Complex reference discriminant functions implemented iteratively on a joint transform correlator

Joseph Rosen, Uri Mahlab, and Joseph Shamir

Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel
Received 10 August 1990.
0003-6935/91/355111-05$05.00/0.

Iterative learning procedures on hybrid electro-optic systems were employed to generate complex reference discriminant functions. The procedure was implemented using a joint transform correlator equipped with a single inexpensive spatial light modulator. Experimental results demonstrate efficient two-class discrimination.

The iterative generation of spatial filters on digital computers was pioneered in Ref. 1. This was extended in Refs. 2 and 3 to the concept of direct learning on the actual optical correlator. The basic principle of this concept is the use of an adaptively variable reference function located alongside the input pattern in a joint transform correlator (JTC) configuration.[4] The desired reference discriminant function (RDF) is obtained iteratively by measuring the correlation output distributions and adjusting the RDF values accordingly. In a conventional JTC application, however, only real, positive values of the RDF were implemented reducing the discrimination capabilities compared with complex discriminant functions.[5]

In the present work we describe the experiment of two-class learning with the electro-optical system described in Ref. 3. This system's procedure makes it possible to generate a complex-valued RDF on a regular JTC configuration. Although the RDF is displayed on a spatial light modulator (SLM) as a binary (1, 0) mask, it effectively behaves like a complex function. This function is generated by a learning procedure that is also implemented on the JTC controlled by a personal computer.

In the first stage of the experiment the correlator is presented with a set of patterns divided into two classes in any arbitrary way. Defining a recognition criterion, we instructed the system to discriminate between the two classes. During this stage of the experiment the JTC operates in its learning mode. After completing the learning mode the computer has in its memory an RDF designed for the specific task, and it can be presented in the proper way on the SLM. This is done during the second stage of the process, the operating mode, when the system performs the classification task.

The fundamental concept behind our approach is that the learning and operating modes are both implemented on the same intelligent system in its natural environment. Adopting this concept will enable in the future quick learning and updating of the RDF's stored in the system memory during a real-time recognition process. Moreover, unlike a similar learning algorithm that is implemented on digital computers alone,[6] the process here takes into account all the distortions and the constraints induced by the optical system.

The correlation process, which uses one-dimensional (1-D) signals for clarity, is illustrated graphically in Fig. 1. The procedure is started with the learning mode, an arbitrary binary reference function $r(x, y)$, $[r(x, y) \in \{0, 1\}]$, which is presented on an SLM in the input plane together with a pattern to be recognized. We may assume without loss of generality that the object and the reference are located at distances b and $-b$ from the origin along the y axis, respectively [Fig. 1(a)]. The input object function $f(x, y)$ is multiplied by a 1-D grating, $\Sigma_{m=-\infty}^{\infty}$ rect$[(y - md)/a]$, as illustrated in Fig. 1(b). The overall amplitude transmittance at the input plane is shown in Fig. 1(c) and the complex amplitude distribution at the Fourier plane in Fig. 1(d). Selecting only the first diffraction order by the television camera yields the interference pattern shown in Fig. 1(e). This intensity distribution is displayed again on the SLM. Given a unit illumination on the SLM the complex amplitude distribution over the output plane after another Fourier transformation (FT) is given by [see Fig. 1(f)]

$$s(x, y) = A[f(x, y) \oplus f(x, y) + r'(x, y) \oplus r'(x, y) + f(x, y + 2b) \oplus r'(x, y) + r'(x, y) \oplus f(x, y - 2b)], \quad (1)$$

where $\oplus$ denotes correlation. The cross correlation between $f(x, y)$ and $r'(x, y)$ is obtained around the point $(0, 2b)$. Although the original reference function $r(x, y)$ is a real binary function, our RDF $r'(x, y)$ may acquire any complex distribution. The FT of $r(x, y)$, $R(f_x, f_y)$, exists over the entire Fourier plane and satisfies the condition

$$R(f_x, f_y) = R^*(-f_x, -f_y), \quad (2)$$

as it is the FT of a real function. On the other hand $R'(u, v)$ [the FT of $r'(x, y)$] is a band-limited function centered around the point $(0, 1/d)$, and it does not necessarily satisfy a condition that is similar to Eq. (2). Thus the function $r'(x, y)$ will no longer be restricted to real values.

The minimal condition to guarantee the concentration of adequate energy from the distribution of $R(u, v)$ around the point $(0, 1/d)$ is that the pixel size in $r(x, y)$ does not exceed d. This requirement allows the origin of the reference spectrum to be shifted along the v direction to create a region of overlap between the spectrum of the object and that of the reference pattern. Within this region an interference pattern is created that yields a high correlation peak in the FT plane.

The main advantage of this system is its ability to perform a correlation between an input signal and a desired complex, multilevel, reference function without actually recording one. Although the reference mask is only binary it contains all the required complex information as a binary computer-generated hologram. In our JTC with a complex reference function we exploit the advantages of the conventional JTC architecture together with those of the 4-F correlator. To be specific the JTC is compact and does not require tedious alignment procedures, while the 4-F correla-

