Vector Space Formulation of Two-Dimensional Signal Processing Operations

WILLIAM K. PRATT

Electrical Engineering Department, Image Processing Institute,
University of Southern California, Los Angeles, California 90007

Communicated by A. Rosenfeld
Received August 15, 1974

Signal processing operations on two-dimensional fields of data can be cast into a vector space formulation. This formulation often leads to a simplified analysis of two-dimensional processing systems and to the discovery of more efficient computational algorithms.

This paper considers the general representation of two-dimensional fields as vectors and discusses common types of operators on these data vectors. Particular emphasis is given to various forms of the superposition operation. Computation of superposition operations using unitary transform techniques is analyzed.

1. INTRODUCTION

Linear signal processing operations in one dimension such as convolution, transformation, and filtering can be succinctly expressed in vector space notation. It is also possible to extend this formulation to two-dimensional linear systems. The inherent advantages of doing so are a compact notation and the ability to use previously derived results for one-dimensional problems.

There are two basic means of expressing linear processing operations, series form and vector space form. As an illustration consider a one-dimensional example in which a data sequence $f(j)$ ($j = 1, 2, \ldots, J$) is mapped into a data sequence $p(k)$ ($k = 1, 2, \ldots, K$) by a linear processing rule. In a series formulation

$$p(k) = \sum_{j=1}^{J} f(j) W(k,j),$$

where $W(j,k)$ is a $K \times J$ array defining the operation. In a vector space formulation

$$p = Wf,$$}

where $p$ is a $K \times 1$ column vector containing the sequence elements of $p(k)$, $f$ is a $J \times 1$ column vector representing $f(j)$ and $W$ is a $K \times J$ matrix defining the operation.

\[ \text{Notation: uppercase boldface indicates a matrix; lowercase boldface indicates a vector; script letters denote transform domain vectors and matrices.} \]
Whether one adopts the series formulation or the vector space formulation for a particular problem is often a matter of taste and educational background. There are advantages and disadvantages to each formulation. Some advantages of the vector space formulation are as follows.

(a) Some linear operations which are cumbersome to express and manipulate in series form are simplified in vector space form. For example if a one-dimensional Hadamard transform is expressed in series form as in Eq. (1), the kernel $W(k,j)$ is a complicated function of the indices $(j,k)$ requiring modulo two arithmetic for manipulation, while the matrix operator $W$ is a Hadamard matrix containing only plus-or-minus-one terms [1].

(b) Some linear operations are only stated in vector space form. As an example, in Eq. (2) an estimate $\hat{f}$ of the vector $f$ can be obtained by premultiplying $p$ by the pseudoinverse $W^-$ of the matrix $W$ [2, p. 95].

(c) Periodicities and structures useful in developing computational algorithms are often more apparent in the vector space formulation. Many developers of fast computational algorithms for unitary transforms utilize matrix factorization techniques to eliminate computational redundancy [3].

(d) Many signal processing algorithms for signal enhancement, restoration, and detection are available in vector space form, and these algorithms may be utilized directly. For example, Eq. (2) is often employed as a model for physical signal degradation in which $p$ represents an observation, $f$ denotes an ideal signal to be estimated, and $W$ models a physical degradation process. A linear estimator of the form $\hat{f} = Tp$ is then found according to some error criterion. Solutions to this problem in vector form are available using techniques of regression, mean-square error estimation, and constrained optimization [2,4].

The major disadvantage of the vector space approach to signal processing operations is the converse of point (a) above: some linear operations which are cumbersome to express and manipulate in vector space form are simplified in series form. An example is the analysis to show that the fast Fourier transform algorithm can be utilized to produce the result of discrete convolution.

Clearly, the designer and analyst of signal processing systems should view the series and vector space formulations as alternate tools to be employed when the problem situation dictates. The purpose of this paper is to extend the vector space formulation to two-dimensional systems.

2. VECTOR IMAGE REPRESENTATION

Consider a data array $F(j,k)$ represented as an $N \times N$ matrix

$$ F = [F(j,k)]. $$

The extension to rectangular data arrays is straightforward. The data matrix can be cast into vector form through the use of an $N \times 1$ operational vector $v_n$ and an $N^2 \times N$ matrix $N_n$ defined as
Then, the vector representation of the data matrix $F$ is given by

$$f = \sum_{n=1}^{N} N_n F v_n.$$  \hspace{1cm} (3)

In essence the vector $v_n$ extracts the $n$th column from $F$ and the matrix $N_n$ places this column into the $n$th segment of the $N^2 \times 1$ vector $f$. Thus, $f$ contains the column scanned elements of $F$. The inverse relation of casting the vector $f$ into matrix form is obtained from

$$F = \sum_{n=1}^{N} N_n^T f v_n^T.$$ \hspace{1cm} (4)

With the matrix to vector operator of Eq. (3) and the vector to matrix operator of Eq. (4) it is now possible to easily convert between vector and matrix representations of a two-dimensional array. It should be noted that the vector matrix conversion equations represent more than a simple lexicographic ordering of data; the conversion equations contain linear operators, $N_n$ and $v_n$, which can be manipulated in an algebraic sense. Examples of manipulative procedures are abundant in subsequent sections.

3. LINEAR OPERATOR

Let $T$ denote an $M^2 \times N^2$ matrix performing a linear transformation on the $N^2 \times 1$ data vector $f$ yielding the $M^2 \times 1$ vector

$$p = T f,$$ \hspace{1cm} (5)

where $T$ is partitioned into $M \times N$ submatrices $T_{m,n}$

$$T = \begin{bmatrix}
T_{1,1} & T_{1,2} & \cdots & T_{1,N} \\
T_{2,1} & T_{2,2} & \cdots & T_{2,N} \\
\vdots & \vdots & \ddots & \vdots \\
T_{M,1} & T_{M,2} & \cdots & T_{M,N}
\end{bmatrix}.$$
From Eq. (3), the vector $p$ can be related to the data matrix $F$ by

$$p = \sum_{n=1}^{N} T N_n F v_n.$$  \hspace{1cm} (6)

In many signal processing applications the linear operator $T$ is highly structured, and computational simplifications are possible. Several special forms of the linear operator are defined below and illustrated in Fig. 1 for input and output arrays of the same dimension ($M = N$).

