\mathbf{t}_6 &= \mathbf{f}_4 + \mathbf{f}_5 + \mathbf{f}_6 + \mathbf{f}_7 + \mathbf{f}_{12} + \mathbf{f}_{13} + \mathbf{f}_{14} + \mathbf{f}_{15}, \\
\mathbf{t}_7 &= \mathbf{f}_8 + \mathbf{f}_9 + \mathbf{f}_{10} + \mathbf{f}_{11} + \mathbf{f}_{12} + \mathbf{f}_{13} + \mathbf{f}_{14} + \mathbf{f}_{15}.
\end{aligned} \tag{4}$$

We now illustrate single-bit error correction. If $\mathbf{f}_9$ is input, the result is

$$\mathbf{f}_9{}^T\,[\mathbf{t}_1|\mathbf{t}_2|\mathbf{t}_3|\mathbf{t}_4|\mathbf{t}_5|\mathbf{t}_6|\mathbf{t}_7] = [0011001]. \tag{5}$$

Table 2. Error-Correction Coding Map for Four CMF's

τ_0	1	3	5	7
τ_1	2	3	6	7
τ_2	4	5	6	7

Suppose, however, that, because of input or processor noise, there were a single-bit error and, instead of the binary vector in Eq. (5), we had as an output

$$[0011101]. \tag{6}$$

Referring again to row τ_0 in Table 2, we see that there is a mistake in the output of at least one of the CMF's $\mathbf{t}_1$, $\mathbf{t}_3$, $\mathbf{t}_5$, or $\mathbf{t}_7$. Why? Because these CMF outputs contain an odd number of 1's. By design, the number of 1's should be even. The bits for row τ_1, on the other hand, are even parity, and there is no apparent mistake at the outputs of CMF's $\mathbf{t}_2$, $\mathbf{t}_3$, $\mathbf{t}_6$, and $\mathbf{t}_7$. There is, however, a second mistake corresponding to row τ_2. Using a 1 for a detected error and 0 otherwise, we conclude that our bit error occurred at the output of CMF number $(101)_2 = 5$. Equivalently, $\mathbf{f}_5$ is the only element that is in τ_0 and τ_2 but not in τ_1. Comparing Eq. (5) with expression (6) substantiates our result.

The parameters of our single-bit error correction revolve around M = number of error-correction CMF's. For the example above, $M = 3$. The total number of CMF's, including error-correction CMF's, is $P = 2^M - 1$. The total number of (nonerror-correcting) CMF's that would be used if there were no coding is $Q = P - M = 2^M - M - 1$. Finally, the number of objects cataloged for detection is $N = 2^Q$. As we claimed above, for $N = 10^6$ objects a total of $P = 26$ CMF's (corresponding to $M = 6$) is needed for single-bit error correction. However, in the presence of either input or processor noise, the uncertainty (variance) of the processor output increases with the number of library elements. The effect of this increased uncertainty has yet to be studied.

Here we extend the error-correction results above to the case when the N objects are not orthogonal. For N objects, we form the P CMF's

$$\mathbf{s}_p = \sum_{n=1}^{N} a_{pm}\mathbf{f}_n \qquad 1 \le p \le P, \tag{7}$$

where the a_{pm} coefficients are determined by

$$\mathbf{s}_p{}^T\mathbf{f}_m = \begin{cases} 1 & \mathbf{f}_m \in t_p \\ 0 & \mathbf{f}_m \notin t_p \end{cases} = \hat{\Delta}_{pm}. \tag{8}$$

As an example, for $M = 3$ ($P = 7, N = 16$), the matrix $\hat{\Delta}$ containing the elements $\hat{\Delta}_{pm}$ is, from Eqs. (4),

$$\hat{\Delta} = \begin{bmatrix}
0&1&1&0&0&1&1&0&1&0&0&1&1&0&0&1 \\
0&1&0&1&1&0&1&0&1&0&1&0&0&1&0&1 \\
0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1 \\
0&0&1&1&1&1&0&0&1&1&0&0&0&0&1&1 \\
0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1 \\
0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1 \\
0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1
\end{bmatrix}. \tag{9}$$

Substituting Eq. (7) into Eq. (8) gives

$$\sum_{n=1}^{N} a_{pn} r_{nm} = \hat{\Delta}_{pm}, \tag{10}$$

where $r_{nm} = \mathbf{f}_m{}^T\mathbf{f}_n$. In matrix form, Eq. (10) can be written as $\mathbf{AR} = \hat{\mathbf{\Delta}}$, where $\mathbf{R}$ is the $N \times N$ correlation matrix. The a_{pm} coefficients can thus be found from

$$\mathbf{A} = \hat{\mathbf{\Delta}}\mathbf{R}^{-1}.$$

Use of these coefficients in Eq. (7) will yield $\mathbf{s}_p{}^T\mathbf{f}_n = 1$ when $\mathbf{f}_n \in t_p$ and is zero otherwise. The same single-bit error-correction algorithm previously presented is applicable. Note that when the object vectors are orthonormal, this algorithm reduces to that above.

To illustrate CMF single-error correction, we performed a Monte Carlo simulation for $N = L = 16$. The library entry $\mathbf{f}_m$ had a 1 as its $m + 1$st element $(0 \leq m \leq 15)$ and was otherwise 0. Zero-mean white Gaussian noise was added to each element of the input and each element of the $\mathbf{A} = \hat{\mathbf{\Delta}}$ matrix. The respective noise variances are denoted by $\sigma_F{}^2$ and $\sigma_A{}^2$. For a given noisy input, detection was also performed without error correction by examining only the original four CMF outputs.