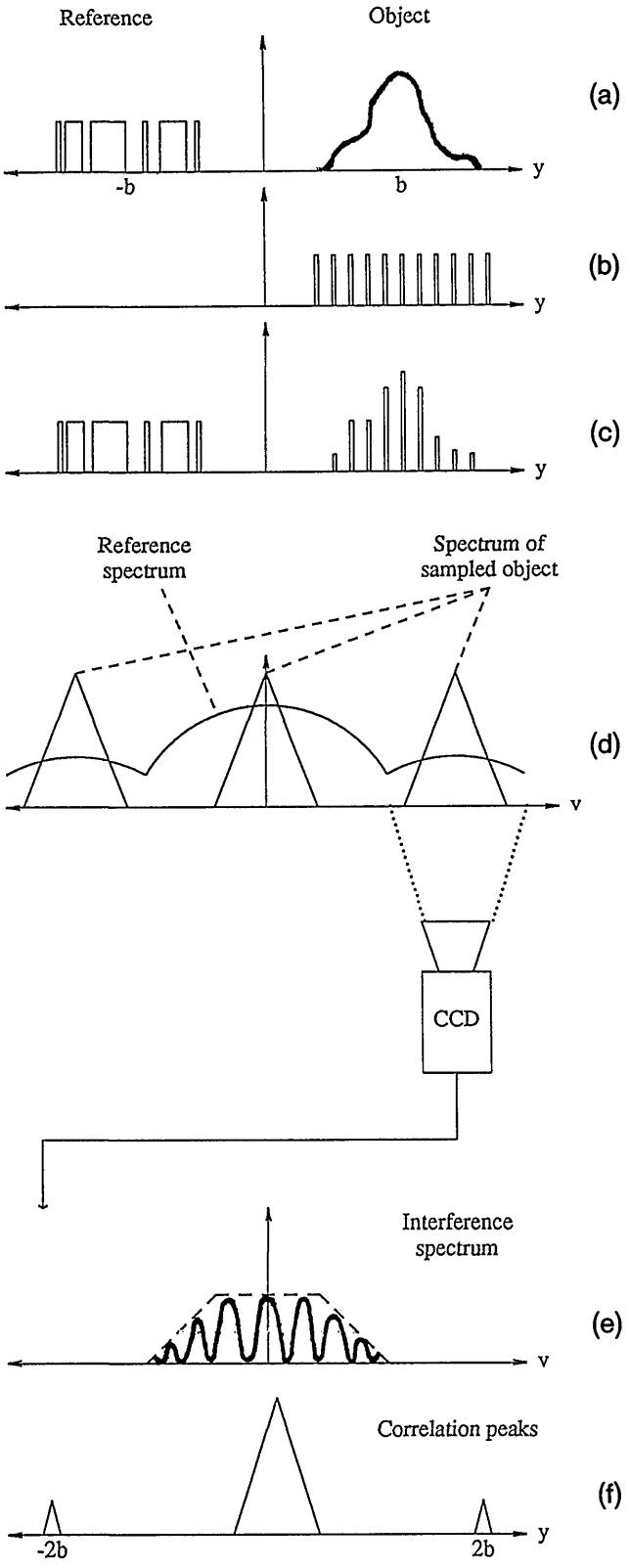

Fig. 1. Schematic illustration of the correlation process between an effectively complex reference function and an object by means of a JTC (see text for details).

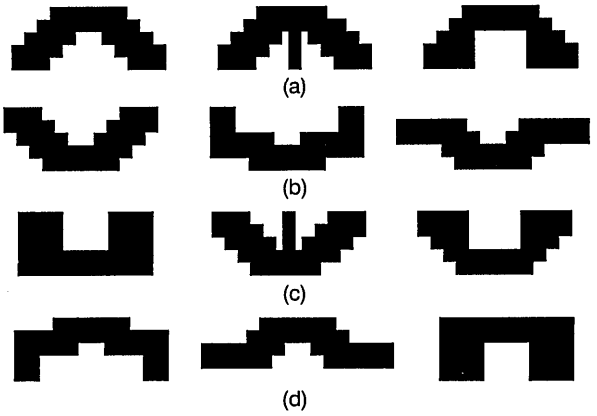

Fig. 2. Input training set: (a) patterns of class A to be detected in the first part of the experiments; (b) patterns of class B to be rejected; (c), (d) additional test objects.

tor accepts complex, multilevel, impulse response functions. Our system possesses all these attributes. However, there is a penalty for these benefits in terms of the reduced space–bandwidth product. Compared with the conventional JTC our system has only one third of its bandwidth along one of the coordinate axes. The reason is that we have to introduce three diffraction orders of $F(u, v)$ within the bandwidth provided by the optical aperture.

The algorithm for generating the RDF is based on a minimization of a cost function, which takes into account the discrimination demands over the complete correlation plane. The cost function is defined by using the elements of the RDF as variables. Assuming a set of patterns $f_n(x, y)$, we define the goal of the system as the detection of the presence of patterns out of the subset $f_n^p(x, y)$ while rejecting all other patterns denoted by the subset $f_n^q(x, y)$. A reasonable criterion for detection is the appearance of a strong peak as contrasted with a uniform distribution for the patterns to be rejected. Therefore, we define a cost

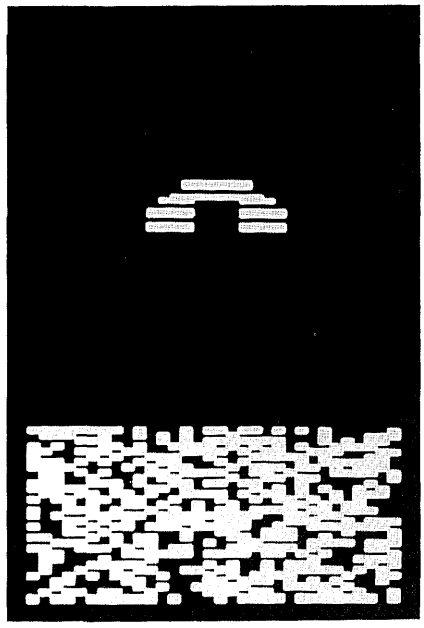

Fig. 3. Input plane distribution. The sampled object is shown above, and the binary reference is shown below.

