Denote the joint pdf of the n-variate normal random vector X with mean μ and variance matrix Σ by $\phi_n(x \mid \mu, \Sigma)$.

Theorem: Let e be the n-variate normal random vector with mean $\mathbf{0}$ and covariance matrix $\delta^2 I$. Then, the joint pdf minimizing the time-domain KL information number

$$I(f; \phi_n) = \int f(e) \ln \frac{f(e)}{\phi_n(e \mid \mathbf{0}, \delta^2 I)} de$$

subject to the first $p + 1$ autocovariance constraints

$$\sigma(0) = \alpha_0, \sigma(1) = \alpha_1, \cdots, \sigma(p) = \alpha_p$$

is $\phi_n(e \mid \mathbf{0}, S_n)$.

Proof: The KL information number can be decomposed into two parts:

$$\int f(e) \ln \frac{f(e)}{\delta\phi_n(e \mid \mathbf{0}, \delta^2 I)} de$$
$$= \int f(e) \ln \frac{f(e)}{\phi_n(e \mid \mathbf{0}, V_n)} de + \int f(e) \ln \frac{\phi_n(e \mid \mathbf{0}, V_n)}{\phi_n(e \mid \mathbf{0}, \delta^2 I)} de.$$

The first integral of the RHS is greater than or equal to 0 by Jensen's inequality. It equals 0 if the joint pdf is $\phi_n(e \mid \mathbf{0}, V_n)$. The second integral of the RHS satisfies the following:

$$\int f(e) \ln \frac{\phi_n(e \mid \mathbf{0}, V_n)}{\phi_n(e \mid \mathbf{0}, \delta^2 I)} de$$
$$= \frac{n}{2} \ln \delta^2 - \frac{1}{2} \ln \det (V_n) - \frac{1}{2} \int f(e)(e' V_n^{-1} e - e'e) \, de$$
$$= \frac{n}{2} \ln \delta^2 - \frac{1}{2} \ln \det (V_n) - \frac{1}{2} \left\{ \operatorname{tr}(V_n^{-1} V_n) - \operatorname{tr}(V_n) \right\}$$
$$= -\frac{1}{2} \ln \det (V_n) - \frac{n}{2} (\ln \delta^2 - 1 + \alpha_0).$$

Thus, our problem becomes to maximize $\det(V_n)$ subject to the first $p + 1$ autocovariance constraints. It is known [8] that the maximum of $\det(V_n)$ is attained when $\sigma(j) = \alpha_j$ for $j = p + 1, p + 2, \cdots$. It finishes the proof. Q.E.D.

III. Comments

The theorem means that the Gaussian AR(p) process is the closest in the time-domain Kullback-Leibler sense to independently, identically, and normally distributed random variables subject to the first $p + 1$ autocovariance constraints. Thus, if the residuals found from parametric time series modeling or regression modeling are autocorrelated, the theorem implies that it would be better to regard them as from an AR process. Also, it implies that the KL spectrum estimate is theoretically more reasonable than the initial estimate in the time-domain Kullback-Leibler sense.

References

[1] T. W. Anderson, *The Statistical Analysis of Time Series.* New York: Wiley, 1971.
[2] G. M. Jenkins and D. G. Watts, *Spectral Analysis and its Applications.* San Francisco: Holden-Day, 1968.
[3] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, pp. 79-86, 1951.
[4] J. M. Van Campenhout and T. M. Cover, "Maximum entropy and conditional probability," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 483-489, 1981.
[5] B. S. Choi, "An information theoretic spectral density," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-38, pp. 717-721, 1990.
[6] S. Zohar, "The solution of a Toeplitz set of linear equations," *J. Ass. Comput. Mach.*, vol. 21, 1974.
[7] S. M. Kay and S. L. Marple, Jr., "Spectrum analysis—a modern perspective," *Proc. IEEE*, vol. 69, pp. 1380-1419, 1981.
[8] B. S. Choi, "On the relation between the maximum entropy probability density function and the autoregressive model," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1659-1661, 1986.

