of the deconvolution algorithm, the input signal $x(n)$ is computed with no error.

V. Discussion

The new computational algorithm works for any kind of deconvolution kernel—nonsingular or singular. It is much faster than the time-domain method and its complexity can be compared to that of the FFT. In contrast to the direct DFT deconvolution, this method uses only multiplications and no division except for the final scaling.

We also tested the described method in a noisy environment. The noise behavior of the algorithm was tested by adding zero-mean white noise with preselected signal-to-noise ratios (SNR's) to the output signal sequences and to the kernels, respectively. The experiments were carried out on real signals: we took them from the American Heart Association Database of standard electrocardiograms. Kernels $h(n)$ were obtained as the averages of 500 ECG periods and truncated to length $N/2$. Randomly chosen extrasystoles from the same ECG were considered the output signals $y(n)$ of length $N(N = 256)$. To achieve statistical significance, 15 different ECG's, two leads from each, were tested and each measurement was averaged across 30 runs.

We investigated input SNR's (denoted by SNR_x) versus output SNR's (SNR_y) and versus kernels' SNR's (SNR_h) as explained by Fig. 6. Input SNR's are presented with their average (circles) and the double standard deviation ($\pm\sigma$). The solid lines correspond to deconvolution without division (the described method), the dashed lines to the direct deconvolution with interpolation. We made the kernels singular synthetically by replacing with zeros all the frequency samples $H(k)$ whose magnitudes were lower than 1.5 times the smallest magnitude value.

As it was shown in [9], deconvolution with interpolation performs 3 dB better on average than the direct DFT deconvolution. According to noise study here, the achieved SNR's are practically the same as for deconvolution with interpolated samples when the output is noisy (Fig. 6(a)), while the results are 0.7 dB better in favor of the presented method when noise is added to the deconvolution kernels (Fig. 6(b)).

Acknowledgment

The author would like to thank Professor G. S. Moschytz for his valuable discussions and many helpful suggestions during the author's six months leave at the Institute for Signal and Information Processing, ETH Zurich.

References

[1] N. S. Nahman and M. E. Guillaume, *Deconvolutions of Time Domain Waveforms in the Presence of Noise.* Washington: US Government Printing Office, National Bureau of Standards, 1981.

[2] T. K. Sarkar, F. I. Tseng, S. M. Rao, S. A. Dianat, and B. Z. Hollmann, "Deconvolution of impulse response from time-limited input and output: Theory and experiment," *IEEE Trans. Instrum. Meas.*, vol. IM-34, pp. 541–546, Dec. 1985.

[3] J. M. Mendel, *Maximum-Likelihood Deconvolution.* New York: Springer-Verlag, 1990.

[4] P. A. Jansson, *Deconvolution With Applications in Spectroscopy.* Orlando, FL: Academic, 1984.

[5] R. Pan and C. L. Nikias, "The complex cepstrum of higher order cumulants and nonminimum phase system identification," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 186–205, Feb. 1988.

[6] A. Ziolkowski, *Deconvolution.* Dordrecht, Holland: Reidel, 1984.

[7] M. Bernabini, P. Carrion, G. Jacovitti, F. Rocca, S. Treitel, and M. Worthington (Eds.), *Deconvolution and Inversion*, Proceedings of a workshop sponsored by the European Association of Exploration Geophysicists, the Society of Exploration Geophysicist, the National Research Council of Italy and the National Science Foundation in the United States, 1986. Oxford: Blackwell Scientific Publications, 1987.

[8] W. Kwong Yeung and F.-Nian Kong, "Time domain deconvolution when the kernel has no spectral inverse," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 912–918, Aug. 1986.

[9] D. Zazula and L. Gyergyek, "Direct frequency-domain deconvolution when the signals have no spectral inverse," *IEEE Trans. Signal Processing*, vol. 41, pp. 977–981, Feb. 1993.