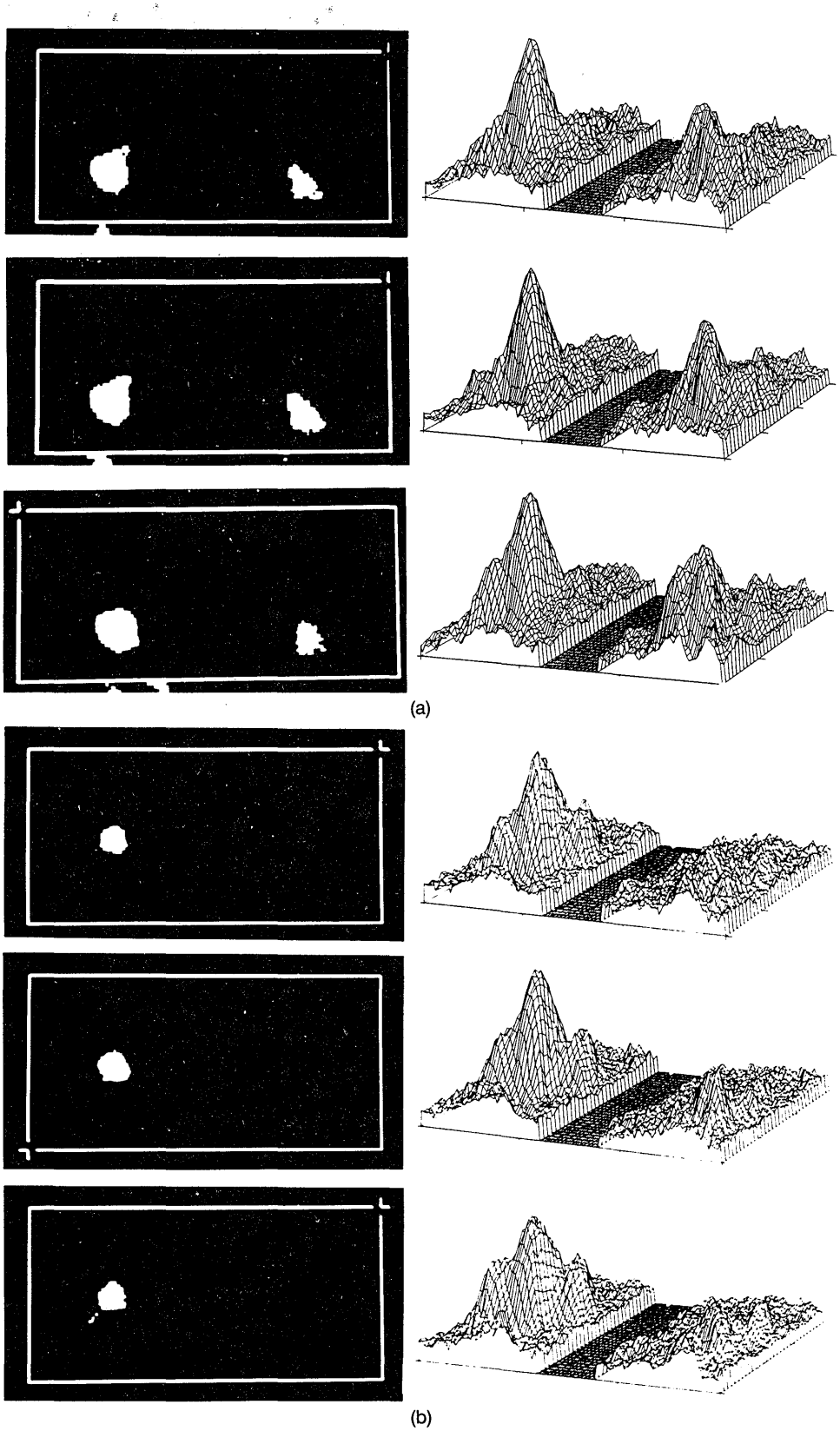

Fig. 4. Two steps toward the convergence of the process (see text for details).

function M by the relation

$$M(\mathbf{r}) = \frac{\max_{n\in q}\left\{\frac{1}{\Delta^2}\int_{\underline{X}_n}^{\overline{X}_n}\int_{\underline{Y}_n}^{\overline{Y}_n}|f_n \oplus r'|^2\,\mathrm{d}x\mathrm{d}y\right\}}{\min_{n\in p}\left\{\frac{1}{\Delta^2}\int_{\underline{X}_n}^{\overline{X}_n}\int_{\underline{Y}_n}^{\overline{Y}_n}|f_n \oplus r'|^2\,\mathrm{d}x\mathrm{d}y\right\}}, \tag{3}$$

with

$$\underline{X}_n = x_o^n - \frac{\Delta}{2},\quad \overline{X}_n = x_o^n + \frac{\Delta}{2},\quad \underline{Y}_n = y_o^n - \frac{\Delta}{2},\quad \overline{Y}_n = y_o^n + \frac{\Delta}{2}.$$

Here (x_o^n, y_o^n) is the point where $(\max_{x,y}|f_n \oplus r'|^2)$ is achieved for the nth pattern. The small integration area Δ^2 is chosen to be the width of the spot where the intensity drops to half of its maximum value. The meaning of the cost function is the ratio between the maximum among all the correlation peaks belonging to the rejected class ($n \in q$) and the minimum of all the correlation peaks belonging to the detected class ($n \in p$) for a given reference $\mathbf{r}$. For an efficient class discriminator the cost function of Eq. (3) must be minimized. The whole learning and recognition process introduced here was implemented on the actual optical correlator; all the constraints and distortions of the system were taken into account. The iterative minimization process employs the direct binary search algorithm.[2,3,7,8] This iterative process is continued until there is no significant reduction for a long time in the low state of $M(\mathbf{r})$.