The fraction of correct results (with and without the error-correction processor) versus σ_F is shown in Fig. 1. Each of the four curves consists of six linearly interpolated points. Each point was generated by 800 simulations. The dashed curves are for single-error correction, and the solid lines are for no error correction. The vertical lines represent 90% confidence intervals computed under the assumption that each simulation was a Bernoulli trial.[11] Clearly, in each case error correction improved performance.

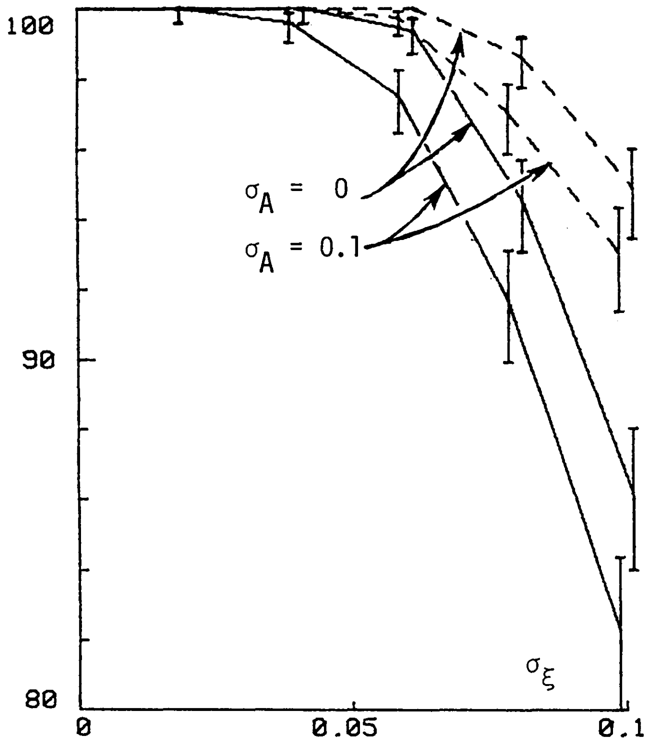

Fig. 1. Monte Carlo simulation of CMF performance without (solid line) and with (dashed line) error correction. The plots are percent of correct results versus input noise standard deviation σ_F. Error bars represent 90% confidence intervals.

Processor inaccuracies are a fundamental problem for optical computers. We have illustrated a technique whereby accuracy can be increased by a modest expansion of throughput. For example, as shown for $N = 10^6$ objects, increasing the number of CMF's by 30% can allow for single-bit error correction. This behavior is in contrast to digital techniques in which dynamic range is sacrificed. Ideally, any technique to increase the accuracy of optical processors should be formulated in such a manner as to preserve the computational advantages inherent in the processor physics.

We believe that the techniques of correcting errors in CMF's can also be applied to problems of very large wafer scale integration in which errors can result from a distribution of defects. The advantage of this technique over those conventionally used[12,13] is that the overhead of special control or testing would be greatly reduced. The unavoidable occurrence of defects would cause processing errors, yet the error-correction capability would reduce or eliminate these errors with only a small amount of added hardware and computation. The effects of generalizing the error-correction algorithm to multilevel and multiple errors still needs investigation, as does performance comparison between the resulting CMF's and the (sometimes optimal[14]) conventional matched filter.

The authors thank James Ritcey and Kwan Cheung for their stimulating discussions concerning this work. This research was supported by the SDIO/IST's Ultra High Speed Computing Program administered through the U.S. Office of Naval Research in conjunction with the Optical Systems Laboratory at Texas Tech University and was also partially supported by a National Science Foundation Presidential Young Investigator Award.

References

1. A. A. Sawchuk and T. C. Strand, Proc. IEEE **72,** 758 (1984).
2. Special issue on digital optical computing, Opt. Eng. **25,** (1986).
3. C. E. Shannon, Bell. Syst. Tech. J. **27,** 379 (1948).
4. R. E. Blahut, *Theory and Practice of Error Control Codes* (Addision-Wesley, Reading, Mass., 1983).
5. Also called synthetic discriminant function and linear combination filter.
6. Q. B. Braunecker, R. Hauck, and A. W. Lohmann, Appl. Opt. **18,** 2746 (1980).
7. B. V. K. V. Kumar, Appl. Opt. **22,** 1445 (1983).
8. D. Casasent, Appl. Opt. **23,** 1620 (1984).
9. H. J. Caulfield and R. S. Putnam, Opt. Eng. **24,** 65 (1985).
10. N. H. Farhat, D. Psaltis, A. Prata, and E. Paek, Appl. Opt. **24,** 1469 (1985).
11. H. J. Larson, *Introduction to Probability Theory and Statistical Inference* (Wiley, New York, 1969).
12. R. H. Kuhn, in *VLSI and Modern Signal Processing*, S. Kung, H. Whitehouse, and T. Kailath, eds. (Prentice-Hall, Englewood Cliffs, N.J., 1985), pp. 178–184.
13. R. C. Aubusson and I. Catt, IEEE J. Solid-State Circuits **CS-15,** 677 (1980).
14. H. L. VanTrees, *Detection, Estimation and Modulation Theory, Part I* (Wiley, New York, 1968).