[10] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

The Two-Dimensional Complex LMS Algorithm Applied to the 2-D DFT

Bo Liu and Leonard T. Bruton

Abstract— **The one-dimensional (1-D) LMS spectrum analyzer, as proposed by Widrow, is extended to the two-dimensional (2-D) case. The relationship between the 2-D discrete Fourier transform (DFT) and the 2-D least mean square (LMS) algorithm is established and employed to recursively compute a 2-D LMS spectrum, or the 2-D DFT, by proper choice of the LMS adaptation step μ.**

I. Introduction

Widrow *et al.*, have shown [1] that the complex LMS adaptive algorithm may be employed to recursively compute the one-dimensional (1-D) discrete Fourier transform (DFT) of a real or complex sequence d_i and to obtain a 1-D spectral estimate of d_i. This method employs the conventional 1-D adaptive linear combiner, having the adaptive weight column vector $\boldsymbol{W}_i \equiv [w_{0i}, w_{1i}, \cdots, w_{(N-1)i}]^T$ and the input column vector $\boldsymbol{X}_i \equiv [x_{0i}, x_{1i}, \cdots, x_{(N-1)i}]^T$, to generate the adaptive signal estimate $y_i \equiv \boldsymbol{W}_i^T \boldsymbol{X}_i$. The essential feature of Widrow's method is that, for the computation of the DFT of a signal d_i, of block length N, the input column vector $\boldsymbol{X}_i$ is the set of N periodic discrete phasor sequences given by $\boldsymbol{X}_i \equiv (1/\sqrt{N})[1, e^{j(2\pi/N)i}, \cdots, e^{j(2\pi(N-1)/N)i}]^T$ with $j \equiv \sqrt{-1}$, and the mean squared value of the complex error $\varepsilon_i \equiv d_i - y_i$ is minimized according to the complex LMS algorithm. Widrow *et al.* showed that the algorithm operates as a spectrum analyzer and the output vector of this spectrum analyzer is then a weighted spectral estimate of the sequence d_i. The weighting is controlled by the LMS adaptation (or convergence) step μ and, for $\mu \equiv 1/2$, this output vector is *exactly* the DFT of d_i over the block of length N defined by the index range $i - N$ to $i - 1$.

In Section II of this paper, the 2-D complex LMS algorithm is developed and applied to a sequence of 2-D images. In Section III, the 2-D LMS spectrum analyzer is proposed for obtaining the 2-D DFT of images or the "steady- flow" 2-D DFT of sequences of 2-D images. The algorithm is summarized in Fig. 2, where the 2-D exponential basis functions $(1/\sqrt{N_1 N_2})e^{j2\pi((ln_1/N_1)+(qn_2/N_2))}$ at

Manuscript submitted August 27, 1990; revised October 8, 1991, July 31, 1992, and December 31, 1992. This work was supported by Micronet, the Federal Centre of Excellence on Microelectronic Devices, Circuits and Systems of the Natural Sciences and Engineering Research Council of Canada. This paper was recommended by Associate Editor D. C. Farden.

The authors are with the Department of Electrical and Computer Engineering, University of Calgary, Calgary, Alberta, Canada T2N 1N4.

IEEE Log Number 9208314.

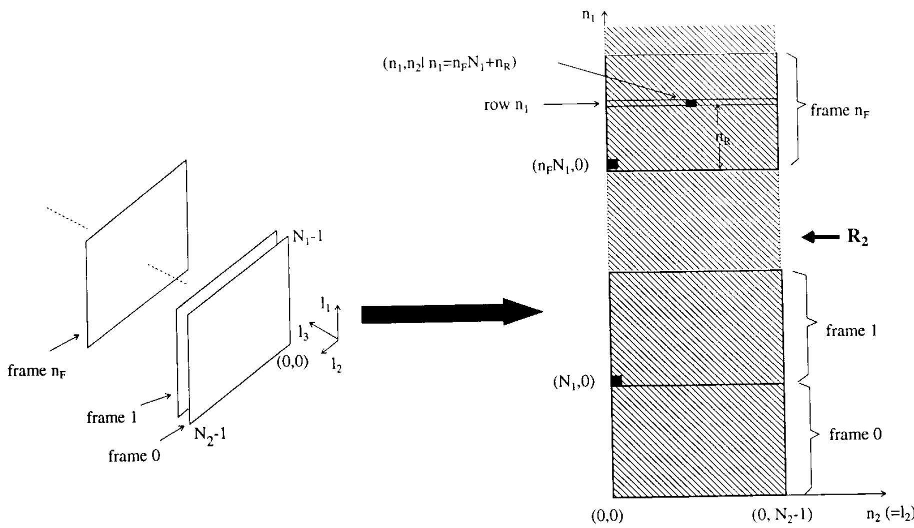

Fig. 1. Two-dimensional representation of an image sequence. (a) 2-D frame sequence. (b) 2-D infinite strip representation $x(n_1, n_2)$.

the outputs of the multipliers are adaptively weighted by the 2-D weights $[w_{l,q}]$. The (desired) input signal $x(n_1, n_2)$ is the sequence of 2-D images and the $N_1 \times N_2$ output matrix $[o_{l,q}]$ is then the required 2-D DFT.