Unlike similar procedures employed by digital computers,[7] the real-life measurements of the intensity distributions at each iteration are noisy. A breakdown of the iterative procedure may occur when a random noise fluctuation causes the value of the calculated cost function to jump to an unreasonably small value that cannot be approached by subsequent iterations. To avoid such locking into an unacceptable value of the cost function, one should modify the algorithm. For each iteration we calculate the mean $E_i(M)$ and standard deviation σ_i of the measurements of the last k iterations (k is large enough to satisfy statistical conditions). Denoting by M_i the last minimum value of $M(\mathbf{r})$, we define the decision criterion at each iteration by three distinct situations:

$$\text{if } M(\mathbf{r}_{i+1}) \geq \underline{M}_i \text{ then } \underline{M}_{i+1} = \underline{M}_i,$$

$$\text{if } M(\mathbf{r}_{i+1}) < \underline{M}_i \text{ and } |M(\mathbf{r}_{i+1}) - E_i(M)| \leq \sigma_i \text{ then } \underline{M}_{i+1} = M(\mathbf{r}_{i+1}),$$

$$\text{if } M(\mathbf{r}_{i+1}) < \underline{M}_i \text{ and } |M(\mathbf{r}_{i+1}) - E_i(M)| > \sigma_i$$

$$\text{then } \underline{M}_{i+1} = \min[\underline{M}_i, [E_i(M) - \sigma_i]]. \tag{4}$$

In other words we do not change the actual minimum value of the cost function unless it is within reasonable statistical limits.

For our demonstration experiments we used a single SLM to implement the JTC.[3] The SLM is an inexpensive liquid crystal television set with 162 × 144 pixels. The operation of the system was controlled by a CUE-2 image processor.[9] We chose as the training set three objects of class A and three of class B, as shown in Figs. 2(a) and 2(b), respectively. Six additional objects [Figs. 2(c) and (d)], which could be associated with the two classes but did not participate in the training set, were employed to test the system performance for more general class discrimination. Our primary goal was to detect the patterns shown in Fig. 2(a) and reject those shown in Fig. 2(b). Figure 3 is the JTC input plane where the input pattern is positioned on the upper part and the binary reference function is placed on the lower part. The size of the reference function in this experiment was 64 × 32 pixels. The minimum size of the smallest element of the reference is one pixel in the y direction and two pixels in the x direction. Thus we effectively have the binary reference function represented on a matrix of 32 × 32 elements. The grating sampling the input object consists of two pixels for d. The same procedure is repeated for the other objects. The measurements of the six correlation peaks, three for each class, are used to calculate the cost function [Eq. (3)], and decisions are made according to Eq. (4). After the decision is made the new reference function is displayed and a new cycle starts. The iterations are repeated until a satisfactory discrimination is achieved. Based on a set of 30 independent experiments the search process converged on the average after 20,000 accesses to the RDF mask. The convergence rate depends on the RDF size and may be improved by changing the search algorithms, which is a subject for future research.

Figure 4 depicts two stages along the iterative learning process. Each figure contains the correlation regions of the six objects along with a 3-D display of the intensity distribution. The images over the left-hand side in each frame belong to the three objects of Fig. 2(a), while those on the right-hand side originate from the patterns of Fig. 2(b). The discrimination between the different classes is improved from Fig. 4(a) to Fig. 4(b), which shows the final results of the learning stage when the process is terminated. The system achieved a minimum cost function value of 0.67; i.e., the ratio between the strongest correlation peak of the undesired class to the weakest correlation peak of the desired class is 1:1.5. These results can be improved if we

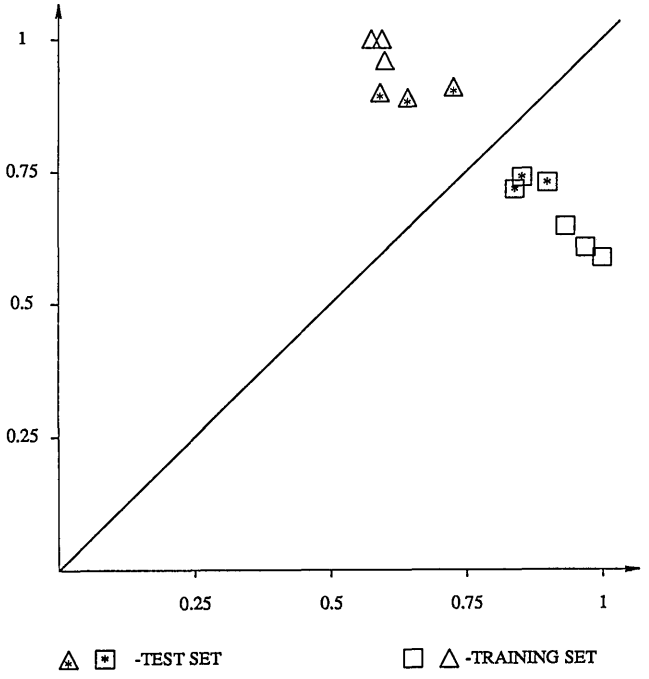

Fig. 5. Classification results with the complex reference of the two classes shown in Fig. 2. The vertical axis is the normalized correlation strength with the filter for class A, and the horizontal axis is the filter prepared to detect the patterns of class B. The triangles represent the patterns in Fig. 2(a), and the patterns in Fig. 2(d) are represented by triangles with asterisks. The open rectangles represent the patterns in Fig. 2(b), and the rectangles with asterisks denote the patterns of Fig. 2(c).

subtract from the output intensity distribution the dark signal level of the camera. This level value is ~30% of the maximum intensity. Another technical limitation is the low contrast of the liquid-crystal television SLM, which suppresses the intensity spectrum at high spatial frequencies.

To expose the system to additional tests the procedure was repeated with the two classes interchanged. An RDF was generated to recognize the objects of class B shown in Fig. 2(b) and reject those of class A [Fig. 2(a)]. The performances of the two RDF's were tested with all six objects of the training set and also the six objects of the test set [Figs. 2(c) and 2(d)]. The measured values of the correlation peaks for all twelve patterns are plotted in Fig. 5. The axes of this plot indicate the correlation peak values of the two RDF's. The diagonal line denotes all the points of equal peak values in both filters—the region of nondiscrimination. The farther the clusters are from the diagonal the better is the discrimination. The patterns of Fig. 2(a) are denoted by open triangles, and the triangles with the asterisks represent the test set of Fig. 2(d). The patterns of Fig. 2(b) are denoted by open rectangles, and the rectangles with the asterisks represent the test set of Fig. 2(c). As indicated by Fig. 5, we have successfully demonstrated the generalization and classification capability of the complex reference. There was excellent discrimination among the two classes participating in the training while other patterns were rejected by both filters, although with reduced discrimination.

The high discrimination capability of these reference patterns was compared by laboratory experiments to the performances of a conventional JTC that yielded a discrimination ratio of only 1:1.05. This emphasizes the need for an unconventional procedure. The hybrid learning architecture is promising because the FT's are performed optically, and they take into account the actual optical system parameters. Thus aberrations and distortions are automatically handled. The optical system is compact and easy to align and uses a small number of inexpensive components.

This work was performed within the Technion Advanced Opto-Electronics Center established by the American Technion Society, N.Y.

References

1. R. D. Juday and B. J. Daiuto, "Relaxation method of compensation in an optical correlator," Opt. Eng. **26,** 1094–1101 (1987).
2. J. Rosen, U. Mahlab, and J. Shamir, "Adaptive learning with joint transform correlators," Opt. Eng. **29,** 1101–1106 (1990).
3. U. Mahlab, J. Rosen, and J. Shamir, "Iterative generation of complex reference functions in a joint-transform correlator," Opt. Lett. **16,** 330–332 (1991).
4. C. S. Weaver and J. W. Goodman, "A technique for optically convolving two functions," Appl. Opt. **5,** 1248–1249 (1966).
5. G. Gheen, "Design considerations for low-clutter, distortion invariant correlation filters," Opt. Eng. **29,** 1029–1032 (1990).
6. U. Mahlab and J. Shamir, "Phase-only entropy optimized filter generated by simulated annealing," Opt. Lett. **14,** 146–148, (1989).
7. M. A. Seldowitz, J. P. Allebach, and D. W. Sweeney, "Synthesis of digital holograms by direct binary search," Appl. Opt. **26,** 2788–2798 (1987).
8. U. Mahlab, J. Rosen, and J. Shamir, "Iterative generation of holograms on spatial light modulator," Opt. Lett. **15,** 556–558 (1990).
9. The CUE-2 is an image processing system based on a personal computer manufactured by Galai Laboratories, Industrial Zone, Migdal Haemek, Israel.

Improved arithmetic Fourier transform algorithm

Palacharla Paparao and Anjan Ghosh

The authors are with the Department of Electrical and Computer Engineering, University of Iowa, Iowa City, Iowa 52242.

Received 3 August 1990.
0003-6935/91/355115-04$05.00/0.

A new, improved version of the arithmetic Fourier transform algorithm is presented. This algorithm computes the Fourier coefficients of continuous-time signals by using the number-theoretic technique of Mobius inversion. The improved algorithm can calculate all the Fourier coefficients including the dc component. It also requires a smaller number of delays and arithmetic operations than the standard arithmetic Fourier transform algorithm.

Recently Tufts and Sadasiv[1] proposed a method for computing the Fourier coefficients of zero-dc continuous-time signals by using the number-theoretic technique of Mobius inversion. They called this method of Fourier analysis the arithmetic Fourier transform (AFT).[1–3] We use this algorithm to calculate Fourier coefficients from approximately $2N^2/3$ (N is the number of harmonics in the signal) nonuniformly spaced samples of the signal.[1–3] The AFT can be used for accurate high-speed Fourier analysis and synthesis. The major advantage of the AFT algorithm is that it needs mostly addition operations, except for a few multiplications by real-scale factors. The AFT computations are not complex and are designed for real-time spectral analysis of signals. The AFT can be easily realized on general-purpose, signal-processing, very large scale integrated (VLSI) chips. It is suitable for parallel processors. This algorithm has been applied to the computation of a discrete cosine transform[4] and also extended to two-dimensional signals for image processing.[5] In a previous paper[6] we showed that the AFT algorithm can be realized efficiently on an incoherent optical parallel processor by using nonuniformly tapped fiber-optic delay lines. An advantage of the optical AFT processor compared to the standard optical discrete Fourier transform calculator[7] is that we avoid complex number manipulations and electronic postprocessing for computing the basic Fourier coefficients.

In this paper, we present two improvements to the AFT algorithm such that it can be realized more efficiently. The modified AFT (MAFT) algorithm can calculate all the Fourier coefficients including the dc component. This additional feature of dc calculation will allow an incoherent optical processor to determine the Fourier coefficients of nonzero-mean light-wave signals accurately. The MAFT algorithm also requires a smaller number of taps and arithmetic operations to calculate the Fourier coefficients than the standard AFT proposed in Ref. 2. The spectral resolution of an MAFT-based system is dependent on the spacings between the taps and the number of taps on the delay line. The basic structure of the MAFT is the same as

the standard AFT and thus, the MAFT can also be implemented easily on VLSI chips and parallel processors.