A Class of Fast Gaussian Binomial Filters for Speech and Image Processing

Richard A. Haddad and Ali N. Akansu

***Abstract*—The Gaussian Binomial filters are a family of one- and two-dimensional FIR filters with binary-valued coefficients $(-1, 1)$. The family can function as a bank of filters, with taps corresponding to low-pass, band-pass with differing center frequencies, and high-pass filters. The low-pass filter (1D and 2D) has a Gaussian shaped amplitude frequency response and a binomial impulse response which approximates a Gaussian point spread function in the (time) spatial domain.**

We present an efficient, in-place algorithm for the batch processing of linear data arrays. These algorithms are efficient, easily scaled, and have no multiply operations.

They are suitable as front end filters for a bank of quadrature mirror filters, and pyramid coding of images. In the latter application, the Binomial filter was used as the low-pass filter in pyramid coding of images, and compared with the Gaussian filter devised by Burt. The Binomial filter yielded a slightly larger SNR in every case tested. More significantly, for an $(L + 1) \times (L + 1)$ image array processed in $(N + 1) \times (N + 1)$ subblocks, the fast Burt algorithm requires a total of $2(L + 1)^2 N$ adds and $2(L + 1)^2 (N/2 + 1)$ multiplies. The Binomial algorithm requires $2L^2 N$ adds and zero multiplies.

I. Introduction

Over the past decade and half, several investigators have sought to design FIR filters with finite precision coefficients. The extreme case is the class of FIR filters with coefficients quantized to the ternary set $(-1, 0, 1)$. The binary transversal filter of Lockhart [3], and the one-dimensional Binomial filters introduced by Haddad [4] were early members of this class. These efforts were followed by the papers of Van Gerwen *et al.* [5], Benvenuto *et al.* [6], and Bateman and Liu [7].

A common theme among these structures is that the filter can be configured as a tapped delay line followed by a first- or second-order accumulator (or "resonator") of various sorts. The tap coefficients were selected from the ternary set $(-1, 0, 1)$ to cancel the poles in the accumulator and thus render the overall filter as FIR. Benvenuto *et al.* [8] described how the remaining coefficient values are determined by a dynamic programming algorithm to minimize some performance measure. The motivation behind these ideas can be traced to delta modulation signal encoding concepts [6], wherein the sampling frequency must be increased to compensate for coarse signal quantization. In the present context, the clock rate and the number of coefficients are increased considerably to obtain the desired filter response.

The serial form of the Binomial filter [4] is shown in Fig. 1. Note that there is a cancellation of the poles in the resonator sections by

Manuscript received April 13, 1989; revised April 8, 1990.
R. A. Haddad is with the Department of Electrical Engineering and Computer Science, Polytechnic University, Hawthorne, NY 10532.
A. N. Akansu is with the Electrical and Computer Engineering Department, New Jersey Institute of Technology, Center for Communications and Signal Processing Research, Newark, NJ 07102.
IEEE Log Number 9041617.

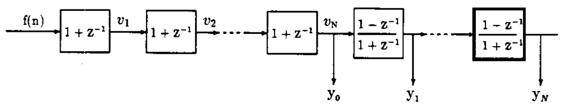


Fig. 1. Sequential canonic processor for Binomial filters.

the zeros in the nonrecursive Binomial part. Each tap ouptut y_r corresponds to a distinct filter. Hence this serial structure realizes a bank of filters.

Burt [1] and Burt and Adelson [2] devised a different kind of filter, the "hierarchical discrete correlation" filter, which is also capable of functioning as a low-pass or band-pass processor with Gaussian-like magnitude frequency response characteristics. The Burt filter which can provide fast correlations with nonternary valued coefficients, has been used in image pyramids [2]. The Binomial filter is compared with the Burt filter in an image pyramid application. The results show that the Binomial filter is slightly better than the Burt low pass in performance, but enormously superior in computational efficiency and speed.

II. Canonic Realizations of the Binomial Family

The Binomial family of sequences $x_r(k)$ is defined in [4] by

$$x_r(k) = \binom{N}{k} H_r(k), \quad r = 0, 1, \cdots, N \tag{1}$$

$$H_r(k) = \sum_{n=0}^{r} (-2)^n \binom{r}{n} \frac{k^{(n)}}{N^{(n)}} = H_k(r) \tag{2}$$

$$k^{(n)} = \begin{cases} k(k-1)\cdots(k-n+1), & n \geq 1 \\ 1, & n = 0 \end{cases} \tag{3}$$

where $\binom{r}{n}$ is the binomial coefficient, and $\{H_r(k)\}$ is the family of discrete Hermite polynomials.

The Binomial sequences $\{x_r(k)\}$ and the Hermite polynomials $\{H_r(k)\}$ are orthogonal on $[0, N]$ with respect to the weighting functions $\binom{N}{k}^{-1}$ and $\binom{N}{k}$, respectively.

$$\sum_{k=0}^{N} \binom{N}{k}^{-1} x_r(k) x_s(k) = \sum_{k=0}^{N} \binom{N}{k} H_r(k) H_s(k) = 2^N \binom{N}{r}^{-1} \delta_{r-s} \tag{4}$$

where δ_{r-s} is the Kronecker delta.

The key property of the Binomial family for signal processing purposes is the recursion formula

$$x_{r+1}(k) = -x_{r+1}(k-1) - x_r(k-1) + x_r(k)$$
$$0 \leq k \leq N, \quad 0 \leq r \leq N-1 \tag{5}$$

with initial value, and initial sequence

$$x_r(-1) = 0, \quad 0 \leq r \leq N$$

$$x_0(k) = \binom{N}{k}, \quad 0 \leq k \leq N. \tag{6}$$

The transform of these 1D Binomial sequences is

$$X_r(z) = \left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) X_{r-1}(z) = (1 - z^{-1})^r (1 + z^{-1})^{N-r} \tag{7}$$

since

$$X_0(z) = (1 + z^{-1})^N.$$

The corresponding frequency response is

$$X_r(e^{j\omega}) = A_r(w) e^{j\theta_r(\omega)} \tag{8}$$

with magnitude and phase

$$A_r(\omega) = (2)^N (\sin \omega/2)^r (\cos \omega/2)^{N-r}$$

$$\theta_r(\omega) = r\frac{\pi}{2} - N\frac{\omega}{2}. \tag{9}$$

In the foregoing, ω is the normalized frequency $\omega = \Omega T$, and T is the spacing between samples (or pixels).

The phase characteristic is linear, and the magnitude response has a slightly asymmetric bandpass shape about a center frequency

$$\omega_m = 2 \sin^{-1} \sqrt{r/N}. \tag{10}$$

For N large, $A_r(w)$ is almost Gaussian [4] with half-power bandwidths

$$\mathrm{BW} = \begin{cases} 2.34/\sqrt{N}, & r > 0 \\ 1.66/\sqrt{N}, & r = 0. \end{cases} \tag{11}$$

The transfer functions of the Binomial family can be expressed in two alternate forms, each suggesting a different filter realization. The sequential representation

$$X_r(z) = \left(\frac{1 - z^{-1}}{1 + z^{-1}}\right)^r X_0(z)$$

$$X_0(z) = (1 + z^{-1})^N \tag{12}$$

suggests the network of Fig. 1, in which the data stream $\{f(0), f(1), \cdots\}$ is fed sequentially in time and processed via the recurrences implicit in that network. The low-pass output is obtained at the $y_0(n)$ tap. The rth bandpass output is picked off at $y_r(n)$ and the high-pass filtered signal at $y_N(n)$. Note that only additions and subtractions are performed here and that an entire bank of filters is realized simultaneously. Because of the pole-zero cancellation implicit in Fig. 1, the initial states must be set to zero, i.e., $v_j(-1) = 0, j = 1, \cdots, N$ and $y_r(-1) = 0$, for $r = 1, 2, \cdots, N$.

The batch canonic representation shown in Fig. 2 is based on the purely nonrecursive representation

$$X_r(z) = (1 + z^{-1})^{N-r} (1 - z^{-1})^r \tag{13}$$

which depicts the bandpass filter as $(N - r)$ stages of the add operator $(1 + z^{-1})$, followed by r stages of the difference operator $(1 - z^{-1})$. This form lends itself to batch processing of the data, or, as it is termed in the literature [10], [11], to block implementation of the FIR algorithm. Rather than applying the signal $\{f(n)\}$ sequentially, to Fig. 2, we instead store $(L + 1)$ successive samples as the linear array,

$$f^T = [f(0), f(1), \cdots, f(L)] \tag{14}$$

and apply successive sum and difference matrix operators to this input vector. For the first stage we want

$$v_1(0) = f(0) \qquad \text{since } f(-1) = 0$$
$$v_1(1) = f(1) + f(0)$$
$$\vdots$$
$$v_1(L) = f(L) + f(L-1)$$

or

$$v_1 = Sf \tag{15}$$

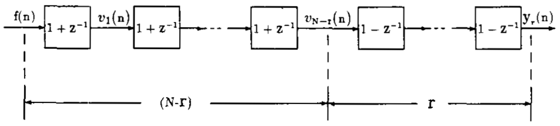


Fig. 2. Batch canonic processor for Binomial filters.

where

$$S = \begin{bmatrix} 1 & 0 & 0 & \cdot & \cdot & & 0 \\ 1 & 1 & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & 1 & 1 & \cdot & \cdot & \cdot & 0 \\ 0 & 0 & \cdot & \cdot & 0 & 1 & 1 \end{bmatrix}. \tag{16}$$

S is the transmission matrix for a single add stage. For any differencing stage, $(1 - z^{-1})$, the transmission matrix derived from

$$g(k) = h(k) - h(k-1), \qquad k = 0, 1, \cdots, L$$
$$g(-1) = 0 \tag{17}$$

is

$$g = Dh \tag{18}$$

where

$$h^T = [h(0), h(1), \cdots, h(L)]$$
$$g^T = [g(0), g(1), \cdots, g(L)]$$

and

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & & & & & & \\ 0 & 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix}. \tag{19}$$

Combining the add and difference operators gives the bandpass filter

$$y_r = D^r S^{N-r} f \tag{20}$$

shown in the flowgraph of Fig. 3.

The transmission matrices, D and S, commute (as do the transmission matrices for all linear time-invariant systems). The signal after successive add operators could get large, with $(2)^{N-r}$ as the upper bound. This can be reduced by combining add and difference operators wherever possible. Thus, for $r < N/2$, we can use

$$y_r = (D^r S^r) S^{N-2r} f$$
$$= [(DS)^r S^{N-2r}] f. \tag{21}$$

The DS operator represents the symmetric bandpass filter $(1 - z^{-2})$, with normalized center frequency $\omega_0 = \pi/2$. Explicitly,

$$DS = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & & 0 & 0 & 0 & 0 \\ \vdots & & & & & & & & \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 & -1 \end{bmatrix}. \tag{22}$$

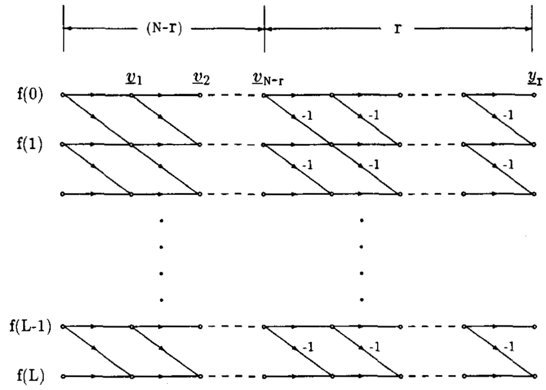


Fig. 3. Band-pass filter for batch-canonic processor. Unmarked branches have unity gain.

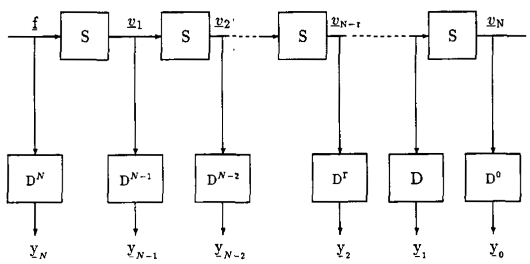


Fig. 4. Bank of batch-canonic Binomial filters.

A parallel bank of Binomial filters can be realized by implementing (20) for each r in parallel. A structure for achieving this is shown in Fig. 4. The vector outputs $y_0, y_1, \cdots, y_N$ can be obtained simultaneously. There is an inherent delay of $(L + 1)$ clock pulses to fill the f array. After which, the output vectors are available in the computation time required to execute the D and S oeprations, each of which can be done either in parallel, or in a time-shared mode. Great speeds are thus possible, since the entire structure consists only of add and subtract operations.

III. The Two-Dimensional Binomial Batch Algorithm

The 2D Binomial sequences, denoted as $\{x_{rs}(m, n)\}$ on the interval $[0 \le m, n \le N]$ are defined in [9] as the separable product of the 1D sequences

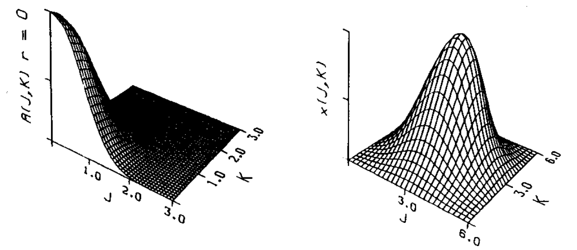
$$x_{rs}(m, n) = x_r(m) x_s(n), \qquad 0 \le r, s \le N \tag{23}$$

$$X_{rs}(z_1, z_2) = X_r(z_1) X_s(z_2) = [1 - z_1^{-1})^r (1 + z_1^{-1})^{N-r}] \cdot [1 - z_2^{-1})^s (1 + z_2^{-1})^{N-s}]. \tag{24}$$

These constitute a family of low-pass, band-pass, and high-pass 2D filters with almost Gaussian envelopes. Sample plots of the Binomial filter responses in the spatial and frequency domains are shown in Figs. 5 and 6, for low-pass and band-pass filters, respectively. Observe that the low-pass filter spatial impulse response, as well as the frequency response, has a Gaussian look to it—as well as it should. For $r = s = 0$, the impulse response is just the product of two binomials

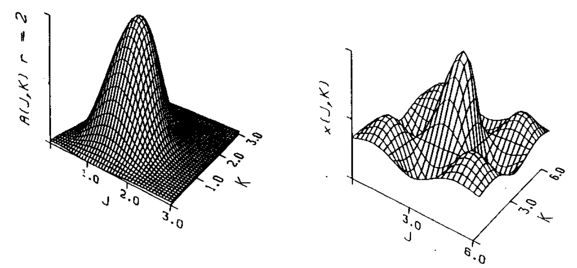
$$\binom{N}{m}\binom{N}{n}$$

each of which is almost one-dimensional Gaussian.



(a) (b)

Fig. 5. (a) Magnitude frequency response and (b) impulse response of low-pass Binomial filter.



(a) (b)

Fig. 6. (a) Magnitude frequency response and (b) impulse response of band-pass Binomial filter.

The 2D Binomial filter can be implemented in the sequential mode or the batch canonic form. The sequential mode is described in [9], while the batch mode is developed here as a straightforward extension of the 1D case. The processing is very simple due to the separability of the operators. N batch row operations are followed by N batch column operations. This is illustrated for the low-pass filter defined by

$$\mathcal{Y}(z_1, z_2) = [(1 + z_1^{-1})(1 + z_2^{-1})]^N \mathcal{F}(z_1, z_2) \tag{25}$$

where $\mathcal{F}(z_1, z_2)$ and $\mathcal{Y}(z_1, z_2)$ are the transforms of the 2D input and output signals $f(m, n)$, $y(m, n)$, respectively. Equation (25) can be decomposed into row and column operations as indicated in Fig. 7:

$$\begin{aligned} \mathcal{V}(z_1, z_2) &= (1 + z_1^{-1})^N \mathcal{F}(z_1, z_2) \\ \mathcal{Y}(z_1, z_2) &= (1 + z_2^{-1})^N \mathcal{V}(z_1, z_2) \end{aligned} \tag{26}$$

where $\mathcal{V}(z_1, z_2)$ is the transform of the intermediate signal $v(m, n)$.

Equation (26) suggests that each row of the input array $[f(m, n)] = F$ can be batch processed using S^N, the low-pass batch canonic operator of (8), to obtain the intermediate array $[v(m, n)] = V$. Then each column of V is processed by S^N to obtain the final array $[y(m, n)] = Y$. Thus

$$\begin{aligned} V &= S^N F \\ Y &= V(S^T)^N \end{aligned} \tag{27}$$

or

$$Y = S^N F (S^T)^N \tag{28}$$

where the superscript T implies a matrix transpose.

These row and column operations are similar to the processing of images by separable image transforms [13].

The band-pass algorithm can be expressed as

$$\begin{aligned} \mathcal{Y}(z_1, z_2) = &[(1 - z_2^{-1})^s (1 + z_2^{-1})^{N-s}] \\ &\cdot [(1 - z_1^{-1})^r (1 + z_1^{-1})^{N-r}] \mathcal{F}(z_1, z_2). \end{aligned} \tag{29}$$

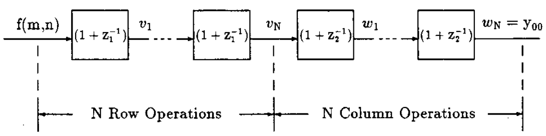


Fig. 7. Two-dimensional low-pass Gaussian filter for batch processing.

Equation (29) can be implemented by applying the band-pass operator $(D^r S^{N-r})$ to the rows of $[f(m, n)]$ to form the intermediate array V. Then the columns of V are operated on by $(D^s S^{N-s})$ to produce the filtered output array Y:

$$Y = [D^r S^{N-r}][F][D^s S^{N-r}]^T. \tag{30}$$

IV. Binomial Low-Pass Filter for Pyramid Coding of Images

In pyramid coding, an image is successively reduced in size by low-pass filtering followed by decimated spatial sampling. As the image is successively reduced to form a pyramid, the difference between the two layers of the pyramid is also calculated. The process continues until the minimum reduced image size is reached. The reduced image on the top of the pyramid (the smallest size) can be used for initial transmission. It can be expanded progressively by adding the difference information between the two consecutive layers of the pyramid. Since the introduction of progressive image transmission by Sloan and Tanimoto [12] various implementation techniques have been developed.

Burt has introduced fast filter transforms for use in image processing [1]. These low-pass filters are used in the "reduce" and "expand" operations of the pyramid image coding technique [2]. In this study, the Binomial low-pass filter is also used in pyramid coding and compared with the low-pass structure proposed by Burt.

In our simulations, the minimum image size is 16×16. A 4×4 vector quantization is used for the two largest size difference images and linear quantizers are employed in the rest of the structure. The test images used are 256×256 monochrome arrays, 8 b/pixel standard images, Lena, Building, etc. The 5×5 and 7×7 2D low-pass filters are used for the comparison of the two filter types in pyramid image coding applications. The impulse responses of the filters used here are given in Table I. The overall coding performance criterion is defined as

$$\mathrm{SNR}_{pp}(\mathrm{dB}) = 10 \log_{10} \left(255^2 / E\left\{\left(p(m, n) - \hat{p}(m, n)\right)^2\right\}\right)$$

where $\hat{p}(m, n)$ is the reconstructed value of the pixel (m, n).

In Table II, a, b refer to the parameters in the Burt filter [1]. The test results shown in Table II also indicate that the Binomial filter matches the performance of the Burt filter in pyramid coding. For an $(L + 1) \times (L + 1)$ image array processed by one pass of an $(N + 1) \times (N + 1)$ separable filter block using row and column operations, the Burt algorithm requires $2(L + 1)^2 N$ adds and $2(L + 1)^2((N/2) + 1)$ multiplies, while the Binomial using the batch processing mode does the same filtering operation in $2NL^2$ adds and zero multiplies. Table III compares the computational burden for a 256×256 image and 5×5 filter.

V. Conclusions

The Binomial filters are a family of very fast low-pass, band-pass, and high-pass 2D filters. They are easy to implement in hardware or software, using only add and subtract operations. There are no multiply operations, and hence no roundoff. Signal bounds can be precalculated and the registers sized, or the operation scaled *a priori* to prevent overflow.

The batch processing mode permits an entire row (or column) to be processed using the S and D operators. The order (or the filter width) determines the number of S and D stages used.

TABLE I
IMPULSE RESPONSES OF BINOMIAL AND BURT FILTERS, FOR $N = 5$, AND $N = 7$

	n	0	1	2	3	4	5	6
$[N = 5]$	BINOMIAL	$\frac{1}{16}$	$\frac{4}{16}$	$\frac{6}{16}$	$\frac{4}{16}$	$\frac{1}{16}$		
	BURT	.025	.25	.45	.25	.025		
$[N = 7]$	BINOMIAL	$\frac{1}{64}$	$\frac{6}{64}$	$\frac{15}{64}$	$\frac{20}{64}$	$\frac{15}{64}$	$\frac{6}{64}$	$\frac{1}{64}$
	BURT	.01	.02	.24	.46	.24	.02	.01

TABLE II
LENA IMAGE

		SNR_{pp} (dB)	
	Bits/pel	Burt's LPF	Binomial LPF
5 × 5 FILTER	1.11	28.80	29.14
$a = 0.45$	1.035	27.57	27.61
7 × 7 FILTER			
$a = 0.46$	1.11	28.43	28.80
$b = 0.24$	1.035	27.27	27.54

TABLE III

	Burt	Binomial
Number of Adds	524,288	520,200
Number of Multiplies	393,216	zero

This latter feature strongly suggests the possibility of coding and decoding images in real time using array processors and a pipeline architecture.

REFERENCES

[1] P. J. Burt, "Fast filter transforms for image processing," *Comput. Graph. Image Processing*, vol. 16, pp. 20–51, 1981.
[2] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. COM-31, no. 4, pp. 532–540, Apr. 1983.
[3] G. B. Lockhart, "Binary transversal filter with quantized coefficients," *Electron. Lett.*, vol. 7, no. 11, pp. 305–307, June 3, 1971.
[4] R. A. Haddad, "A class of orthogonal nonrecursive binomial filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-19, no. 4, pp. 296–304, Dec. 1971.
[5] P. J. Van Gerwen, W. F. Mecklenbrauker, N. A. M. Verhoeckx, W. A. M. Snijders, and H. A. Van Essen, "A new type of digital filter for data transmission," *IEEE Trans. Commun.*, vol. COM-23, pp. 222–234, Feb. 1975.
[6] N. Benvenuto, L. E. Franks, and F. S. Hill, Jr., "Realization of finite impulse response filters using coefficients +1, 0, and −1," *IEEE Trans. Commun.*, vol. COM-22, no. 10, pp. 1117–1125, Oct. 1985.
[7] M. R. Bateman and B. Liu, "An approach to programmable CTD filters using coefficients 0, +1, and −1," *IEEE Trans. Circuits Syst.*, vol. CAS-27, pp. 451–456, June 1980.
[8] N. Benvenuto, L. E. Franks, and F. S. Hill, Jr., "Dynamic programming methods for designing FIR filters using coefficients −1, 0, and +1," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, no. 4, pp. 785–792, Aug. 1986.
[9] R. A. Haddad and B. Nichol, "Efficient filtering of images using binomial sequences," presented at the Int. Conf. ASSP, Glasgow, Scotland, May 1989, pap. 18.M4 7.
[10] C. W. Barnes and S. Shinnaka, "Block shift in variance and block implementation of discrete-time filter," *IEEE Trans. Circuits Syst.*, vol. CAS-27, no. 8, pp. 667–672, Aug. 12, 1980.
[11] R. A. Roberts and C. T. Mullis, *Digital Signal Processing.* Reading, MA: Addison-Wesley, 1987, pp. 433–440.
[12] K. R. Sloan and S. L. Tanimoto, "Progressive refinement of raster images," *IEEE Trans. Comput.*, vol. C-28, pp. 871–874, Nov. 1979.
[13] R. C. Gonzales and P. Wintz, *Digital Image Processing*, second ed. Reading, MA: Addison-Wesley, 1987, pp. 72–77.

Computing Time-Frequency Distributions

Brian Harms

***Abstract*—Recently, numerous strategies have been proposed for computing discrete time-frequency distributions such as the Wigner distribution. The purpose of this correspondence is to point out an efficient and straightforward strategy for computing time-frequency distributions that are members of Cohen's class. The strategy is based on the insightful work by Nuttall which has finally resolved the questions concerning aliasing and required sampling rate for the Wigner distribution.**

I. INTRODUCTION

Interest in joint time-frequency distributions as tools for the study of nonstationary signals is growing [9]. Because the computation and presentation of these distributions nearly always requires a computer or special-purpose digital hardware, interest in efficient computational approaches is also on the rise [1]–[5], [8], [10], [13], [14]. The purpose of this correspondence is to point out an efficient and straightforward strategy for computing time-frequency distributions that are members of Cohen's class [7], [9].

To illustrate the apparent difficulties in computing discrete time-frequency distributions, we will first consider the specific case of the Wigner distribution function (WDF). The continuous-time definition of the WDF of a signal $s(t)$ is

$$W(t, f) = \int_{-\infty}^{\infty} s\left(t + \frac{\tau}{2}\right) s^*\left(t - \frac{\tau}{2}\right) e^{-j2\pi f\tau} \, d\tau. \qquad (1)$$

Available to us are uniformly spaced samples of $s(t)$ which have been acquired over the variable t, with a spacing Δ_t.

By defining

$$R(t, \tau) = s\left(t + \frac{\tau}{2}\right) s^*\left(t - \frac{\tau}{2}\right) \qquad (2)$$

to be the temporal correlation function (TCF), we see that the WDF is a Fourier transform of the TCF. Consequently, we would expect to be able to use an FFT to obtain samples of the WDF from samples of the TCF. However, there is a difficulty. Equation (1) is a Fourier transform over τ not t. For a given value of t, the discrete values of τ at which the TCF is available must necessarily be separated by $2\Delta_t$ due to the symmetry of the arguments in the TCF and the factor of two which scales τ. Consequently, the effective sampling rate has been halved so far as the time variable τ is concerned. This would seem to imply that the original sampling rate should be increased to four times the highest frequency present in the signal $s(t)$.

Manuscript received July 28, 1990; revised September 17, 1990. This work was supported by Motorola, Inc. Government Electronics Group, Chandler, AZ.
The author is with the Department of Electrical and Computer Engineering, Kansas State University, Manhattan, KS 66506.
IEEE Log Number 9041605.

1053-587X/91/0300-0727$01.00 © 1991 IEEE