II. The Proposed 2-D Complex LMS Algorithm

The infinite sequence of $N_1 \times N_2$ 2-D images in Fig. 1 may be represented as an infinite length 2-D strip of width N_2, as shown. This 2-D strip $x(n_1, n_2)$ has region of support $\boldsymbol{R}_2 \equiv \{n_1, n_2 \mid 0 \le n_1, 0 \le n_2 \le (N_2 - 1), N_2 < +\infty\}$ and, as shown in Fig. 1, the variables n_F and n_R may be used to point at the data sample $x(n_1, n_2)$ in row n_R of frame n_F, where $n_1 = n_F N_1 + n_R$. This infinite strip $\boldsymbol{R}_2$ has frame boundaries at integer multiples of N_1.

The direction of recursion for the 2-D LMS algorithm is assumed to be pixel-by-pixel in Fig. 1, starting at $(n_1, n_2) = (0, 0)$ and proceeding first along row 0, then row 1, etc. to row $(N_1 - 1)$, in order to coincide with the order of transmission of the conventional signal. The final computation in frame $n_F = 0$ is therefore at $(n_1, n_2) = (N_1 - 1, N_2 - 1)$, after which the next computation is at $(n_1, n_2) = (N_1, 0)$ and is the first computation in frame $n_F = 1$. With this notation and ordering of computations, the application of the 2-D LMS algorithm to image sequences is an algebraically consistent extension of the 1-D case [1].

2.1. The 2-D Adaptive Linear Combiner

Assume that a 2-D adaptive linear combiner has the input signal $\boldsymbol{U}(n_1, n_2) = [u_{l,q}(n_1, n_2)]$ and the weight matrix $\boldsymbol{W}(n_1, n_2) = [w_{l,q}(n_1, n_2)]$, where $u_{l,q}(n_1, n_2)$ and $w_{l,q}(n_1, n_2)$ respectively denote the elements in the lth row and qth column of $\boldsymbol{U}(n_1, n_2)$ and $\boldsymbol{W}(n_1, n_2)$, and $l = 0, 1, \cdots, N_1 - 1$ and $q = 0, 1, \cdots, N_2 - 1$.

The output signal $\hat{x}(n_1, n_2)$ of the adaptive linear combiner is then

$$\hat{x}(n_1, n_2) = \sum_{m=0}^{N_1-1} \sum_{n=0}^{N_2-1} u_{m,n}(n_1, n_2) w_{m,n}(n_1, n_2) \tag{2.1}$$

and the error signal $\varepsilon(n_1, n_2)$ required for adaptation is defined as

$$\varepsilon(n_1, n_2) = x(n_1, n_2) - \hat{x}(n_1, n_2). \tag{2.2}$$

2.2 The 2-D Complex LMS Algorithm Applied to a Sequence of 2-D Images

The general idea of the proposed 2-D complex LMS algorithm is to adapt the weight matrix $\boldsymbol{W}(n_1, n_2)$ as the recursion proceeds to minimize the mean squared error using a similar strategy to the 1-D complex LMS algorithm [2]. To derive the LMS algorithm, the gradient of the mean squared error is estimated by the instantaneous gradient of the squared error at each iteration and the method of steepest descent is then applied [3]. It is easy to observe that the complex LMS algorithm should be able to adapt the real and imaginary parts of $\boldsymbol{W}(n_1, n_2)$ simultaneously to minimize both the real and the imaginary parts of $\varepsilon(n_1, n_2)$ in the least mean square sense.