The AFT can be used to calculate the Fourier coefficients of real-value band-limited periodic signals of the form

$$x(t) = \sum_{k=1}^{N} a_k \cos\left(\frac{2\pi kt}{T}\right) + \sum_{k=1}^{N} b_k \sin\left(\frac{2\pi kt}{T}\right), \quad (1)$$

where a_k and b_k are the coefficients of the kth harmonic with a frequency of $2\pi k/T$ rad/s. We assume that $x(t)$ is stationary, contains no zero-frequency (dc) term, and each harmonic above the Nth is zero. If we shift the periodic signal $x(t)$ in Eq. (1) in time by an amount αT where $-1 < \alpha < 1$, we get

$$x(t + \alpha T) = \sum_{k=1}^{N} c_k(\alpha) \cos\left(\frac{2\pi kt}{T}\right) + \sum_{k=1}^{N} d_k(\alpha) \sin\left(\frac{2\pi kt}{T}\right), \quad (2)$$

where

$$c_k(\alpha) = a_k \cos(2\pi k\alpha) + b_k \sin(2\pi k\alpha), \quad (3)$$

$$d_k(\alpha) = -a_k \sin(2\pi k\alpha) + b_k \cos(2\pi k\alpha). \quad (4)$$

For the AFT we need a new set of averages of the signal $x(t + \alpha T)$ sampled at n equally spaced points, defined as

$$s(n, \alpha) = \frac{1}{n} \sum_{l=0}^{n-1} x(lT/n + \alpha T). \quad (5)$$

The coefficients $c_k(\alpha)$ in Eq. (3) can be evaluated from the sum $s(n, \alpha)$ by using the Mobius inversion formula for a finite series as follows:

$$c_k(\alpha) = \sum_{m=1}^{[N/k]} \mu(m) s(mk, \alpha) \quad \text{for } k = 1, 2, \ldots, N, \quad (6)$$

where $\mu(m)$ represents a Mobius function.[8] The Mobius function can only take values from the set $\{-1, 0, 1\}$. The basic theory of the AFT algorithm is given in the following theorems.[2]

Theorem 1A

Let $\alpha = 0$ and let k be an integer such that $1 \le k \le N$. Then

$$a_k = c_k(0) \quad \text{for } k = 1, 2, \ldots, N. \quad (7)$$

Theorem 1B

First, let n be an integer such that $0 \le n \le [\log_2(N)]$, where $[z]$ denotes the principal part of z. Next let $\alpha = 1/2^{n+2}$ and assume k is an integer of the form $k = 2^n(2m + 1)$ in the intersection of set $N_n = \{2^n(2m + 1) \mid m$ belongs to the set of all integers including zero$\}$ and the finite set $\mathbf{I}_{(1,N)} = \{1, 2, \ldots, N\}$. Then

$$b_k = (-1)^m c_k(1/2^{n+2}). \quad (8)$$

Equations (6)–(8) provide a number-theoretic method for calculating the kth Fourier coefficients a_k and b_k from $c_k(\alpha)$, and the computations can be performed in parallel. Thus in the AFT each Fourier coefficient is calculated as a linear combination of the signal $x(t)$ and its delayed samples at a particular time. Expressing the terms $s(n, \alpha)$ as sums of delayed samples of $x(t)$ we have shown that the vector of Fourier coefficients, $\mathbf{f} = [a_1 \ldots a_k b_1 \ldots b_k]$, can be obtained from

$$\mathbf{f} = \mathscr{W}\mathbf{X}, \quad (9)$$

where $\mathbf{X}$ is a vector of nonuniform samples of $x(t)$ and $\mathscr{W}$ is the AFT coefficient matrix.[6]

Numerical analysts suggested a similar way of calculating the Fourier series of a given function. They calculated the Fourier coefficients through the numerical approximation of Fourier integrals. One such method for the computation of the Fourier coefficients based on the Mobius inversion of the Poisson summation formula (called the MIPS method in numerical analysis literature) was proposed by Goldberg and Varga[9] and elaborated by Lyness.[10,11] This method requires a greater number of taps or function samples compared with the AFT algorithm. This method is particularly suitable in a situation in which all the Fourier coefficients are required to have uniform accuracy.[10,11]

From Theorem 1A we observe that a single choice of $\alpha = 0$ in Eq. (7) supplies every a_k, $k = 1, 2, \ldots, N$ for all N, while, on the other hand, there is no fixed α (or fixed finite set of α values) supplying b_k, $k = 1, 2, \ldots, N$ for all N. We require α to take $[\log_2 N] + 1$ different values to obtain all the b_k coefficients. In the modified version of the AFT algorithm, we select α to take one more value other than $\alpha = 0$, namely, $\alpha = 1/4N$.

Theorem 2

The b_k coefficients can be represented as a linear combination of the signal samples $x(lT/n)$ and $x(lT/n + T/4N)$.

Proof

From Eq. (6) and $\alpha = 1/4N$, we derive

$$b_k = \frac{1}{\sin(\pi k/2N)} [c_k(1/4N) - a_k \cos(\pi k/2N)]$$

$$= \frac{1}{\sin(\pi k/2N)} [c_k(1/4N) - c_k(0) \cos(\pi k/2N)]$$

$$\text{for } k = 1, 2, \ldots, N. \quad (10)$$

When we combine Eqs. (5), (6), and (10), we notice that b_k can be written as a linear combination of the signal samples $x(lT/n)$ and $x(lT/n + T/4N)$. The theorem is proved.

Our MAFT algorithm comprises Theorems 1A[2] and 2, which replaces Theorem 1B. The MAFT algorithm can still be formulated as a single matrix–vector multiplication of Eq. (9) with a different $\mathbf{W}$ and $\mathbf{X}$. The main advantage of the new algorithm is that it requires a smaller number of taps compared to the standard AFT algorithm. This is mainly due to the selection of a single value of α as opposed to $[\log_2 N] + 1$ values. When we implement the MAFT algorithm with one nonuniformly tapped delay line, the required number of signal samples or taps M_{tap} is twice the number of distinct samples $x(lT/n)$ where $l < n$ and l is relatively prime to n. The number of taps is given approximately by the relation $M_{tap} \approx 2N^2/3 + O(NlnN)$. Table I shows the number of taps required for the cases $N = 4$ and $N = 8$. Notice that the difference between the number of taps required for the standard AFT and the MAFT algorithms increases rapidly as the number of harmonics N increases.