(a) Separable row and column processing of $F$:

$$T = T_C \otimes T_R = \begin{bmatrix} T_R(1, 1)T_C & T_R(1, 2)T_C & \cdots & T_R(1, N)T_C \\ T_R(2, 1)T_C & T_R(2, 2)T_C & \cdots & T_R(2, N)T_C \\ \vdots & \vdots & \ddots & \vdots \\ T_R(N, 1)T_C & T_R(N, 2)T_C & \cdots & T_R(N, N)T_C \end{bmatrix}.$$  

(b) Column processing of $F$:

$$T = \text{diag}[T_{C1}, T_{C2}, \ldots, T_{CN}],$$

where $T_{Cj}$ is the transformation matrix for the $j$th column.

(c) Identical column processing of $F$:

$$T = \text{diag}[T_C, T_C, \ldots, T_C]$$

or

$$T = T_C \otimes I_N.$$  

\footnote{The symbol $\otimes$ denotes the direct product of matrices. The definition employed here is the left direct product [2].}
(d) Row processing of F:
\[ T_{m,n} = \text{diag}[T_{R1}(m,n), T_{R2}(m,n), \ldots, T_{RN}(m,n)], \]
where \( T_{Rj} \) is the transformation matrix for the \( j \)th row.

(e) Identical row processing of F:
\[ T_{m,n} = \text{diag}[T_R(m,n), T_R(m,n), \ldots, T_R(m,n)] \]
or
\[ T = I_N \otimes T_R. \]

(f) Identical row and identical column processing of F:
\[ T = T_C \otimes I_N + I_N \otimes T_R. \]

Table I lists the number of computational operations for each of the above cases.

The special case of separable row and column processing of \( F \) warrants further discussion. Let the \( M \times M \) matrix \( P \) denote the matrix representation of \( p \). Then Eq. (4) leads to the result
\[
P = \sum_{m=1}^{M} M_m^T p u_m^T = \sum_{m=1}^{M} \sum_{n=1}^{N} M_m^T N_n F v_n u_m^T, \tag{7a}
\]
where the operators \( M_m \) and \( N_n \) simply extract the partition \( T_{m,n} \) from \( T \), leaving
\[
P = \sum_{m=1}^{M} \sum_{n=1}^{N} T_{m,n} F v_n u_m^T. \tag{7b}
\]
If the linear transformation is separable such that \( T \) may be expressed in direct product form then
\[ T_{m,n} = T_R(m,n)T_C. \]

As a consequence
\[
P = T_C F \sum_{m=1}^{M} \sum_{n=1}^{N} T_R(m,n) v_n u_m^T = T_C F T_R^T. \tag{8}
\]
Thus, separable two-dimensional linear transforms can be computed by sequential one-dimensional row and column computations on a data array. As indicated

**TABLE 1**

<table>
<thead>
<tr>
<th>Case</th>
<th>Operations (multiply and add)</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>( N^4 )</td>
</tr>
<tr>
<td>Column processing</td>
<td>( N^3 )</td>
</tr>
<tr>
<td>Row processing</td>
<td>( N^3 )</td>
</tr>
<tr>
<td>Row and column processing</td>
<td>( 2N^3 - N^2 )</td>
</tr>
<tr>
<td>Separable row and column processing matrix form</td>
<td>( 2N^3 )</td>
</tr>
</tbody>
</table>
by Table I, a considerable saving in computation is possible for such operators: computation by Eq. (5) in the general case requires \(M^2N^2\) operations; computation by Eq. (8), when it applies, requires only \(MN^2 + M^2N\) operations. Furthermore, \(F\) may be stored in a serial memory such as a disc or drum and fetched line-by-line, thereby obviating the requirement of storing \(F\) in a random access memory that is usually more expensive than a serial memory. With this technique, however, it is necessary to transpose the result of the column transforms in order to perform the row transforms.

The computational simplifications indicated in Table I arise from the partitioning of \(T\) into submatrices that are sparse. In many other computational situations it is possible to factor \(T\) into a product of sparse matrices. Unfortunately, there is no systematic means of determining if an arbitrary linear operator possesses structure that leads to a sparse factorization, which in turn can be utilized to reduce computational requirements \([3]\).

4. STATISTICAL CHARACTERIZATION

In many discrete signal processing operations it is convenient to regard a data vector, \(f\), as a sample of a vector random process. The mean value of \(f\) can be related to the mean value of the elements of \(F\) by

\[
\eta_f = E\{f\} = \sum_{n=1}^{N} N_n E\{F\} v_n, \tag{9}
\]

where

\[
E\{F\} = [E\{F(j,k)\}].
\]

Similarly, the correlation matrix of \(f\), which is dependent upon the correlation of elements of \(F\), is given by

\[
R_f = E\{f f^T\} = E\left\{ \left[ \sum_{m=1}^{N} N_m F v_m \sum_{n=1}^{N} v_n^T F^* T N_n \right] \right\} \tag{10a}
\]

or

\[
R_f = \sum_{m=1}^{N} \sum_{n=1}^{N} N_m E\{F v_m v_n^T F^* T \} N_n^T = \sum_{m=1}^{N} \sum_{n=1}^{N} N_m R_{m,n} N_n^T, \tag{10b}
\]

where \(R_{m,n}\) is the correlation matrix of the \(m\)th and \(n\)th columns of \(F\). Thus, it is possible to express \(R_f\) in partitioned form as the \(N^2 \times N^2\) matrix

\[
R_f = \begin{bmatrix}
R_{1,1} & R_{1,2} & \cdots & R_{1,N} \\
R_{2,1} & R_{2,2} & \cdots & R_{2,N} \\
\vdots & \vdots & \ddots & \vdots \\
R_{N,1} & R_{N,2} & \cdots & R_{N,N}
\end{bmatrix}. \quad \tag{11}
\]