Let $\boldsymbol{W}(n_1, n_2)$ be expressed in terms of its real and imaginary parts, thus

$$\boldsymbol{W}(n_1, n_2) \equiv \boldsymbol{W}_R(n_1, n_2) + j\boldsymbol{W}_I(n_1, n_2). \tag{2.3}$$

Following the conventional LMS optimization strategy, we use the method of steepest descent applied separately to the real and imaginary parts of the weight matrix $\boldsymbol{W}(n_1, n_2)$ to determine the update equations as follows.

$$\boldsymbol{W}'_R = \boldsymbol{W}_R(n_1, n_2) - \mu \nabla_{\boldsymbol{W}_R}(|\varepsilon(n_1, n_2)|^2) \tag{2.4a}$$

and

$$\boldsymbol{W}'_I = \boldsymbol{W}_I(n_1, n_2) - \mu \nabla_{\boldsymbol{W}_I}(|\varepsilon(n_1, n_2)|^2) \tag{2.4b}$$

where $\boldsymbol{W}'$ represents the weight matrix after updating and μ is the adaptation step, so that

$$\boldsymbol{W}' = \boldsymbol{W}(n_1, n_2) - \mu[\nabla_{\boldsymbol{W}_R}(|\varepsilon(n_1, n_2)|^2) + j\nabla_{\boldsymbol{W}_I}(|\varepsilon(n_1, n_2)|^2)]. \tag{2.5}$$

Consider a complex-valued function $f = f(x, y)$, then $|f|^2 = f\bar{f}$ so that

$$\frac{\partial |f|^2}{\partial x} = f\frac{\partial \bar{f}}{\partial x} + \bar{f}\frac{\partial f}{\partial x}, \quad \text{and} \quad \frac{\partial |f|^2}{\partial y} = f\frac{\partial \bar{f}}{\partial y} + \bar{f}\frac{\partial f}{\partial y}. \tag{2.6}$$

With $f = \gamma(x + jy)$, γ a complex constant, we obtain

$$\frac{\partial f}{\partial x} = \gamma, \quad \text{and} \quad \frac{\partial f}{\partial y} = j\gamma \tag{2.7}$$

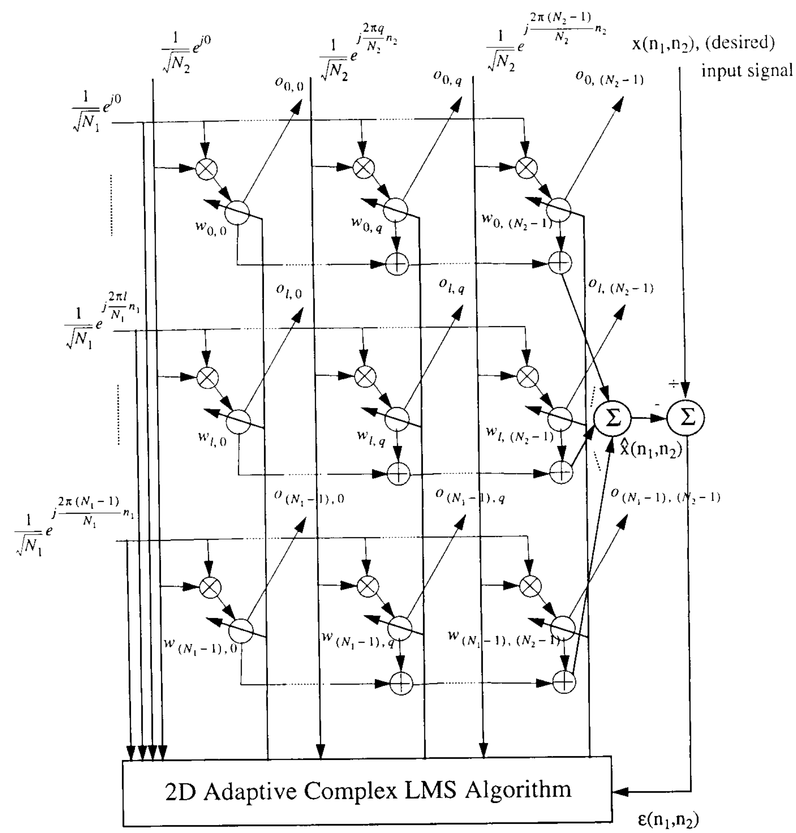

Fig. 2. 2-D LMS spectrum analyzer ("steady-flow" 2-D DFT analyzer).

so that

$$\frac{\partial|f|^2}{\partial x} + j\,\frac{\partial|f|^2}{\partial y} = 2\bar{\gamma} f. \tag{2.8}$$