In all the previous formulations of the AFT algorithm,

Table I. Total Number of Signal Samples (Taps) Required for Implementing the AFT and Improved AFT Algorithms

	Number of Taps Required M_{tap}	
N	AFT	Improved AFT
4	16	12 (+2)
8	56	44 (+4)

The number of extra taps required for dc calculation is indicated in parentheses.

the signal $x(t)$ was assumed to be zero mean, i.e., no dc component. In general, any typical band-limited signal can be represented as a finite series, i.e.,

$$x(t) = a_0 + \sum_{k=1}^{N} [a_k \cos(2\pi kt/T) + b_k \sin(2\pi kt/T)],$$

where a_0 is the dc term (the mean value of the signal). Also, in the incoherent optical systems, the signal $x(t)$ in the optical fiber is always positive, having a dc component along with the harmonics. For signals with a dc component, the AFT algorithm described in Eq. (9) would produce a vector,

$$\mathbf{f}_0 = \mathbf{f} + \mathbf{M}_w a_0, \tag{11}$$

where the ith element of the $2N$-vector $\mathbf{M}_w$ is Σw_i, the ith sum of matrix $\mathbf{W}$.[6] So we need to modify the AFT algorithm to handle nonzero-mean signals. The first step in the modification will be to calculate the dc value. The following theorem gives a method for calculating the dc exactly from the delayed samples of $x(t)$. This method is superior to that of estimating the dc from a moving average of $x(t)$ as described in our previous paper, Ref. 6.

Theorem 3

Let the finite Fourier series of a signal $x(t)$ be of the form

$$x(t) = a_0 + \sum_{k=1}^{N} [a_k \cos(2\pi kt/T) + b_k \sin(2\pi kt/T)]. \tag{12}$$

Then the mean (dc) of the signal is obtained as follows:

$$a_0 = \frac{1}{N+m} \sum_{i=0}^{N+m-1} x\left(\frac{iT}{N+m}\right), \tag{13}$$

for any integer $m \geq 1$.

Proof

We use Eq. (12) and get

$$\frac{1}{N+m} \sum_{i=0}^{N+m-1} x\left(\frac{iT}{N+m}\right)$$

$$= a_0 + \frac{1}{N+m} \sum_{i=0}^{N+m-1} \sum_{k=1}^{N} \left[a_k \cos\left(\frac{2\pi ik}{N+m}\right) + b_k \sin\left(\frac{2\pi ik}{N+m}\right)\right]$$

$$= a_0 + \frac{1}{N+m} \sum_{k=1}^{N} \left[a_k \sum_{i=0}^{N+m-1} \cos\left(\frac{2\pi ik}{N+m}\right) + b_k \sum_{i=0}^{N+m-1} \sin\left(\frac{2\pi ik}{N+m}\right)\right]. \tag{14}$$

But

$$\frac{1}{N+m} \sum_{i=0}^{N+m-1} \exp\left(\frac{j2\pi ik}{N+m}\right)$$

$$= \begin{cases} 1, & \text{if } (N+m) | k \text{ or } k = l(N+m) \text{ for some integer } l \\ 0, & \text{otherwise} \end{cases}, \tag{15}$$

where $j = \sqrt{-1}$ and $(N+m)|k$ denotes "integer $(N+m)$ divides k." Hence, by taking the real and imaginary parts,

$$\frac{1}{N+m} \sum_{i=0}^{N+m-1} \cos\left(\frac{2\pi ik}{N+m}\right) = \begin{cases} 1, & \text{if } k = l(N+m) \\ 0, & \text{otherwise} \end{cases}, \tag{16}$$

$$\frac{1}{N+m} \sum_{i=0}^{N+m-1} \sin\left(\frac{2\pi ik}{N+m}\right) = 0 \quad \text{for all integers } k \text{ and } (N+m). \tag{17}$$

Since k varies from 1 to N, the integer $(N+m)$ does not divide k for all $m \geq 1$. Thus, from Eq. (14),

$$\frac{1}{N+m} \sum_{i=0}^{N+m-1} x\left(\frac{iT}{N+m}\right) = a_0,$$

and the theorem is proved.

The sum in Eq. (13) is equal to $s(N+m, 0)$ according to the definition of $s(n, \alpha)$, Eq. (5). From Eq. (13) it is clear that the dc can be calculated as a linear combination of the signal samples $x[iT/(N+m)]$. To suppress the additional component that is due to the dc term in Eq. (11), we perform the following steps: (1) Add extra taps to the delay lines to sample the signal also at $t = iT/(N+m)$, $i = 0, \ldots, N+m-1$. The value of m in Eq. (13) is selected to keep the number of extra taps required for the dc calculation at a minimum. (2) Calculate the dc value a_0, using Eq. (13). (3) Use a_0 to cancel the extra term $\mathbf{M}_w a_0$ from the observed vector $\mathbf{f}_0$ and determine the Fourier coefficients $\mathbf{f}$.

All these steps can be integrated together in one parallel multiplication of a vector $\mathbf{X}_{new}$ by a new matrix $\mathbf{W}_{new}$. The vector $\mathbf{X}_{new}$ consists of the vector $\mathbf{X}$ and a few extra samples for dc calculation. The matrix $\mathbf{W}_{new}$ is a modified version of the interconnection matrix $\mathbf{W}$. We model $\mathbf{W}_{new}$ as the product of an augmented matrix $[\mathbf{W}: -\mathbf{M}_w]$ and a matrix $\mathbf{B}$. The matrix $\mathbf{B}$ is designed such that

$$\mathbf{B}\mathbf{X}_{new} = [\mathbf{X}: a_0]^T.$$

Notice that the product

$$\mathbf{W}_{new}\mathbf{X}_{new} = \mathbf{f}_0 - \mathbf{M}_w a_0 = \mathbf{f}$$

is the desired vector of Fourier coefficients.[6] Thus the overall strategy for calculating Fourier coefficients using MAFT's can be realized by a single matrix–vector multiplication:

$$\mathbf{f} = \mathbf{W}_{new}\mathbf{X}_{new}, \tag{18}$$

on a parallel processor such as the optical system described in Ref. 6.

The following example illustrates how a MAFT algorithm consisting of the results of Theorems 1A, 2, and 3 can be used to compute all the Fourier coefficients of a nonzero-

mean signal bandlimited to four harmonics [$N = 4$ in Eq. (12)].

Step 1: To calculate the dc component a_0, we use Eq. (13). The value of m is selected to keep the number of extra taps required for dc calculation to a minimum. For example, when $m = 1$ we need 4 extra taps at $T/5$, $2T/5$, $3T/5$, and $4T/5$. The choice of $m = 2$ requires only two extra taps at $T/6$ and $5T/6$. Thus the dc component is given by

$$a_0 = \tfrac{1}{6}[x(0) + x(T/6) + x(T/3) + x(T/2) + x(2T/3) + x(5T/6)]. \quad (19)$$

Step 2: Before calculating the a_k and b_k coefficients, we define a new zero-mean signal:

$$\bar{x}(t) = x(t) - a_0. \quad (20)$$

We note that the values of the Mobius functions are $\mu(1) = 1$, $\mu(2) = -1$, $\mu(3) = -1$, and $\mu(4) = 0$.

Step 3: We form the sums $s(n, 0)$ at the output of the bank of delay lines according to Eq. (5).

Step 4: From Eq. (7) in Theorem 1A, the a_k Fourier coefficients are determined as follows:

$$a_k = c_k(0) = \sum_{l=1}^{[4/k]} \mu(l) s(lk, 0) \quad \text{for } k = 1, 2, 3, 4. \quad (21)$$

Step 5: To compute b_k for $1 \le k \le 4$, let $\alpha = 1/4N = 1/16$. From Eq. (10) in Theorem 2, we obtain

$$b_k = \frac{1}{\sin(\pi k/8)} [c_k(1/16) - c_k(0) \cos(\pi k/8)]$$

$$= \frac{1}{\sin(\pi k/8)} \sum_{l=1}^{[4/k]} \mu(l) [s(lk, 1/16) - s(lk, 0) \cos(\pi k/8)]$$

$$\text{for } k = 1, 2, 3, 4. \quad (22)$$

The a_k and b_k coefficients determined above can be expressed as a linear combination of the samples of signal $x(t)$ by combining Eqs. (20)–(22). These linear equations are formulated in terms of a matrix–vector multiplication ($\mathbf{W}_{new}\mathbf{X}_{new}$). Steps 1–5 can be easily programmed into a computer to generate the matrix $\mathbf{W}_{new}$ and the vector $\mathbf{X}_{new}$ for any value of N.

This research is supported in part by National Science Foundation grant EET-8707863. Encouragement from S. M. Reddy is gratefully acknowledged.

References

1. D. W. Tufts and G. Sadasiv, "The arithmetic Fourier transform," IEEE ASSP Mag. **5**(1), 13–17 (1988).
2. I. S. Reed, D. W. Tufts, X. Yu, T. K. Truong, M. Shih, and X. Yin, "Fourier analysis and signal processing by use of the Mobius inversion formula," IEEE Trans. Acoust. Speech Signal Process. **38,** 458–470 (1990).
3. N. Tepedelenlioglu, "A note on computational complexity of the Arithmetic Fourier Transform," IEEE Trans. Acoust. Speech Signal Process. **37,** 1146–1147 (1989).
4. D. W. Tufts, Z. Fan, and Z. Cao, "Image processing and the arithmetic Fourier transform," in *High Speed Computing II,* K. Bromley, ed., Proc. Soc. Photo-Opt. Instrum. Eng. **1058,** 46–53 (1989).
5. I. S. Reed, Y. Y. Choi, and X. Yu, "Practical algorithm for computing the 2-D arithmetic Fourier transform," in *High Speed Computing II,* K. Bromley, ed., Proc. Soc. Photo-Opt. Instrum. Eng. **1058,** 54–61 (1989).
6. A. Ghosh, S. D. Allen, and P. Paparao, "Arithmetic Fourier-transform calculation using fiber-optical parallel processors," J. Opt. Soc. Am. A **7,** 701–707 (1990).
7. J. W. Goodman, A. R. Dias, and L. M. Woody, "Fully parallel, high speed incoherent optical method for performing discrete Fourier transform," Opt. Lett. **2,** 1–3 (1978).
8. M. R. Schroeder, *Number Theory in Science and Communication* (Springer-Verlag, Berlin, 1986).
9. R. R. Goldberg and R. S. Varga, "Mobius inversion of Fourier transforms," Duke Math. **23,** 553–559 (1956).
10. J. N. Lyness, "The calculation of Fourier coefficients," SIAM J. Numer. Anal. **4,** 301–315 (1967).
11. J. N. Lyness, "The calculation of Fourier coefficients by the Mobius inversion of the Poisson summation formula, part I: functions whose early derivatives are continuous," Math. Comp. **24,** 101–135 (1970).
12. P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration* (Academic, New York, 1975).
13. P. J. Davis, "On the numerical integration of periodic analytic functions," in *On Numerical Approximation,* R. E. Langer, ed. (U. Wisconsin Press, Madison, Wisc., 1959).