\(E\{\cdot\}\) denotes the expected values operation.
The covariance matrix of \( \mathbf{f} \) is defined to be

\[
\mathbf{K}_f = \mathbf{R}_f - \eta_f \eta_f^T.
\]  

(12)

If the data array is wide sense stationary then

\[
\mathbf{R}_{m,n} = \mathbf{R}_{k}, \quad m \geq n,
\]

\[
\mathbf{R}_{m,n} = \mathbf{R}_{k}^*, \quad m < n,
\]

where \( k = |m - n| + 1 \). For a wide stationary data array the correlation matrix

\[
\mathbf{R}_f = \begin{bmatrix}
\mathbf{R}_1 & \mathbf{R}_2 & \mathbf{R}_3 & \cdots & \mathbf{R}_N \\
\mathbf{R}_2^* & \mathbf{R}_1 & \mathbf{R}_3 & \cdots & \mathbf{R}_{N-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{R}_N^* & \mathbf{R}_{N-1}^* & \mathbf{R}_{N-2}^* & \cdots & \mathbf{R}_1
\end{bmatrix}
\]  

(13)

is of block Toeplitz form \([5]\). When the correlation between elements of \( \mathbf{F} \) is separable into the product of row and column correlation functions, then the correlation matrix of \( \mathbf{f} \) can be written in direct product form as

\[
\mathbf{R}_f = \mathbf{R}_C \otimes \mathbf{R}_R,
\]

where \( \mathbf{R}_R \) and \( \mathbf{R}_C \) denote the \( N \times N \) correlation matrices of the rows and columns of \( \mathbf{F} \), respectively.

Finally, if a linear transformation \( \mathbf{p} = \mathbf{Tf} \) is performed on the data vector \( \mathbf{f} \) it is simple to show that the mean of the output vector is

\[
\eta_p = \mathbf{T} \eta_f
\]  

(14)

and its correlation matrix is

\[
\mathbf{R}_p = \mathbf{T} \mathbf{R}_f \mathbf{T}^*.
\]  

(15)

5. UNITARY TRANSFORM OPERATOR

A special class of linear operators of wide use are the unitary transforms for which the transformation matrix \( \mathbf{A} \) has the property \([6-9]\)

\[
\mathbf{A}^{-1} = \mathbf{A}^*.
\]  

(16)

The unitary transform of the \( N^2 \times 1 \) data vector \( \mathbf{f} \) is defined as

\[
\mathbf{j} = \mathbf{Af},
\]  

(17)

where \( \mathbf{j} \) is an \( N^2 \times 1 \) vector and \( \mathbf{A} \) is an \( N^2 \times N^2 \) matrix.\(^4\) If \( \mathbf{A} \) is separable into direct product form

\[
\mathbf{A} = \mathbf{A}_C \otimes \mathbf{A}_R,
\]  

(18)

then by Eq. (8) the matrix form of \( \mathbf{j} \) may be computed as sequential row and column one-dimensional transforms on the data matrix \( \mathbf{F} \). Thus

\[
\mathbf{j} = \mathbf{A}_C \mathbf{F} \mathbf{A}_R^T.
\]  

(19)

\(^4\) Overbars indicate transform vectors and matrices.
The Fourier, Hadamard, and Haar transforms are separable types of transformations.

The Karhunen-Loeve transform, widely employed in one-dimensional signal analysis, can be extended to two dimensions by considering the two-dimensional data to be in vector form. The K-L transformation matrix is then composed of the eigenvectors of the covariance matrix $K_r$ satisfying

$$AK_r = \Lambda A,$$ (20)

where $\Lambda$ is a diagonal matrix of the eigenvalues of $K_r$,

$$\Lambda = \begin{bmatrix} \lambda(1) & 0 \\ \lambda(2) & \ddots \\ 0 & \ddots & \ddots \\ \end{bmatrix},$$

and the corresponding eigenvectors of $K_r$ are arranged as rows of $A$. If $K_r$ is orthogonally separable, then the transform is also separable, and is of the same form as Eq. (19), where

$$A_c K_c = A_c \Lambda_c A_c,$$ (21a)

$$A_R K_R = A_R \Lambda_R A_R,$$ (21b)

and $\lambda(k) = \lambda_{R(i)} \lambda_{C(j)}$ for $i,j = 1,2, \ldots, N^{[10]}$.

If the unitary transform is computed by a conventional matrix multiplication procedure, $N^4$ operations are required for the vector form, but only $2N^3$ operations for the matrix form if the transform is separable. Both the vector and matrix forms require $2N^2 \log_2 N$ operations using a fast computational algorithm such as the fast Fourier transform. The Haar and related unitary transforms require on the order of $2N^2$ operations. However, there are many unitary transforms such as the Karhunen-Loeve transform that do not possess a fast algorithm.

6. SUPERPOSITION OPERATOR

The discrete two-dimensional superposition operator has been defined in the literature in two basic ways: as a discrete linear filtering process to be performed on a data array; and as a discrete model of a continuous filtering process. The two definitions are usually not equivalent. Both definitions are considered in the following. A discrete circular superposition operator is also defined, and relations are given between the three operators.

Finite Area Superposition Operator

Consider first the discrete superposition of a spatially truncated data array

$$F(n_1,n_2) \quad n_1,n_2 = 1,2, \ldots, N$$

with a spatially truncated impulse response operator

$$H(l_1,l_2;m_1,m_2) \quad l_1,l_2 = 1,2, \ldots, L.$$
In the general case the impulse response array changes form for each point \((m_1,m_2)\) in the processed array, \(F\). Then, the finite area superposition operation is defined as

\[
Q(m_1,m_2) = \sum_{n_1=1}^{m_1} \sum_{n_2=1}^{m_2} F(n_1,n_2) H(m_1 - n_1 + 1,m_2 - n_2 + 1;m_1,m_2)
\] (22)

for \(m_1,m_2 = 1,2, \ldots ,M\), where \(H\) and \(F\) are assumed to be zero outside their range of indices. Examination of the indices of the impulse response array at its extremal positions indicates that \(M = N + L - 1\), and hence, the processed array, \(Q\), is of larger dimension than the data array, \(F\). Figure 2a illustrates the geometry of finite area superposition.

If the arrays \(F\) and \(Q\) are represented in vector form by the \(N^2 \times 1\) vector \(f\) and the \(M^2 \times 1\) vector \(q\) respectively, then the finite area superposition operation can be written as

\[
q = Df,
\] (23)

where \(D\) is an \(M^2 \times N^2\) matrix containing the elements of the impulse response. It is convenient to partition the superposition operator matrix \(D\) into submatrices of dimension \(M \times N\). Observing the summation limits of Eq. (22), one sees that

\[
D = \begin{bmatrix}
D_{1,1} & 0 & 0 & \cdots & 0 \\
D_{2,1} & D_{2,2} & 0 & \cdots & 0 \\
& \ddots & \ddots & \ddots & \ddots \\
0 & \cdots & D_{L,2} & D_{L-1,2} & 0 \\
0 & \cdots & 0 & D_{N,N} & \cdot \\
\end{bmatrix}
\] (24)
The general nonzero term of \( D \) is then given by
\[
D_{m_1,n_1}(m_1,n_1) = H(m_1 - n_1 + 1, m_2 - n_2 + 1; m_1, m_2),
\]
where
\[
1 \leq n_1 \leq N, \quad 1 \leq n_2 \leq N,
\]
\[
1 \leq m_1 \leq n_1 + L - 1, \quad n_2 \leq m_2 \leq n_2 + L - 1.
\]
Thus, it is observed that \( D \) is highly structured and quite sparse, with the center band of submatrices containing stripes of zero elements.

If the impulse response is position invariant then the structure of \( D \) does not explicitly depend upon the output array coordinate \((m_1, m_2)\). Also,
\[
D_{m_1,n_1} = D_{m_1+1,n_1+1}.
\]
As a result, the columns of \( D \) are shifted versions of the first column. Under these conditions the finite area superposition operator is known as the finite area convolution operator. Figure 3b contains a computer printout of the finite area convolution operator for a \( 2 \times 2 \) \((N=2)\) data array, a \( 4 \times 4 \) \((M=4)\) filtered data array, and a \( 3 \times 3 \) \((L=3)\) impulse response array. The integer pairs \((i,j)\) at each element of \( D \) represent the element \((i,j)\) of \( H(i,j) \). The basic structure of \( D \) can be seen more clearly in the larger-size matrix depicted in the photograph of Fig. 4a. In this example \( M = 16, N = 8, L = 9 \), and the impulse response has a symmetrical Gaussian shape. Note that \( D \) is a \( 256 \times 64 \) matrix in this example.

Following the same technique leading to Eq. (7), one may write the matrix...
FIG. 4. Examples of spatial and Fourier domain convolution operators.

(a) spatial domain finite area convolution operator

(c) spatial domain sampled infinite area convolution operator

(e) spatial domain circular area convolution operator

(b) Fourier domain

(d) Fourier domain

(f) Fourier domain
form of the superposition operation as

$$Q = \sum_{m=1}^{M} \sum_{n=1}^{N} D_{m,n} F v_{m,n} r.$$  \hfill (27)

If the impulse response is spatially invariant, and is of separable form such that

$$H = h_c h_r^T,$$  \hfill (28)

where $h_r$ and $h_c$ are column vectors representing row and column impulse responses, respectively, then

$$D = D_c \otimes D_r.$$  \hfill (29)

The matrices $D_r$ and $D_c$ are $M \times N$ matrices of the form

$$D_r = \begin{bmatrix}
  h_r(1) & 0 & \cdots & 0 \\
  h_r(2) & h_r(1) & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  h_r(L) & h_r(L-1) & \cdots & h_r(1) \\
  0 & \cdots & \cdots & 0 \\
  \vdots & \vdots & \vdots & \vdots \\
  0 & \cdots & \cdots & 0 \\
  \vdots & \vdots & \vdots & \vdots \\
  0 & \cdots & \cdots & h_r(L)
\end{bmatrix}.$$  

The two-dimensional convolution operation may then be computed by sequential row and column one-dimensional convolutions. Thus,

$$Q = D_c F D_r^T.$$  \hfill (30)

In vector form the finite area superposition or convolution operator requires $N^2 L^2$ operations if the zero multiplications of $D$ are avoided. The separable operator of Eq. (30) can be computed with only $NL(M + N)$ operations.

**Sampled Infinite Area Superposition Operator**

Two-dimensional linear filtering operations of continuous functions may be described by the superposition integral

$$G(x,y) = \int_{-\infty}^{\infty} F(\alpha,\beta) j(x,y;\alpha,\beta) d\alpha d\beta,$$  \hfill (31)

where $j(x,y;\alpha,\beta)$ represents the impulse response of a linear, but possibly spatially variant, system. In the discrete representation of the superposition integral, the continuous processed function will be described by samples, spaced evenly over a grid $\Delta x$, $\Delta y$. For notational simplicity, the continuous data function $F(\alpha,\beta)$ and the continuous impulse response $j(x,y;\alpha,\beta)$ will also be assumed to be sampled over the same grid spacing. At this stage, it is possible to rewrite the superposition integral as a double summation over infinite limits by invoking the
sampling theorem, or by using an integration formula. In order to cast the infinite
area superposition operator into vector form, it is obviously necessary to truncate
the impulse response to some spatial limit, say \((L\Delta x, L\Delta y)\), and to restrict
the description of \(G(x,y)\) to some area, say \((M\Delta x, M\Delta y)\). Then, the truncated
superposition operator can be written as

\[
G(m_1,m_2) = \sum_{n_1=m_1}^{L+m_1-1} \sum_{n_2=m_2}^{L+m_2-1} F(n_1,n_2) H(m_1-n_1+L,m_2-n_2+L;m_1,m_2),
\]

where the array \(H\), assumed to be zero outside its range of indices, represents
the sampled impulse response function and incorporates all quadrature repre-
sentation factors. For simplicity, the grid spacing \(\Delta x, \Delta y\) is dropped. In order to
prevent serious approximation errors at the boundaries of \(G\), \(N\) should be
chosen such that

\[
N \geq M + L - 1.
\]

Thus, in contrast to the finite area superposition operation, the processed array
is of smaller dimension than the data array. The effect of violating the inequality
of Eq. (33) will be considered later. Figure 2b illustrates an infinite data array
and a superimposed impulse response array. From this figure it is observed that
elements outside the dimension of the processed array, \(G\), contribute to it.

If the arrays \(F\) and \(G\) are represented in vector form by \(f\) and \(g\), respectively,
then the superposition operator can be written as

\[
g = Bf,
\]

where \(B\) is an \(M^2 \times N^2\) matrix which can be partitioned as

\[
B = \begin{bmatrix}
B_{1,1} & B_{1,2} & \cdots & B_{1,L} & 0 & 0 & \cdots & 0 \\
0 & B_{2,2} & \cdots & B_{2,L} & B_{2,L+1} & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots & \ddots & \ddots & \vdots \\
0 & \cdots & \cdots & \cdots & 0 & \cdots & B_{M,M} & B_{M,M+1}
\end{bmatrix}
\]

The general term of \(B\) is then given by

\[
B_{m_1,n_1}(m_1,n_1) = H(m_1-n_1+L,m_2-n_2+L;m_1,m_2)
\]

for

\[
1 \leq m_1 \leq M, \quad 1 \leq m_2 \leq M, \\
m_1 \leq n_1 \leq L + m_1 - 1, \quad m_2 \leq n_2 \leq L + m_2 - 1.
\]

If the impulse response is position invariant, the structure of \(B\) is not explicitly
a function of \((m_1,m_2)\). In addition,

\[
B_{m_1,n_1} = B_{m_1+n_1, m_2+n_2+1}.
\]

Consequently, the rows of \(B\) are shifted versions of the first row. The operator \(B\)
then becomes a sampled infinite area convolution operator. Figure 3c contains a computer printout of the sampled infinite area convolution operator for a $4 \times 4$ ($N = 4$) data array, a $2 \times 2$ ($M = 2$) filtered data array, and a $3 \times 3$ ($L = 3$) impulse response array. An extension to larger dimension is shown in Fig. 4c for $M = 8$, $N = 16$, $L = 9$, and a Gaussian shaped impulse response.

When the impulse response is spatially invariant and orthogonally separable as in Eq. (28) then

$$B = B_c \otimes B_R,$$  \hspace{1cm} (37)

where $B_c$ and $B_R$ are $M \times N$ matrices of the form

$$B_R = \begin{bmatrix}
  h_R(L) & h_R(L-1) & \cdots & h_R(1) & 0 & \cdots & 0 \\
  0 & h_R(L) & \cdots & h_R(2) & h_R(1) & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & 0 & h_R(L) & \cdots & h_R(1)
\end{bmatrix},$$  \hspace{1cm} (38)

The two-dimensional convolution operation then reduces to sequential row and column convolutions of the matrix form of the data array. Thus,

$$G = B_c F B_R^T.$$  \hspace{1cm} (39)

The superposition or convolution operator expressed in vector form requires $M^2 L^2$ operations if the zero multiplications of $B$ are avoided. A separable convolution operator can be computed in matrix form with only $ML(M + N)$ operations.

Suppose that both the finite area and the infinite area superposition operations are performed on an $N \times N$ data array for the same $L \times L$ impulse response array with the intention of modeling a continuous superposition process. Then, the processed array for the finite area computation will be equivalent to the processed array for the infinite area computation, surrounded by a boundary of $(L - 1)$ superfluous elements. Conversely, if the processed array size is held common for the two superposition operators, the $L - 1$ boundary elements for the array obtained by the finite area superposition operator will be in error. Therefore, care must be taken in the application of the finite area superposition operator to model continuous superposition processes.

**Circular Superposition Operator [11]**

In circular superposition the input data, the processed output, and the impulse response arrays are all spatially periodic over some common period. In order to unify the presentation, these arrays will be defined in terms of the spatially limited arrays considered previously. First, let the $N \times N$ data array $F(n_1, n_2)$ be

\footnote{An exception occurs when the data array contains a boundary of $(L - 1)/2$ zero elements. A practical example is a blurred photograph of the moon against the night sky.}
imbedded in the upper left corner of a \( J \times J \) array \( (J \geq N) \) of zeros giving
\[
F_E(n_1, n_2) = F(n_1, n_2), \quad 1 \leq n_1 \leq N; \quad (40a)
\]
\[
F_E(n_1, n_2) = 0, \quad N + 1 \leq n_i < J. \quad (40b)
\]

In a similar manner an extended impulse response array is created by imbedding the spatially limited impulse array in a \( J \times J \) matrix of zeros. Thus, let
\[
H_E(l_1, l_2; m_1, m_2) = H(l_1, l_2; m_1, m_2), \quad 1 \leq l_i \leq L; \quad (41a)
\]
\[
H_E(l_1, l_2; m_1, m_2) = 0, \quad L + 1 \leq l_i < J. \quad (41b)
\]

Periodic arrays \( F_p(n_1, n_2) \) and \( H_p(l_1, l_2; m_1, m_2) \) are now formed by replicating the extended arrays over the spatial period \( J \). Then, the circular convolution of these functions is defined as
\[
K_E(m_1, m_2) = \sum_{n_1=1}^{J} \sum_{n_2=1}^{J} F_p(n_1, n_2) H_p(m_1 - n_1 + 1, m_2 - n_2 + 1; m_1, m_2). \quad (42)
\]

Similarity of this equation with Eq. (22) describing finite area superposition is evident. In fact, if \( J \) is chosen such that \( J \geq N + L - 1 \), the terms \( K_E(m_1, m_2) = Q(m_1, m_2) \) for \( 1 \leq m_i \leq M \). The similarity of the circular superposition operation and the sampled infinite area superposition operation of Eq. (32) should also be noted. These relations become clearer in the vector space representation of the circular superposition operation.

Let the arrays \( F_E \) and \( K_E \) be expressed in vector form as the \( J^2 \times 1 \) vectors \( f_E \) and \( k_E \), respectively. Then the circular superposition operator can be written as
\[
k_E = C f_E, \quad (43)
\]

where \( C \) is a \( J^2 \times J^2 \) matrix containing elements of the array \( H_p \). The circular superposition operator can then be conveniently expressed in terms of \( J \times J \) submatrices \( C_{m,n} \) as given by
\[
C = \begin{bmatrix}
C_{1,1} & 0 & 0 & \cdots & 0 & C_{1,J-L+2} & \cdots & C_{1,J} \\
C_{2,1} & C_{2,2} & 0 & \cdots & 0 & \cdots & \cdots & C_{2,J} \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\
C_{L,1} & C_{L,2} & \cdots & \cdots & \cdots & \cdots & \cdots & C_{L,J-L+2} \\
0 & C_{L+1,2} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & \cdots & 0 & \cdots & \cdots & \cdots & C_{J,J-L+1} \\
0 & 0 & \cdots & 0 & \cdots & \cdots & \cdots & C_{J,J} \\
\end{bmatrix} \quad (44)
\]

where
\[
C_{m_2,n_2}(m_1, n_1) = H_E(k_1, k_2; m_1, m_2)
\]
for

\[ 1 \leq n_1 \leq J, \quad 1 \leq n_2 \leq J, \]
\[ 1 \leq m_1 \leq J, \quad 1 \leq m_2 \leq J, \]

with

\[ k_1 = (m_1 - n_1 + 1) \mod(J), \]
\[ k_2 = (m_2 - n_2 + 1) \mod(J), \]
\[ H_E(0,0) \equiv 0. \]

It should be noted that each row and column of \( C \) contains \( L \) nonzero submatrices. If the impulse response array is spatially invariant, then

\[ C_{m_2,n_2} = C_{m_2+1,n_2+1} \]

and the submatrices of the rows (columns) can be obtained by a circular shift of the first row (column). In this instance the matrix \( C \) is a block circulant matrix. Figure 3d illustrates the circular area convolution operator for a \( 16 \times 16 (J = 4) \) data and filtered data array and for a \( 3 \times 3 (L = 3) \) impulse response array. In Fig. 4e the operator is shown for \( J = 16 \) and \( L = 9 \) with a Gaussian shaped impulse response.

Finally, when the impulse response is spatially invariant and orthogonally separable then

\[ C = C_C \otimes C_R, \quad (45) \]

where \( C_C \) and \( C_R \) are \( J \times J \) matrices of the form

\[ C_R = \begin{bmatrix}
  h_R(1) & 0 & \cdots & 0 & h_R(L) & \cdots & h_R(2) \\
  h_R(2) & h_R(1) & \cdots & 0 & 0 & \cdots & h_R(3) \\
  \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
  h_R(L-1) & h_R(L) & \cdots & 0 & 0 & \cdots & h_R(L) \\
  h_R(L) & h_R(L-1) & \cdots & \cdots & \cdots & \cdots & 0 \\
  0 & h_R(L) & 0 & \cdots & \cdots & \cdots & h_R(1) \\
  \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \ddots \\
  0 & \cdots & \cdots & 0 & h_R(L) & \cdots & \cdots \\
\end{bmatrix} \quad (46) \]

Two-dimensional circular convolution may then be computed as

\[ K_R = C_C F_E C_R^T. \quad (47) \]

**Relationship of Superposition Operators**

The elements of the finite area superposition operator, \( D \), and the elements of the sampled infinite area superposition operator, \( B \), can be extracted from the circular superposition operator, \( C \), by use of selection matrices of the first and
second kind defined as

\[
S_1^{(j)} = \begin{bmatrix}
I_K & 0 \\
0 & 0
\end{bmatrix}
\]

\[
S_2^{(j)} = \begin{bmatrix}
0 & I_K & 0 \\
L-1 & 0 & J-K-L+1
\end{bmatrix}
\]

where \(I_K\) is a \(K \times K\) identity matrix. Then, examination of the structure of the various superposition operators indicates that

\[
D = [S_1^{(j)}(M) \circledast S_1^{(j)}(M)]c [S_1^{(j)}(N) \circledast S_1^{(j)}(N)] r, M > N, (50a)
\]

\[
B = [S_2^{(j)}(M) \circledast S_2^{(j)}(M)]c [S_1^{(j)}(N) \circledast S_2^{(j)}(M)] r, N > M. (50b)
\]

That is, the matrix \(D\) is obtained by extracting the first \(M\) rows and \(N\) columns of submatrices \(C_{m,n}\) of \(C\). The first \(M\) rows and \(N\) columns of each submatrix are also extracted. A similar explanation holds for the extraction of \(B\) from \(C\). In Fig. 3 the elements of \(C\) to be extracted to form \(D\) and \(B\) are indicated by shading.

From the definition of the extended input data array of Eq. (40) it is obvious that the input data vector \(f\) can be obtained from the extended data vector, \(f_E\), by the selection operation

\[
f = [S_1^{(j)}(M) \circledast S_1^{(j)}(M)]f_E. (51)
\]

It can also be shown that the output vector for finite area superposition can be obtained from the output vector for circular superposition by the selection operation

\[
q = [S_1^{(j)}(M) \circledast S_1^{(j)}(M)]k_E. (52a)
\]

Similarly, for sampled infinite area superposition

\[
g = [S_2^{(j)}(M) \circledast S_2^{(j)}(M)]k_E. (52b)
\]

Expressing both \(q\) and \(k_E\) of Eq. (50a) in matrix form, one obtains

\[
Q = \sum_{m=1}^{M} \sum_{n=1}^{J} M_{m,n}^T [S_1^{(j)}(M) \circledast S_1^{(j)}(M)] N_{m,n} K_{m,n} u_{m,n}^T.
\]

As a result of the separability of the selection operator, Eq. (51) reduces to

\[
Q = [S_1^{(j)}(M)]k_E [S_1^{(j)}(M)]^T. (53a)
\]

Similarly, for Eq. (50b) describing sampled infinite area superposition

\[
G = [S_2^{(j)}(M)]k_E [S_2^{(j)}(M)]^T. (53b)
\]
Figure 5 illustrates the locations of the elements of Q and G extracted from $K_E$ for finite area and sampled infinite area superposition.

In summary, it has been shown that the output data vectors for either finite area of sampled infinite area superposition can be obtained by a simple selection operation on the output data vector of circular superposition. Computational advantages that can be realized from this result are considered next.

**Fourier Transform Convolution**

Convolution operations can often be performed more efficiently by Fourier transform processing than by direct processing in the spatial domain [12,13]. With transform processing, as illustrated in Fig. 6, a Fourier transformation is performed on the data vector $f$ by multiplication by the matrix

$$A_{y^2} = A_x \otimes A_y,$$

where

$$A_x = \left[ \frac{1}{N} \right]^{1/2} \exp \left( -\frac{2\pi i(x-u)}{N} \right), \quad x,y = 0,1,2, \ldots ,N.$$

Next, the Fourier domain data vector $f$ is multiplied by the appropriate Fourier domain convolution operator, $\mathcal{O}$, $\mathcal{S}$, or $\mathcal{C}$. An inverse Fourier transform reconstructs the output vector. For finite area convolution, since

$$q = Df$$

and

$$q = (A_{y^2})^{-1} \mathcal{O} (A_{y^2})f,$$
then clearly

$$\overline{B} = (A_M)^D(A_{N^2})^{-1}. \quad (56a)$$

Similarly for sampled infinite area and circular convolution, the Fourier transform operators are

$$\overline{B} = (A_M)^B(A_{N^2})^{-1}, \quad (56b)$$

$$\overline{C} = (A_M)^C(A_{N^2})^{-1}. \quad (56c)$$

In all cases the Fourier domain convolution operator can be obtained by a separable two-dimensional transformation of the corresponding spatial domain operator. The spatial convolution operators D, B, and C all exhibit a shift structure between rows or columns. Since the Fourier transform of a shifted sequence is equal to the Fourier transform of the sequence multiplied by a phase factor proportional to the shift, it is not surprising that the Fourier domain convolution operators exhibit considerable structure. It can be shown that

$$\overline{D} = \mathcal{F}_M[P_D \otimes P_D] \quad \text{for} \quad M = N + L - 1, \quad (57a)$$

$$\overline{B} = [P_B \otimes P_B] \mathcal{F}_N \quad \text{for} \quad N = M - L - 1, \quad (57b)$$
where
\[ P_{B}(u,v) = \frac{1}{M^{1/2}} \left( 1 - \mathcal{W}_{N}^{-(v-1)(L-1)}/(1 - \mathcal{W}_{N}^{(v-1)(L-1)}) \right) \].

The matrix \( \mathcal{K} \) is a diagonal matrix obtained by performing a two-dimensional Fourier transform on the \( K \times K \) extended impulse response matrix of Eq. (41), which gives
\[ \mathcal{K} = A_{K} \mathcal{H}_{K} A_{K}. \]

These transform components are then column scanned and inserted as the diagonal elements of the \( K^2 \times K^2 \) matrix
\[ \mathcal{H}_{K} = \text{diag}\{ \mathcal{K}(1,1), \mathcal{K}(2,1), \ldots, \mathcal{K}(K,K) \} \].

The \( \mathcal{D} \) and \( \mathcal{H} \) Fourier domain convolution operators each consist of a scalar weighting matrix \( \mathcal{H}_{E} \) and a matrix \( \mathcal{P} \otimes \mathcal{P} \) that performs the dimensionality conversion (an interpolation) between the \( N^2 \) element input vector and the \( M^2 \) element output vector. Generally, the dimensionality matrix is quite sparse and computational simplifications can be realized [13]. Figures 4b and 4d illustrate the structure of \( \mathcal{D} \) and \( \mathcal{H} \) for the examples of Figs. 4a and 4c, respectively. The Fourier domain circular convolution operator \( \mathcal{C} \) is a diagonal matrix since the circular convolution operator \( C \) is a block circulant matrix, and as indicated by Eq. (56c), the Fourier transform basis vectors are eigenvectors of \( C \) [15]. Figure 4f contains a representation of \( \mathcal{D} \) for the example of Fig. 4e.

As noted previously, the equivalent output vector for either finite area or sampled infinite area convolution can be obtained by an element selection operation on the extended output vector \( k_{E} \) for circular convolution or its matrix counterpart \( K_{E} \). This result combined with Eq. (57c) leads to a particularly efficient means of convolution computation indicated by the following steps.

(a) Imbed the impulse response matrix in the upper left corner of an all-zero \( J \times J \) matrix \((J > M + L - 1 \text{ for finite area convolution or } J > N + L - 1 \text{ for sampled infinite area convolution})\) and take the two-dimensional Fourier transform of the extended impulse response matrix
\[ \mathcal{F}_{E} = A_{J} I_{E} A_{J}; \]

(b) Imbed the input data array in the upper left corner of an all-zero \( J \times J \) matrix and take the two-dimensional Fourier transform of the extended input data matrix
\[ \mathcal{F}_{B} = A_{J} F_{B} A_{J}; \]

(c) Perform the scalar multiplication
\[ \mathcal{F}_{E}(m,n) = J_{\mathcal{F}_{E}}(m,n) \mathcal{F}_{B}(m,n), \]

where \( 1 \leq m \leq J \) and \( 1 \leq n \leq J \):

(d) Take the inverse Fourier transform of \( \mathcal{F}_{E} \),
VECTOR FORMULATION OF 2D SIGNAL PROCESSING

\[ K_B = (A_J)^{-1} \mathcal{F}_E (A_J)^{-1}; \]  

(e) extract the desired output matrix

\[ Q = [S1_j^{(M)}] K_B [S1_j^{(M)}]^T \]  

or

\[ G = [S2_j^{(M)}] K_B [S2_j^{(M)}]^T. \]

It is important that the size of the extended arrays in steps (a) and (b) be chosen large enough to satisfy the indicated inequalities. If steps (a) to (e) are performed with \( J = M = N \), the resulting output array will contain erroneous terms in a boundary region of width \( L - 1 \) elements, as indicated in Fig. 5. This is the so-called wraparound error associated with the incorrect usage of the Fourier domain convolution method.

In many signal processing applications the same impulse response operator is used on different data, and hence, step (a) need not be repeated. The filter matrix \( \mathcal{K}_E \) may be either stored functionally or indirectly as a computational algorithm. Using a fast Fourier transform algorithm the forward and inverse transforms require on the order of \( 2J^2 \log_2 J \) operations each. The scalar multiplication requires \( J^2 \) operations, in general, for a total of \( J^2 \left[ 1 + 4 \log_2 J \right] \) operations. For an \( N \times N \) input array, an \( M \times M \) output array, and an \( L \times L \) impulse response array, finite area convolution requires \( N^2L^2 \) operations and sampled infinite area convolution requires \( M^2L^2 \) operations. If the dimension of the impulse response, \( L \), is sufficiently large with respect to the dimension of the input array, \( N \), Fourier domain convolution will be more efficient than direct convolution, perhaps by an order of magnitude or more. Figure 7 provides a plot of \( L \) versus \( N \) for equality between direct and Fourier domain finite area convolution. The jaggedness of the plot arises from discrete changes in \( J \) (64, 128, 256, ...) as \( N \) increases.

7. INVERSE OPERATOR

A common task in linear signal processing is to "invert" the transformation equation

\[ p = T f \]  

\[ \text{FIG. 7. Comparison of direct and Fourier domain processing for finite area convolution.} \]
to obtain the value of the input data vector \( f \), or some estimate \( \hat{f} \) of the data vector, in terms of the output vector \( p \). If \( T \) is a square matrix, obviously

\[
f = (T)^{-1}p
\]

provided that the matrix inverse exists. If \( T \) is not square, the generalized inverse, \( T^{-} \), can be used to compute solutions to Eq. (66) if they exist, and to compute minimum mean-square error, minimum norm estimates otherwise \[2\]. In either case the estimate is

\[
\hat{f} = T^{-}p,
\]

where the generalized inverse satisfies the following relations.

(a) \( TT^{-} = (TT^{-})^{T} \),
(b) \( T^{-}T = (T^{-}T)^{T} \),
(c) \( TT^{-}T = T \),
(d) \( T^{-TT^{-}} = T^{-} \).

If \( T \) is of rank \( M^{2} \), then the generalized inverse is equal to

\[
T^{-} = T^{T}(TT^{T})^{-1}
\]

and if \( T \) is of rank \( N^{2} \) then

\[
T^{-} = (T^{T}T)^{-1}T^{T}.
\]

A special case of the generalized inverse operator of computational interest occurs when \( T \) is direct product separable. Under this condition

\[
T^{-} = T_{c}^{-} \otimes T_{r}^{-},
\]

where \( T_{c}^{-} \) and \( T_{r}^{-} \) are the generalized inverses of the row and column linear operators.

The generalized inverse operator may be used for inversion of the superposition operators defined in Section 6. For finite area superposition the superposition operator is of maximum rank \( N^{2} \), and the generalized inverse is

\[
D^{-} = (D^{T}D)^{-1}D^{T}.
\]

For sampled infinite area superposition, the rank of the superposition operator is no greater than \( M^{2} \) and the corresponding generalized inverse becomes

\[
B^{-} = B^{T}(BB^{T})^{-1}.
\]

Finally, for circular superposition, the superposition operator is of dimension \( J^{2} \times J^{2} \) and

\[
C^{-} = (C^{T}C)^{-1}C^{T} = C^{-1}.
\]

For circular convolution, the matrix \( C \) is block circulant and its inverse is also block circulant \[12\].

Fourier domain processing can be applied to advantage in generalized inverse computations. For example, it can be easily shown that the Fourier domain gen-
eralized inverses for superposition operations can be computed by
\[
\mathbf{\mathcal{G}}^{-} = (\mathbf{\mathcal{G}}^* \mathbf{\mathcal{G}})^{-1} \mathbf{\mathcal{G}}^* c, \quad (72a)
\]
\[
\mathbf{\mathcal{G}}^{-} = \mathbf{\mathcal{G}}^* (\mathbf{\mathcal{G}}\mathbf{\mathcal{G}}^*)^{-1}, \quad (72b)
\]
\[
\mathbf{\mathcal{G}}^{-} = \mathbf{\mathcal{G}}^{-1}. \quad (72c)
\]
For convolution operations \(\mathbf{\mathcal{G}}^{-}\) and \(\mathbf{\mathcal{G}}^{-}\) have been found to be quite sparse \[16\]. The matrix \(\mathbf{\mathcal{G}}\) is of diagonal form for circular convolution since \(\mathbf{\mathcal{G}}\) is also diagonal. This latter result leads to an efficient algorithm for obtaining the generalized inverse of a finite area convolutional operation \[16\].

8. SUMMARY

The vector formulation presented here has proved useful in the analysis of two-dimensional processing systems and in the organizational design of such systems. It has been pointed out that various superposition formulas exist, and relationships between the formulations can be developed. Finally, it has been shown that two-dimensional convolution operations can often be performed more efficiently through the use of unitary transformations.

ACKNOWLEDGMENT

This work was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Air Force Eastern Test Range under Contract No. F08606-72-C-0008.

REFERENCES