Applying this result as well as (2.1) and (2.2) to the second term on the right side of (2.5) yields the 2-D complex LMS algorithm as

$$\boldsymbol{W}' = \boldsymbol{W}(n_1, n_2) + 2\mu\varepsilon(n_1, n_2)\overline{\boldsymbol{U}}(n_1, n_2). \tag{2.9}$$

This algorithm can be expressed more specifically for the chosen recursion strategy as

$$\boldsymbol{W}(n_1, n_2 + 1) = \boldsymbol{W}(n_1, n_2) + 2\mu\varepsilon(n_1, n_2)\overline{\boldsymbol{U}}(n_1, n_2), \quad 0 \le n_1 < +\infty,\ 0 \le n_2 < N_2 - 1 \tag{2.10a}$$

and

$$\boldsymbol{W}(n_1 + 1, 0) = \boldsymbol{W}(n_1, N_2 - 1) + 2\mu\varepsilon(n_1, N_2 - 1)\overline{\boldsymbol{U}}(n_1, N_2 - 1), \quad 0 \le n_1 < +\infty. \tag{2.10b}$$

(This result may be generalized to the M-dimensional case ($M > 2$), but is pursued here only for the 2-D case). In the following section, it is shown that this 2-D complex LMS algorithm may be used to compute the weighted 2-D spectrum of a sequence of images and also the 2-D DFT of a single image frame.

III. The Relationship Between the 2-D DFT and the 2-D Complex LMS Algorithm

The relationship between (2.10) and the 2-D DFT of image sequences is now developed. It is now shown that the above 2-D Complex LMS Algorithm may be used to implement a 2-D LMS Spectrum Analyzer which, for a particular choice of μ, leads to the exact computation of the 2-D DFT.

3.1. The 2D LMS Spectrum Analyzer

The proposed 2-D spectrum analyzer is shown in Fig. 2, where the input signal of the 2-D adaptive linear combiner is a $N_1 \times N_2$ 2-D complex phasor sequence array given by

$$\boldsymbol{U}(n_1, n_2) = \boldsymbol{P}1(n_1)\boldsymbol{P}2^T(n_2) \tag{3.1}$$

where

$$\boldsymbol{P}1(n_1) = \frac{1}{\sqrt{N_1}}\left[1, e^{j\frac{2\pi}{N_1}n_1}, \ldots, e^{j\frac{2\pi(N_1-1)}{N_1}n_1}\right]^T$$

and

$$\boldsymbol{P}2(n_2) = \frac{1}{\sqrt{N_2}}\left[1, e^{j\frac{2\pi}{N_2}n_2}, \ldots, e^{j\frac{2\pi(N_2-1)}{N_2}n_2}\right]^T$$

We define the *trailing block* with respect to (n_1, n_2) as the $N_1 \times N_2$ sample points that *trail* the sample point (n_1, n_2) in the recursion as shown shaded in Fig. 3(a). It will be useful to consider the trailing block $\boldsymbol{X}_{\mathrm{TB}}(n_1, n_2)$ in the *rectangular form* $\boldsymbol{X}_{\mathrm{RTB}}(n_1, n_2)$ shown in Fig. 3(b). Clearly the rectangular form of $\boldsymbol{X}_{\mathrm{TB}}(n_1, n_2)$ does not then correspond to an image frame except where $(n_1, n_2) = (n_F N_1, 0)$, in which case, $\boldsymbol{X}_{\mathrm{TB}}(n_F N_1, 0)$ corresponds to frame number $(n_F - 1)$ in the original sequence. The concept of the *trailing*

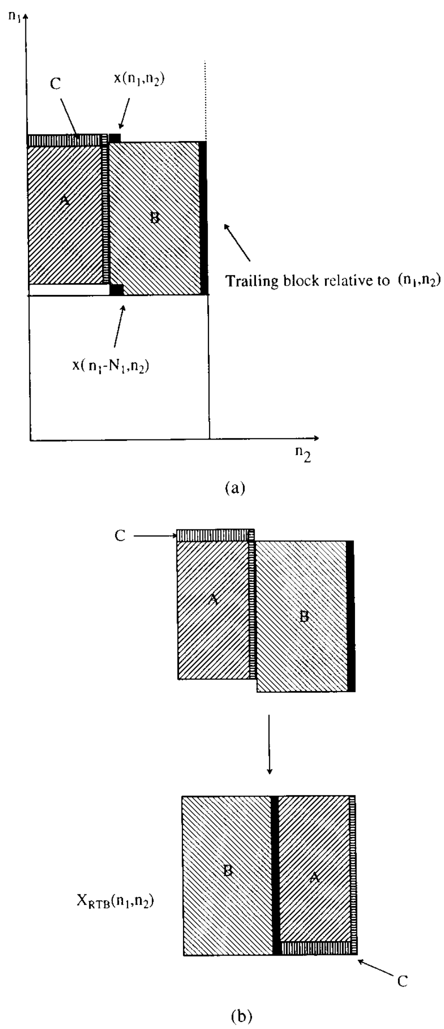

Fig. 3. Definition of the trailing block $X_{\mathrm{TB}}(n_1, n_2)$ and its regular form $X_{\mathrm{RTB}}(n_1, n_2)$. (a) Trailing block $X_{\mathrm{TB}}(n_1, n_2)$. (b) Rectangular form $X_{\mathrm{RTB}}(n_1, n_2)$ of $X_{\mathrm{TB}}(n_1, n_2)$.

block can be used in the tracking of dynamic objects in the sequence of 2-D images and 2-D frequency-domain adaptive filters [6].

It is now shown that the system in Fig. 2 does indeed operate on a sequence of 2-D images as a 2-D spectrum analyzer. For $\mu = 1/2$, the output matrix $O(n_1, n_2)$ of this analyzer is exactly the 2-D DFT [4] of the rectangular form of the trailing block $X_{\mathrm{TB}}(n_1, n_2)$ and the output of the analyzer is the $N_1 \times N_2$ matrix

$$O(n_1, n_2) = \mathrm{Diag}\,(P1(n_1))W(n_1, n_2)\,\mathrm{Diag}\,(P2(n_2)) \tag{3.2}$$

as shown in Fig. 2.

Let $i = n_1 N_2 + n_2$ and $M = N_1 N_2$. Define

$$V^T(i) = [W_0^T, \cdots, W_m^T, \cdots, W_{N_1-1}^T] \tag{3.3}$$

and

$$\rho^{\mathrm{T}}(\mathrm{i}) = [R_0^{\mathrm{T}}, \cdots, R_{\mathrm{m}}^{\mathrm{T}}, \cdots, R_{\mathrm{N}_1-1}^{\mathrm{T}}] \tag{3.4}$$

where W_m^T and R_m^T are the mth rows of $W(n_1, n_2)$ and $U(n_1, n_2)$, respectively. Then the update equation (2.10) may be expressed as a 1-D algorithm:

$$V(i+1) = V(i) + 2\mu\xi(i)\overline{\rho}(i) \tag{3.5}$$

or

$$V(i+1) = (I - 2\mu\overline{\rho}(i)\rho^T(i))V(i) + 2\mu\chi(i)\overline{\rho}(i) \tag{3.6}$$

where $\chi(i) = x(n_1, n_2)$, and

$$\xi(i) = \varepsilon(n_1, n_2) = \chi(i) - V^T(i)\rho(i). \tag{3.7}$$

Noting that $\rho^T(k)\overline{\rho}(m) = 1$ if $k = m + lM$ for some integer l and 0 otherwise, and assuming $V(0) = 0$, we obtain

$$V(i) = 2\mu \sum_{l=0}^{\infty} (1 - 2\mu)^l \sum_{m=i-(l+1)M}^{i-lM-1} \chi(m)\overline{\rho}(m) \tag{3.8}$$

where we have assumed $\chi(m) = 0$ for $m < 0$. Note that if $\mu = 1/2$, then all but the $l = 0$ term are zero. Letting $\mu = 1/2$, and $m = m_1 N_2 + m_2$, we find

$$\begin{aligned} W(n_1, n_2) = &\sum_{m_1=n_1-N_1+1}^{n_1-1} \sum_{m_2=0}^{n_2-1} Q(m_1, m_2) \\ &+ \sum_{m_1=n_1-N_1}^{n_1-1} \sum_{m_2=n_2}^{N_2-1} Q(m_1, m_2) + \sum_{m_2=0}^{n_2-1} Q(n_1, m_2) \end{aligned} \tag{3.9}$$

where $Q(m_1, m_2) = x(m_1, m_2)\overline{P1}(m_1)\overline{P2}^T(m_2)$. From (3.2) and (3.9), it follows that

$$O(n_1, n_2) = \frac{1}{\sqrt{N_1 N_2}} \sum_{m=0}^{N_1-1} \sum_{n=0}^{N_2-1} g(m, n)\overline{P1}(m)\overline{P2}^T(n) \tag{3.10}$$

where

$$g(m, n) = \begin{cases} x(m + n_1 - N_1, n + n_2 - N_2), \\ \quad 1 \le m \le N_1 - 1, \\ \quad N_2 - n_2 \le n \le N_2 - 1, & (3.11a) \\ x(m + n_1 - N_1, n + n_2), \\ \quad 0 \le m \le N_1 - 1, & (3.11b) \\ \quad 0 \le n \le N_2 - n_2 - 1, & (3.11c) \\ x(n_1, n + n_2 - N_2), \\ \quad m = 0, \\ \quad N_2 - n_2 \le n \le N_2 - 1. \end{cases}$$

Consequently, $O(n_1, n_2)$ is exactly the 2-D DFT of $X_{\mathrm{RTB}}(n_1, n_2) = [g(m, n)]$. If $(n_1, n_2) = (n_F N_1, 0)$, then $g(m, n) = x(m + (n_F - 1)N_1, n)$ and $X_{\mathrm{RTB}}(n_1, n_2) = X_{\mathrm{TB}}(n_1, n_2)$. We then have, in general,

$$O(n_F N_1, 0)_{\mu=\frac{1}{2}} = \mathbf{2\,DDFT}\,[X_{\mathrm{TB}}(n_F N_1, 0)]. \tag{3.12}$$

That is, with $\mu = 1/2$, *the output matrix* $O(n_1, n_2)$ *at points* $(n_F N_1, 0)$ *is exactly the 2-D DFT of frame number* $(n_F - 1)$. *Therefore, this method may be used to compute the 2-D DFT of the 2-D frames in an infinite sequence of frames. The output matrix* $O(n_F N_1, 0)$ *of the 2-D LMS spectrum analyzer is then the required 2-D DFT.*

3.2. The "Steady-flow" 2-D DFT

It is shown from (3.10) and (3.11) that $O(n_1, n_2)$, for $\mu = 1/2$, is the 2-D DFT of $X_{\mathrm{RTB}}(n_1, n_2) = [g(m, n)]$. Comparing (3.11) with the diagrammatic partitioning of the trailing block in Fig. 3, (3.11a) corresponds to subblock A, (3.11b) to subblock B, and (3.11c) to subblock C. After [1], we call this kind of 2-D DFT the "steady-flow" 2-D DFT. The output of the spectrum analyzer $O(n_1, n_2)$ is equal to the "steady-flow" 2-D DFT defined above at any point (n_1, n_2).

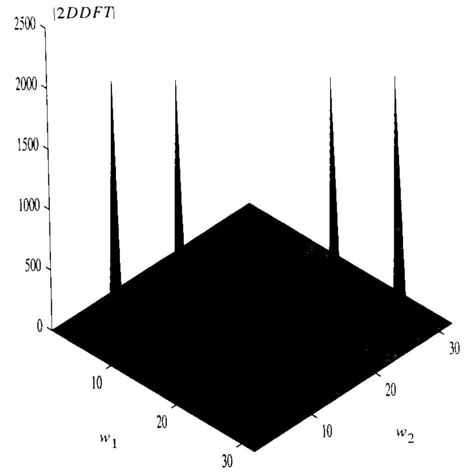

Fig. 4. Magnitude spectrum of the 2-D LMS/DFT of $x(n_1, n_2) = 128 (\sin(2\pi(5n_1 + 5n_2)/32) + \cos(2\pi(10n_1 + 10n_2)/32))$.

It represents the steady change of the 2-D spectrum of the sequence of images. It can be used in the tracking of dynamic objects in image sequence and 2-D frequency-domain adaptive filters [6].

By using the relations in (3.2) and (3.8), the output $O(n_1, n_2)$ of the "steady-flow" 2-D DFT spectrum analyzer can be written as the geometric sum:

$$O(n_1, n_2) = 2\mu \sum_{l=0}^{\infty} (1 - 2\mu)^l \mathbf{2DDFT}[\boldsymbol{X}_{\mathrm{RTB}}(n_1 - lN_1, n_2)]. \tag{3.13}$$

This geometric coherent-average "steady-flow" 2-D DFT, a new form of the DFT, is evaluated from (3.13) by choosing μ as an appropriate "memory factor" of the 2-D DFT's of previous frames. The stability of the complex 2-D LMS/DFT algorithm is guaranteed iff $|1 - 2\mu| < 1$, requiring that $0 < \mu < 1$. Larger deviations from the value of $\mu = 1/2$ in this interval imply a long "memory factor" and, at $\mu = 1/2$, the "memory factor" is zero according to (3.13).

3.3 Numerical Verification

Consider the simple test signal

$$x(n_1, n_2) \equiv 128\left(\sin\left(2\pi\frac{(5n_1 + 5n_2)}{32}\right) + \cos\left(2\pi\frac{(10n_1 + 10n_2)}{32}\right)\right) \tag{3.14}$$

where the image size is given by $(N_1, N_2) \equiv (32, 32)$.

The magnitude of the 2-D LMS/DFT of $x(n_1, n_2)$, computed according to the algorithm proposed in this contribution, is shown in Fig. 4 and corresponds with that obtained using the conventional 2-D DFT method of calculation.

3.4. Remarks On Computational Complexity

For complex data $x(n_1, n_2)$, the conventional 2-D FFT of the $N_1 \times N_2 \boldsymbol{X}_{\mathrm{RTB}}(n_1, n_2)$ requires $2N_1N_2 \log_2(N_1N_2)$ real multiplications and $3N_1N_2 \log_2(N_1N_2)$ real additions, whereas it can easily be shown that the 2-D LMS/DFT algorithm proposed in this paper requires $8N_1N_2$ real multiplications and $10N_1N_2 - 4N_1 - 4N_2 + 6$ real additions for $\boldsymbol{X}_{\mathrm{RTB}}(n_1, n_2)$. Therefore, for $N_{1,2} \gg 1$, the 2-D LMS/DFT algorithm is significantly more efficient in trailing block applications, such as adaptive filtering where the trailing block concept ensures smooth variations in the adaptive weights retained over translations between very different image frames. Of course, for calculating only the 2-D DFT of separate image frames, the 2-D LMS/DFT algorithm is significantly less efficient because the recursion must proceed through N_1N_2 steps in (n_1, n_2) between frame boundaries. It can be shown, however, that by updating the 2-D complex LMS algorithm on a row-by-row (rather than pixel-by-pixel) basis, this 2-D LMS/DFT may be vectorized and thereby the processing speed increased by the factor $N_2 - 1$ [6].

IV. Conclusions

In this contribution, Widrow's 1-D spectrum analyzer has been extended to the 2-D case to establish the relationship between the 2-D DFT and the 2-D complex LMS algorithm. It results directly in the 2-D LMS spectrum analyzer, which is a new recursive method to calculate the 2-D DFT. Using the conventional frame-by-frame computations of the 2-D DFT's to compute the geometric coherent-average "steady-flow" 2-D DFT ((3.13)) would require that the 2-D DFT's of many previous frames to be stored in memory whereas the proposed method requires no memory storage of previous image samples because it is totally recursive. However, internally, the algorithm does require that $\boldsymbol{W}(n_1, n_2)$, $\boldsymbol{P}1(n_1)$, and $\boldsymbol{P}2(n_2)$ be stored. *The 2-D LMS algorithm has the advantage that it allows concurrent computations of the $N_1 \times N_2$ terms in the update matrix equation (2.10) and therefore the potential for very fast parallel computations of the 2-D DFT.*

Acknowledgment

The authors wise to thank the anonymous reviewers for their valuable comments. They are particularly grateful to Associate Editor Dr. David C. Farden's personal comments.

References

[1] B. Widrow, P. Baudrenghien, M. Vetterli, and P. F. Titchener, "Fundamental relations between the LMS algorithm and the DFT," *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 814–820, July 1987.
[2] B. Widrow, J. M. McCool, and M. Ball, "The complex LMS algorithm," *Proc. IEEE*, vol. 63, pp. 719–720, Apr. 1975.
[3] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.
[4] A. V. Oppenheim, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.
[5] M. Bellanger, *Digital Processing of Signals*. New York: Wiley, 1989.
[6] B. Liu, "The complex LMS algorithm applied to the DFT and adaptive filtering," M.Sc. thesis, Univ. Calgary, Canada, Nov. 1990.