Is the SVD-Based Low-Rank Approximation Optimal?

W.-S. Lu

Dept. of Elec. & Comp. Engineering
University of Victoria
Victoria, B.C., Canada V8W 3P6
E-mail: wslu@ece.uvic.ca

ABSTRACT

It is well known that the singular value decomposition (SVD) offers an optional approximation in F-norm for a given image. In this paper, we show that using a 2D-to-4D mapping in conjunction with optimal 4-D low rank approximation, improved compression rate can be achieved compared to that obtained from the conventional (2-D) SVD approach.

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. Consider an $n \times n$ gray level image characterized by matrix $A \in R^{n\times n}$, each of whose entries represents the gray level of the corresponding pixel. Assume the rank of A is r, then the SVD of A is the representation

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

where $\sigma_1 \geq \cdots \geq \sigma_r > 0$ are the singular values of A, $\{\hat{u}_k,\, \hat{v}_k\}$ is the kth pair of singular vectors of A, and $u_k = \sigma_k^{1/2}\hat{u}_k$, $v_k = \sigma_k^{1/2}\hat{v}_k$. Here a two-dimensional (2-D) array in the spatial domain is transformed within the same domain but the signal is decomposed into a low-dimensional (one-dimensional (1-D) in this case) representation. The importance of this decomposition arises from the fact that the best $K(K < r)$ outer products whose sum optimally approximates A in the Frobenious norm (F-norm) and Euclidean norm (2-norm) sense are given by $\{u_k v_k^T,\ k = 1,\ \ldots,\ K\}$ with error bounds $(\sum_{K+1}^{n} \sigma_k^2)^{1/2}$ and σ_{K+1}, respectively [9][10]. Note that each $u_k v_k^T$, which is often referred to as the kth singular image, has only $2n$ entries to store and its importance compared with other singular images can be measured by the associated singular value σ_k. These features of the SVD have led to its applications in image representation and compression [2]–[5].

In an image-processing context, the reason we are interested in the solution provided by the SVD is because it offers an optimal approximation in F-norm with a suitable structure: it is given in terms of a set of individual singular images with a distinguishable "importance measure". Therefore, lower bit rate compression can be achieved by assigning less bits to less important singular images [4]. 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 $1 \leq i,\ l \leq N_1$, $1 \leq j,\ k \leq N_2$, 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 $N_1 \times N_2 \times N_1 \times N_2$. A question that arises at this point is whether or not a bit rate lower than the conventional SVD approach can be achieved via a 4-D LRA of array D. In [8] an 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 this 4-D approach.

In this paper we present a theory with accompanied algorithmic development for optimal and suboptimal 4-D low rank approximations (LRA) of a 2-D signal.

II. OPTIMAL LRA of 4-D Arrays

A. Notation, Definitions and Problem Formulation

A real-valued 4-D array can be denoted by $D = \{d_{kjkl},\ 1 \leq i \leq M_1,\ 1 \leq j \leq M_2,\ 1 \leq k \leq N_1,\ 1 \leq l \leq N_2\}$. The F-norm of D is defined by

0-7803-2972-4/96$5.00©1996 IEEE

$$||D||_F = \left[\sum_{i=1}^{M_1}\sum_{j=1}^{M_2}\sum_{k=1}^{N_1}\sum_{p=1}^{N_2} d_{ijkl}^2\right]^{1/2}$$

A 4-D array is said to be elementary if it can be expressed as $D_e = \{d_{ijkl}\}$ with $d_{ijkl} = r(i)s(j)u(k)v(l)$ for some vectors r, s, u, and v. The rank of D is defined as the smallest integer q such that

$$D = \sum_{i=1}^{q} r_i \cdot s_i \cdot u_i \cdot v_i$$

for some vectors r_i, s_i, u_i and v_i, $1 \leq i \leq q$.

Although a 4-D array itself can hardly be visualized, it can be characterized by several low-dimensional "facets" which are visualizable. For instance, by fixing two of the indexes, say i and j, D becomes an $N_1 \times N_2$ matrix. If we associate each index with an axis, say $i \longrightarrow w$, $j \longrightarrow x$, $k \longrightarrow y$, and $l \longrightarrow z$, then with i and j fixed D is the matrix obtained by projecting it onto the plane $\{w = i\}\bigcap\{x = j\}$ where $\{w = i\}$ and $\{x = j\}$ are 3-D space, and $\bigcap$ denotes the set intersection. From this interpretation we can write

$$D = \bigcup_{i,j} D_{ij} \tag{1}$$

where D_{ij} is the projection of D onto the plane $\{w = i\}\bigcap\{x = j\}$ with $i = 1, \ldots, M_1$, $j = 1, \ldots, M_2$. Note that (1) gives

$$||D||_F^2 = \sum_{i=1}^{M_1}\sum_{j=1}^{M_2} ||D_{ij}||_F^2 \tag{2}$$

The problem to be studied in this section can now be formulated as follows. Given a finite 4-D array D, and integer K with $K \leq \text{rank}(D)$, find a rank K approximation of D, D_K, such that $||D - D_K||_F$ is minimized among all rank K arrays. