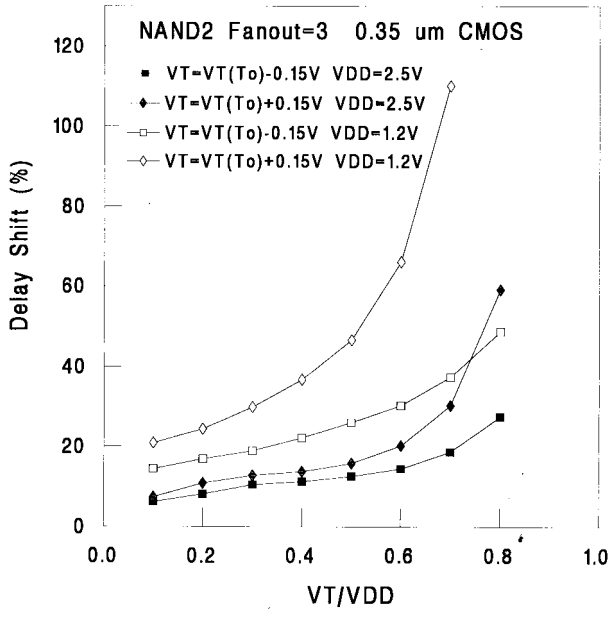

Fig. 4. Delay shift versus V_T/V_{DD} at 27°C for 2-input NAND with fanout = 3, 0.35 μm CMOS and $\Delta V_T = \pm 0.15$ V.

IV. Conclusion

As a result of our study outlined herein, a CMOS supply voltage of around 1 V is expected to be widely accepted as a standard in the design of the next generation bulk-CMOS and SOI VLSI, offering a temperature insensitive operation. Thus the transition from today's 3.3 V supply voltage standard to the temperature insensitive supply voltage of around 1 V could happen sooner than most designers have predicted.

References

[1] R. Brodersen, A. Chandrakasan, and S. Sheng, "Design techniques for portable systems," *ISSCC'93 Tech. Dig.*, pp. 168–169, 1993.

[2] M. Kakumu, "Process and device technologies of CMOS devices for low-voltage operations," *IEICE Trans. Electronics*, vol. E76-C, pp. 672–680, May 1993.

[3] S. W. Sun and P. G. Y. Tsui, "Limitation of CMOS supply-voltage scaling by MOSFET threshold voltage variation," *IEEE J. Solid-State Circuits*, vol. 30, no. 8, pp. 947–949, Aug. 1995.

On Optimal Low-Rank Approximation of Multidimensional Discrete Signals

Wu-Sheng Lu and S.-C. Pei

***Abstract*—This brief describes an algorithmic development of the optimal low-rank approximation (LRA) of multidimensional (M-D) signals with $M \geq 3$. The algorithms developed can be regarded as a dimensional generalization of the singular value decomposition (SVD) which is of fundamental importance for analyzing signals that can be represented in a matrix form. In particular, iterative algorithms for optimal and suboptimal LRA of three-dimensional (3-D) arrays are presented in detail. Application of the 3-D LRA to the compression of image sequences is discussed.**

I. Introduction

Data array representation and approximation are of practical importance as they are closely related to the problem of data compression as well as many decomposition-based digital signal processing techniques [1]–[8]. There are two distinct classes of transform techniques that have proven useful for signal representation and approximation. One is the class of "interdomain" transform techniques that transform the signals at hand from the spatial (or time) domain to the frequency domain or vice versa. The discrete Fourier transform (DFT) and discrete cosine transform (DCT) are well known representatives in this class. The other is the class of "intradomain" transform techniques that transform the signals within the same domain. The singular value decomposition (SVD) is a typical example belonging to the second class. In [8] a multidimensional (*M*-D) outer product expansion (OPE) algorithm was proposed and applied to several sample images. Although the algorithm developed in [8] does not produce optimal LRA in general, the results reported there have demonstrated that lower bit rate can be achieved by considering the problem in a higher dimension. Mathematically a discrete *M*-D signal can be treated as an Mth-order tensor, and the approximation of *M*-D signals can be considered in tensor spaces. Reference [9] presents an approximation theory in tensor product spaces.

This brief describes an algorithmic development of the optimal low-rank approximation (LRA) of *M*-D discrete arrays with $M \geq 3$. The *M*-D LRA also belongs to the class of intradomain methods and can be viewed as a dimensional generalization of the SVD. The dimensions of arrays considered here are higher than two, and emphasis will be given to the three-dimensional (3-D) case. Application of the proposed LRA algorithms to the approximation of image sequences is presented in Section III.

II. Optimal LRA of 3-D Arrays

A. A Brief Overview of the SVD

Let A be an $m \times n$ real-valued matrix of rank r, the SVD of A is the decomposition $A = U\Sigma V^T$ where U and V are $m \times m$ and

Manuscript received January 25, 1996; revised March 17, 1997. This paper was recommended by Associate Editor C.-Y. Wu.

W.-S. Lu is with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, B.C., Canada V8W 3P6.

S.-C. Pei is with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.

Publisher Item Identifier S 1057-7130(98)00786-1.

$n \times n$ orthogonal matrices, and

$$\Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & 0 \\ & & \sigma_r & \\ & 0 & & 0 \end{bmatrix}$$

with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$. The column vectors of $U(V)$ are called the left (right) singular vectors, and $\{\sigma_1, \cdots, \sigma_r\}$ are the nonzero singular values of A. If we denote $U = [\hat{u}_1 \cdots \hat{u}_m]$ and $V = [\hat{v}_1 \cdots \hat{v}_n]$, then the SVD of A can be written as

$$A = \sum_{k=1}^{r} \sigma_k \hat{u}_k \hat{v}_k^T = \sum_{k=1}^{r} u_k v_k^T \tag{1}$$

where $u_k = \sigma_k^{1/2} \hat{u}_k$ and $v_k = \sigma_k^{1/2} \hat{v}_k$ are called the weighted left and right singular vectors, respectively. For $K \leq r$, we construct matrix A_K of rank K as

$$A_K = \sum_{k=1}^{K} \sigma_k \hat{u}_k \hat{v}_k^T = \sum_{k=1}^{K} u_k v_k^T. \tag{2}$$

The following properties are of crucial importance in many applications of the SVD, and will be frequently cited in the rest of the paper.

Property 1: Each pair of singular vectors $\{\hat{u}_k, \hat{v}_k\}$ satisfies

$$\sigma_k \hat{u}_k = A \hat{v}_k \tag{3a}$$

$$\sigma_k \hat{v}_k = A^T \hat{u}_k. \tag{3b}$$

We shall call a pair of vectors satisfying (3) a Schmidt pair. Thus a matrix with rank r possesses r Schmidt pairs which can be obtained by the SVD of the matrix.

Property 2: [10]: The matrix A_K defined by (2) is an optimal rank-K approximation of A in the 2-norm and F-norm. Namely,

$$\|A - A_K\|_{2,F} = \min_{\text{rank}(\hat{A}_K)=K} \|A - \hat{A}_K\|_{2,F}.$$

Property 3: The SVD of $A - \sum_{k=1}^{p} u_k v_k^T$ is given by $\sum_{k=p+1}^{r} u_k v_k^T$ for any integer p between 0 to r.

From Property 3 it follows that the SVD of A can be obtained by *recursively* computing the first Schmidt pair and the associated (largest) singular value of

$$A - \sum_{k=1}^{p} u_k v_k^T$$

for $p = 0, 1, \cdots, r-1$.

B. Notation and Problem Formulation

A 3-D real-valued discrete array with finite region of support can be denoted by $D = \{d_{ijk}, 1 \leq i \leq m, 1 \leq j \leq n, 1 \leq k \leq p\}$, and the Frobenius (F) norm of D is defined as

$$\|D\|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{p} d_{ijk}^2 \right)^{1/2}.$$

With the index i fixed, array D induces a matrix $D_{x,i} = \{d_{ijk}, i \text{ fixed}, 1 \leq j \leq n, 1 \leq k \leq p\}$. Similarly, matrices $D_{y,j}$ and $D_{z,k}$ can be induced from D by fixing indices j and k, respectively. Note that D can be expressed in terms of these matrix slices as

$$D = \bigcup_{i=1}^{m} D_{x,i} = \bigcup_{j=1}^{n} D_{y,j} = \bigcup_{k=1}^{p} D_{z,k}$$

where $\bigcup$ denotes union of sets. Moreover, we can compute the F-norm of D in terms of the F-norms of $D_{x,i}$, $D_{y,j}$, and $D_{z,k}$, i.e.,

$$\|D\|_F^2 = \sum_{i=1}^{m} \|D_{x,i}\|_F^2 = \sum_{j=1}^{n} \|D_{y,j}\|_F^2 = \sum_{k=1}^{p} \|D_{z,k}\|_F^2. \tag{4}$$

A 3-D array D_e is said to be elementary if it can be expressed as an outer product of three vectors $u \in R^m$, $v \in R^n$, $w \in R^p$, namely, $D_e = \{d_{ijk}\}$ with $d_{ijk} = u(i)v(j)w(k)$. In this brief, an elementary array is expressed as $D_e = u \cdot v \cdot w$. Obviously, any finite array can be decomposed into a sum of finitely many elementary arrays. An important structural parameter of D is the rank of D that tells us the minimum number of elementary arrays needed to construct D.

Definition: The rank of an $m \times n \times p$ array D is the smallest integer r such that $D = \sum_{k=1}^{r} u_k \cdot v_k \cdot w_k$, where $u_k \in R^m$, $v_k \in R^n$, and $w_k \in R^p$.

Note that this definition is consistent with the concept of rank for ordinary matrices. The optimal LRA problem to be investigated in this section can now be formulated as follows. Given a 3-D array $D = \{d_{ijk}, 1 \leq i \leq m, 1 \leq j \leq n, 1 \leq k \leq p\}$ of rank r, find a rank K approximation D_K such that

$$\|D - D_K\|_F = \min_{\text{rank}(\hat{D}_K)=K} \|D - \hat{D}_K\|_F. \tag{5}$$

Using the definition of rank, the above problem can be reformulated as to find for a given D vectors $u_k \in R^n$, $v_k \in R^m$, and $w_k \in R^p$ for $k = 1, \cdots, K$ such that the error function

$$J_K = \frac{1}{2} \left\| D - \sum_{k=1}^{K} u_k \cdot v_k \cdot w_k \right\|_F^2 \tag{6}$$

is minimized.

C. An Optimal Solution to the 3-D LRA Problem

Define

$$u = \begin{bmatrix} u_1 \\ \vdots \\ u_K \end{bmatrix}_{Kn \times 1}, \qquad v = \begin{bmatrix} v_1 \\ \vdots \\ v_K \end{bmatrix}_{Km \times 1}, \qquad w = \begin{bmatrix} w_1 \\ \vdots \\ w_K \end{bmatrix}_{Kp \times 1}$$

where u_k, v_k, and w_k themselves are vectors denoted by

$$u_k = \begin{bmatrix} u_{1k} \\ \vdots \\ u_{mk} \end{bmatrix}, \quad v_k = \begin{bmatrix} v_{1k} \\ \vdots \\ v_{nk} \end{bmatrix}, \quad w_k = \begin{bmatrix} w_{1k} \\ \vdots \\ w_{pk} \end{bmatrix}.$$

If we temporarily fix $u_1, \cdots, u_K$ in (6) and use (4), the error function can be computed as follows:

$$\begin{aligned} J_K(u, v, w) &= \frac{1}{2} \sum_{i=1}^{m} \left\| D_{x,i} - \sum_{k=1}^{K} u_{ik} (v_k w_k^T) \right\|_F^2 \\ &= \frac{1}{2} \text{tr} \left\{ \sum_{i=1}^{m} \left[D_{x,i}^T - \sum_{k=1}^{K} u_{ik} (w_k v_k^T) \right] \right. \\ &\qquad \left. \cdot \left[D_{x,i} - \sum_{k=1}^{K} u_{ik} (v_k w_k^T) \right] \right\} \\ &= \frac{1}{2} \sum_{k=1}^{K} \sum_{j=1}^{K} a_{kj} b_{kj} c_{kj} \\ &\qquad - \sum_{k=1}^{K} v_k^T S_{x,k} w_k + c_x \end{aligned} \tag{7}$$

where

$$a_{kj} = u_k^T u_j, \quad b_{kj} = v_k^T v_j, \quad c_{kj} = w_k^T w_j \tag{8a}$$

$$S_{x,k} = \sum_{i=1}^{m} u_{ik} D_{x,i} \tag{8b}$$

$$c_x = \tfrac{1}{2} \sum_{i=1}^{m} \|D_{x,i}\|_F^2.$$

Hence, $\partial J_K / \partial v_k = 0$ and $\partial J_K / \partial w_k = 0$ for $k = 1, \cdots, K$ yield

$$\sum_{j=1}^{K} a_{kj} c_{kj} v_j = S_{x,k} w_k \tag{9a}$$

$$\sum_{j=1}^{K} a_{kj} b_{kj} w_j = S_{x,k}^T v_k. \tag{9b}$$

By using the Kronecker product notation, the above K sets of equations can be combined into the following pair of matrix equations:

$$(C \otimes I_n) v = S_x w \tag{10a}$$

$$(B \otimes I_p) w = S_x^T v \tag{10b}$$

where

$$C = \begin{bmatrix} a_{11}c_{11} & \cdots & a_{1K}c_{1K} \\ a_{21}c_{21} & \cdots & a_{2K}c_{2K} \\ \vdots & & \vdots \\ a_{K1}c_{K1} & \cdots & a_{KK}c_{KK} \end{bmatrix}$$

$$B = \begin{bmatrix} a_{11}b_{11} & \cdots & a_{1K}b_{1K} \\ a_{21}b_{21} & \cdots & a_{2K}b_{2K} \\ \vdots & & \vdots \\ a_{K1}b_{K1} & \cdots & a_{KK}b_{KK} \end{bmatrix} \tag{11a}$$

$$S_x = \operatorname{diag}\{S_{x,1}, \cdots, S_{x,K}\}. \tag{11b}$$

I_n and I_p are $n \times n$ and $p \times p$ identity matrices, respectively, and $\otimes$ denotes the Kronecker product [10].

Equations (10a) and (10b) are two key equations in our study because they not only characterize the vectors v and w that minimize the error function J_K (with a fixed u!), but also exhibit a structure similar to (3a) and (3b), from which the consistency of our development with the conventional SVD can be appreciated. As a matter of fact, if array D is degenerated to a matrix and thus vectors $\{u_i,\ 1 \le i \le K\}$ and $\{a_{kj}\}$ are eliminated in (8)–(10), then matrix $S_{x,k}$ defined by (8b) becomes matrix D itself, and (9a) and (9b) become

$$\sum_{j=1}^{K} c_{kj} v_j = D w_k \tag{12a}$$

$$\sum_{j=1}^{K} b_{kj} w_j = D^T v_k. \tag{12b}$$

Note that although (12a) and (12b) are still nonlinear in v and w, they are readily solvable: the SVD of D provides a solution to (12). In fact, if $\{v_k,\ 1 \le k \le K\}$ and $\{w_k,\ 1 \le k \le K\}$ are the first K pairs of the singular vectors, (12a) and (12b) become $\sigma_k v_k = D w_k$, $\sigma_k w_k = D^T v_k$ which are satisfied by v_k and w_k automatically since $\{v_k, w_k\}$ is the kth Schmidt pair of D. On comparing (12) to (9), we see a substantial difference between the two-dimensional (2-D) and 3-D cases: as far as $K > 1$, vectors v_k and w_k in (9) cannot be obtained by the SVD since matrix $S_{x,k}/\|u_k\|^2$ depends on index k, and the Schmidt pairs obtained from that matrix with different index k are not orthogonal to each other in general. In what follows, we shall focus on (10) and develop an iterative method to solve the problem with $K > 1$. The $K = 1$ case is treated in Section III-D where a suboptimal solution to the problem will be deduced.

From (8a) it follows that both C and B are symmetric and nonsingular, hence (10) can be written as

$$v = [C^{-1}(w) \otimes I_n] S_x w \tag{13a}$$

$$w = [B^{-1}(v) \otimes I_p] S_x^T v \tag{13b}$$

where the property that for invertible P and Q, $(P \otimes Q)^{-1} = P^{-1} \otimes Q^{-1}$ has been used and C^{-1}, B^{-1} have been written as $C^{-1}(w)$, $B^{-1}(v)$ to emphasize their dependence on w and v. Equation (13) suggests the following iterative scheme for solving (10):

$$\tilde{v}^{(i)} = [C^{-1}(w^{(i)}) \otimes I_n] S_x w^{(i)} \tag{14a}$$

$$\tilde{w}^{(i)} = [B^{-1}(v^{(i)}) \otimes I_p] S_x^T v^{(i)} \tag{14b}$$

and

$$v^{(i+1)} = \alpha \tilde{v}^{(i)} + (1-\alpha) v^{(i)} \tag{14c}$$

$$w^{(i+1)} = \alpha \tilde{w}^{(i)} + (1-\alpha) w^{(i)} \tag{14d}$$

for $i = 0, 1, \cdots$, where $0 < \alpha < 1$ is a scalar weight. Our numerical study of the algorithm indicates that a value of α between 0.4 and 0.8 often leads to a satisfactory convergence rate. The initial vectors $v^{(0)}$ and $w^{(0)}$ in (14) can be chosen arbitrarily, but a better choice is to form $v^{(0)}$ and $w^{(0)}$ using the first K pairs of weighted singular vectors of S_x. The iteration continues until $\|v^{(i+1)} - v^{(i)}\| + \|w^{(i+1)} - w^{(i)}\|$ is less than a prescribed tolerance, and at that time $v = v^{(i+1)}$ and $w = w^{(i+1)}$ are claimed to be the solution of (10). Although a mathematical convergence proof of scheme (14) has not been available, in our simulation study scheme (14) works well for arrays of various sizes.

Now recall that the above solution vectors v and w are obtained with a fixed u, hence the triple $\{u, v, w\}$ is unlikely to be the one that minimizes J_K in (6). However, since a quasioptimal pair $\{v, w\}$ has been obtained, the pair can be used to obtain an improved u by minimizing J_K in (6) with respect to u, with v and w fixed. From (7) we have

$$\frac{\partial J_K}{\partial u_k} = \sum_{j=1}^{K} b_{kj} c_{kj} u_j - h_{x,k} \tag{15}$$

where b_{kj} and c_{kj} are defined in (8a), and

$$h_{x,k} = \begin{bmatrix} v_k^T D_{x,1} \\ \vdots \\ v_k^T D_{x,m} \end{bmatrix} w_k. \tag{16}$$

Hence, $\partial J_K / \partial u_k = 0$ for $k = 1, \cdots, K$ is the linear system of equation given by

$$(B_c \otimes I_m) u = h_x$$

where

$$B_c = \begin{bmatrix} b_{11}c_{11} & \cdots & b_{1K}c_{1K} \\ b_{21}c_{21} & \cdots & b_{2K}c_{2K} \\ \vdots & & \vdots \\ b_{K1}c_{K1} & \cdots & b_{KK}c_{KK} \end{bmatrix}, \quad h_x = \begin{bmatrix} h_{x,1} \\ \vdots \\ h_{x,K} \end{bmatrix}. \tag{17}$$

Since B_c is symmetric and nonsingular, we obtain

$$u = (B_c^{-1} \otimes I_m) h_x. \tag{18}$$

With the new vector u from (18), matrices S_x, B, and C in (11) are updated, and an improved pair $\{v, w\}$ can be computed using iteration (14). This procedure repeats until the difference between the new triple $\{u, v, w\}$ and the preceding triple is less than a given

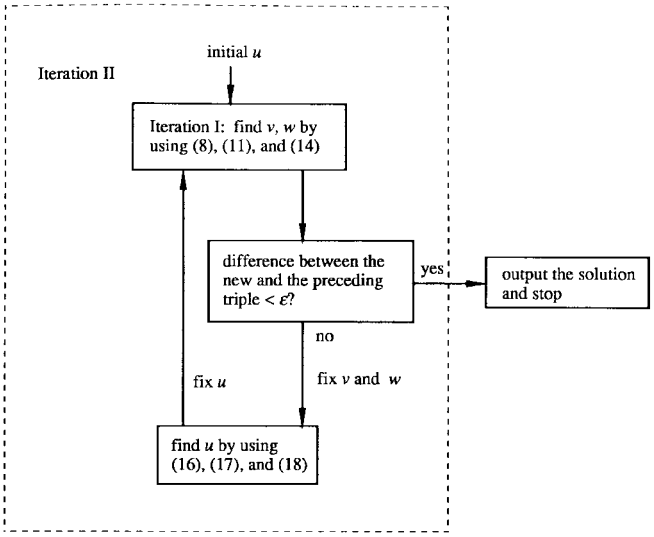

Fig. 1. Algorithm 1—the two-level iteration approach to solving the 3-D optimal LRA problem.

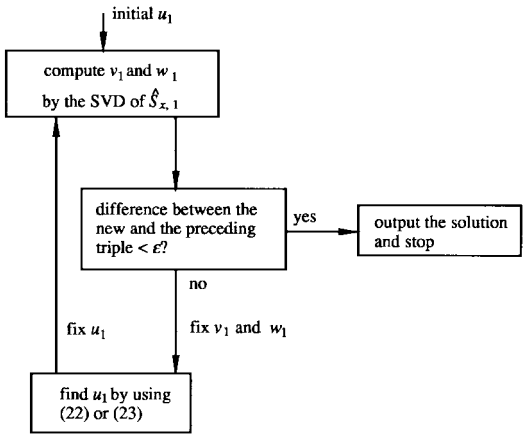

Fig. 2. Algorithm 2—the iterative approach to solving the 3-D optimal LRA problem with $K = 1$.

tolerance. In the rest of the brief, the above solution procedure will be referred to as Algorithm 1, and a diagrammatic explanation of the algorithm is shown in Fig. 1.

Two remarks about the algorithm are now in order. First, a solution obtained from Algorithm 1 is only a *local* minimum of J_K in (6). By (6) we see that J_K is a highly nonlinear function of the unknown vectors—if we collect all u, v, and w and define $x = [u^T\ v^T\ w^T]^T$, then J_K is a 6th-order polynomial of x. Note also that Algorithm 1 starts with an initial u, consequently the solution so obtained depends on the initial u. A possible way to obtain an improved local minimum is to set up an additional optimization process at a level higher than the optimization sketched in Fig. 1 as a supervisory mechanism which feeds an initial u into the optimization process in Fig. 1, compares its output to the outputs obtained from different initial u, and generates a new initial u that would lead to a better local solution. Second, the u_k's (and v_k's and w_k's) obtained from Algorithm 1 are not orthogonal in general. Although similarity between the key equations (3) (for matrices) and (10) (for the 3-D arrays) exists, the crucial difference between them is that (3) implies $\sigma_k^2 \hat{u}_k = AA^T u_k$ and $\sigma_k^2 \hat{v}_k = A^T A \hat{v}_k$, thus the orthogonality of $\{\hat{u}_k\text{'s}\}$ (and $\hat{v}_k$'s) is an immediate consequence from the theory of symmetric matrices; however, even for a fixed u, the matrices involved in (10), namely S_x, B, and C, are dependent nonlinearly on vectors v and w, and therefore the orthogonality of $\{u_k\text{'s}\}$ (and $\{v_k\text{'s}\}$ and $\{w_k\text{'s}\}$) do not hold for $M = 3$ and beyond.

D. Evaluation of u_k, $v^{(i+1)}$, and $w^{(i+1)}$

The dimensions of $v^{(i+1)}$, $w^{(i+1)}$ in (14) and u in (18) are Kn, Kp, and Km, which could be fairly high for a large size array. Consequently, the matrices involved in (14) and (18) may cause storage difficulties for the computer. The problem can be considerably eased off by taking the advantage of the special structure of the Kronecker products involved as well as the block diagonal structure of S_x. As a matter of fact, if we denote $C^{-1}(w^{(i)}) = \{f_{kj}^{(i)}\}$ and $B^{-1}[v^{(i)}] = \{e_{kj}^{(i)}\}$, then (14) implies that, for $k = 1, \cdots, K$,

$$\tilde{v}_k^{(i)} = \sum_{j=1}^{K} f_{kj}^{(i)} S_{x,j} w_j^{(i)} \tag{19a}$$

$$\tilde{w}_k^{(i)} = \sum_{j=1}^{K} e_{kj}^{(i)} S_{x,j}^T v_j^{(i)}. \tag{19b}$$

Similarly, denoting $B_c^{-1} = \{g_{kj}\}$, (18) then implies that, for $k = 1, \cdots, K$,

$$u_k = \sum_{j=1}^{K} g_{kj} h_{x,j}. \tag{20}$$

From (19) we see that each vector $\tilde{v}_k^{(i)}(\tilde{w}_k^{(i)})$ can be evaluated by summing up K vectors, each of which is a result of a matrix multiplication of size $n \times p$ by $p \times 1$ ($p \times n$ by $n \times 1$), followed by a scalar multiplication. As compared to (14), (19) considerably reduces the data memory required. Similarly, compared to (18), the evaluation of u_k using (20) requires a much-reduced data memory. So for algorithm implementation, (14) and (18) in Fig. 1 should be replaced by (19) and (20), respectively.

E. The $K = 1$ Case and a Suboptimal Solution

The $K = 1$ case corresponds to the problem of finding vectors $u_1 \in R^m$, $v_1 \in R^n$, and $w_1 \in R^p$ such that $J_1 = \|D - u_1 \cdot v_1 \cdot w_1\|_F^2$ is minimized. With $K = 1$, (9a) and (9b) are reduced to

$$\|u_1\|^2 \|w_1\|^2 v_1 = S_{x,1} w_1 \tag{21a}$$

$$\|u_1\|^2 \|v_1\|^2 w_1 = S_{x,1}^T v_1 \tag{21b}$$

i.e., $\sigma_1 v^* = \hat{S}_x w^*$ and $\sigma_1 w^* = \hat{S}_x^T v^*$, where $\sigma_1 = \|v_1\| \|w_1\|$, $v^* = v_1/\|v_1\|$, $w^* = w_1/\|w_1\|$, and

$$\hat{S}_x = S_{x,1}/\|u_1\|^2 = \frac{1}{\|u_1\|^2} \sum_{i=1}^{m} u_{i1} D_{x,i}.$$

On comparing (21) to (3), we conclude that v^* and w^* can be obtained as the first Schmidt pair of $\hat{S}_{x,1}$, and σ_1 is the largest singular value of $\hat{S}_{x,1}$. Having obtained σ_1, v^*, and w^*, v_1 and w_1 can be obtained as $v_1 = \sigma_1^{1/2} v^*$ and $w_1 = \sigma_1^{1/2} w^*$. Evidently, this SVD approach greatly reduces the computation complexity that the $K > 1$ case requires. Again we recall that the above v_1 and w_1 are computed to a fixed u_1, hence the second-level iteration is still needed here to update u_1. Taking $K = 1$ in (15) and setting $\partial J_1/\partial u_1 = 0$, we obtain

$$u_1 = \frac{1}{\|v_1\|^2 \|w_1\|^2} h_{x,1} \tag{22}$$

i.e.,

$$u_1 = \frac{1}{\sigma_1} \begin{bmatrix} v^{*T} D_{x,1} \\ \vdots \\ v^{*T} D_{x,m} \end{bmatrix} w^*. \tag{23}$$

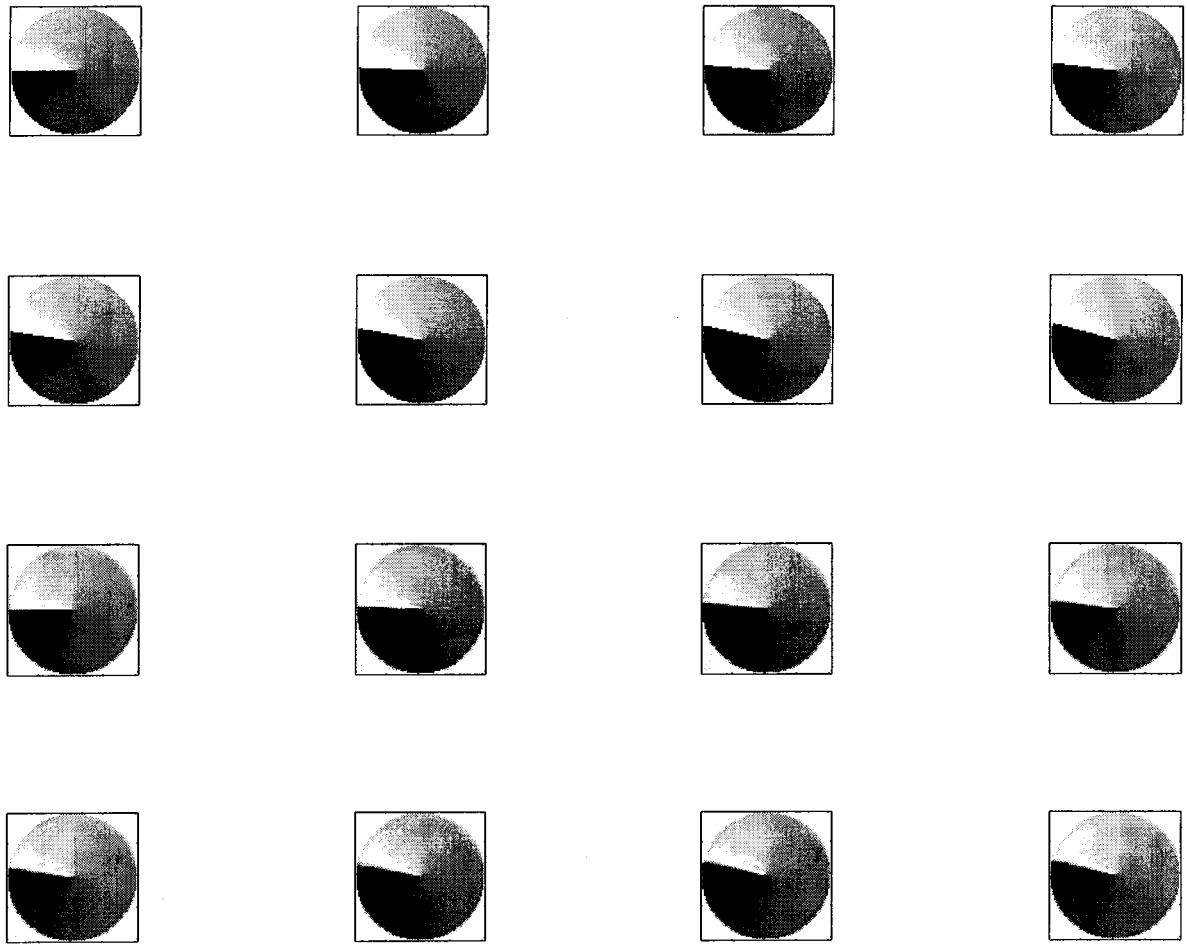

Fig. 3. Even number of frames. The two upper rows are original images, and the two lower rows are optimal rank-20 approximations.

Equation (23) provides a formula for the evaluation of u_1 in terms of σ_1, v^*, and w^*. The vector u_1 so obtained is in turn used to update matrix $\hat{S}_x$ whose SVD yields an improved Schmidt pair $\{v^*, w^*\}$ and σ_1. Like the general case, this iteration continues until the new triple $\{u_1, v_1, w_1\}$ shows no difference from the preceding one up to a prescribed tolerance. The above solution procedure will be referred to as Algorithm 2, and a diagram that summarizes the algorithm is shown in Fig. 2.

Motivated by its reduced computation complexity, a suboptimal solution to the LRA problem may be obtained by applying Algorithm $2K$ times. That is, once the first optimal triple $\{u_1, v_1, w_1\}$ is found, the residual array

$$R_1 = D - u_1 \cdot v_1 \cdot w_1$$

is evaluated, to which the same algorithm is applied to obtain the second triple $\{u_2, v_2, w_2\}$ that best approximates R_1. In general, at the kth step of the procedure, the $(k-1)$th residual array

$$R_{k-1} = D - \sum_{i=1}^{k-1} u_i \cdot v_i \cdot w_i$$

is constructed, to which Algorithm 2 is applied to obtain the kth triple $\{u_k, v_k, w_k\}$. As far as the rank of D is greater than K, applying Algorithm 2 K times generates a residual sequence $R_1, R_2, \cdots, R_K$ whose F-norm is strictly monotonically decreasing, and each $\|R_k\|_F$ can be viewed as the error of the approximation of D by $D_k = \sum_{i=1}^{k} u_i \cdot v_i \cdot w_i$. We shall call D_K so obtained a suboptimal solution to the LRA problem.

We now conclude this section with two remarks. First, we emphasize that the solution obtained by the above approach is a local one, and that the rank-K approximation D_K is only *suboptimal*, except the $K = 1$ case. This is to say that for a given $K > 1$, the approximation made using this method yields larger error as opposed to that of Algorithm 1. This is another important difference between the processing of 3-D and 2-D arrays. As was noted at the end of Section II-A, for a 2-D array these two solutions are identical. We shall touch upon this issue again in Section III. Second, it is noted that the suboptimal solution was also proposed in [8]. However, the analytic method used there to compute the solution is different from ours.

III. Approximation of an Image Sequence

In this section, we describe an example in which the algorithms developed in Section II are applied to an image sequence in order to obtain its approximations that are close enough to the original sequence with substantially reduced data storage. The sequence at hand contains sixteen frames of 81×81 images that describes a circular object with 256 gray levels (8 bits) rotating about 15° with respect its center. This image sequence is selected for testing the algorithms for two reasons: 1) each image has edges (one line segment and one circle) as well as smooth areas where the gray levels are linearly distributed; and 2) a large portion of the object, i.e., the disk, is moving throughout the sequence. The even number of images of the sequence are shown in the two upper rows in Fig. 3. Algorithms 1 and 2 with $\alpha = 0.5$ were applied to the sequence with K varying from

TABLE I
APPROXIMATION ERRORS OF ALGORITHMS 1 AND 2 IN EXAMPLE 1

	Algorithm 1		Algorithm 2	
rank K	appr. error	Mflops	appr. error	Mflops
2	54.6972	2.7105	54.6972	46.6412
4	37.3053	2.3816K	40.9826	143.1240
6	30.5844	4.0426K	33.7879	322.8889
8	26.3313	5.6774K	29.6992	500.5248
10	23.4154	15.7483K	25.7247	603.3538
12	21.1695	21.5096K	23.2810	739.7914
14	19.7351	27.9546K	21.6194	897.4940
16	18.1256	42.8397K	19.8908	1.1770K
18	16.3969	301.6440K	18.9858	1.3349K
20	15.1310	1270.37K	17.7717	1.4397K

2 to 20, and the approximation errors obtained are listed in Table I. It is observed that the optimal LRA's that Algorithm 1 generates are consistently better than those obtained from Algorithm 2, although the computation complexity of Algorithm 1 grows rapidly with K.

From Table I we see that with $K = 20$, the F-norm of the optimal LRA is 15.1310. This means that, on average, the error for each pixel of the sequence is about

$$\frac{15.1310}{\sqrt{16 \times 81 \times 81}} \approx 0.0467$$

for the normalized range of gray level [0, 1]. The even number of images from the optimal rank-20 optimations are shown in the two lower rows in Fig. 3. The ratio of the total number of entries of the original sequence to the total number of entries required by the rank-20 approximation is

$$\eta = \frac{16 \times 81 \times 81}{20 \times (16 + 81 + 81)} = 29.4876.$$

It is important to stress that by using the proposed approximation method combined with the coding techniques similar to [4] or [11], where statistical properties of the singular vectors are taken into account, substantially higher compression ratio can be achieved.

IV. CONCLUDING REMARKS

We have described two algorithms for LRA of 3-D signals. Extension of the algorithms to the four-dimensional (4-D) case is straightforward, and omitted here. It is interesting to note that the problem of approximating a 2-D array can also be tackled in a 4-D framework. Indeed, using the one-to-one mapping

$$D(i, j, l, k) = A[(i-1)N_2 + j, (l-1)N_2 + k]$$

where $1 \le i, l \le N_1$, $1 \le j, k \le N_2$, with N_1 and N_2 being the positive integers satisfying $n = N_1 N_2$, the 2-D array A is converted into the 4-D array D of size $N_1 \times N_2 \times N_1 \times N_2$.

ACKNOWLEDGMENT

The authors are grateful to the reviewers for their constructive comments and for bringing [9] to their attention.

REFERENCES

[1] L. L. Scharf, "The SVD and reduced rank signal processing," *Signal Process.*, vol. 25, pp. 113–133, Nov. 1991.
[2] H. C. Andrews and C. L. Patterson, "Singular value decomposition and digital image processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 26–53, 1976.
[3] ——, "Outer product expansions and their use in digital image processing," *IEEE Trans. Comput.*, vol. C-25, pp. 140–148, 1976.
[4] N. Garguir, "Comparative performance of SVD and adaptive cosine transform in coding images," *IEEE Trans. Commun.*, vol. COM-27, pp. 1230–1234, Aug. 1979.
[5] D. P. O'Leary and S. Peleg, "Digital image compression by outer product expansion," *IEEE Trans. Commun.*, vol. COM-31, pp. 441–444, Mar. 1983.
[6] M. Ohki and M. Kawamata, "Design of three-dimensional digital filters based on the outer product expansion," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 1164–1167, Sept. 1990.
[7] W.-S. Lu, H.-P. Wang, and A. Antoniou, "Design of 3-D digital filters by using outer-product array decomposition," in *Proc. 3rd Int. Conf. Advances Commun. Contr. Syst.*, Victoria, B.C., Canada, Oct. 1991.
[8] T. Saitoh, T. Komatsu, H. Harasima, and H. Miyakawa, "Still picture coding by multidimensional outer product expansion," *IECE Trans.*, vol. J68-B, pp. 547–548, Apr. 1985.
[9] W. A. Light and E. W. Cheney, *Approximation Theory in Tensor Product Spaces.* New York: Springer-Verlag, 1985.
[10] R. A. Horn and C. R. Johnson, *Matrix Analysis.* Cambridge, U.K.: Cambridge Univ. Press, 1985.
[11] T. Minami, H. Sakamoto, A. Suzuki, and O. Nakamura, "Encoding of pictures using the singular value decomposition (SVD) and 1-D discrete cosine transform (DCT)," in *Proc. IEEE Int. Commun. Conf.*, Seattle, WA, June 1995, pp. 1418–1422.

A Noise-Exclusive Adaptive Filtering Framework for Removing Impulse Noise in Digital Images

H. Kong and L. Guan

***Abstract*— A class of noise-exclusive adaptive filters for removing impulse noise from digital images is developed and analyzed in this brief. The filtering scheme is based on noise detection using a self-organizing neural network and noise excluding estimation. These filters suppress impulse noise effectively while preserving fine image details. Applications of the filters to several images show that their properties of efficient impulse noise suppression, edges and fine details preservation, minimum signal distortion, or minimum mean square error are better than those of the traditional median-type filters.**

I. INTRODUCTION

When an image is coded and transmitted over a noisy channel or degraded by electrical sensor noise, degradation appears as salt-and-pepper noise (i.e., positive and negative impulses) [1]. Removal of such impulse noise while preserving the integrity of the image is an essential issue in image processing. The well-known median filter has been recognized as an effective technique for impulse noise suppression due to its edge preserving characteristic and its computational simplicity [2], [3].

Manuscript received April 23, 1996; revised October 11, 1996. This paper was recommended by Associate Editor C.-Y. Wu.

The authors are with the Department of Electrical Engineering, The University of Sydney, Sydney, NSW 2006 Australia.

Publisher Item Identifier S 1057-7130(98)00780-0.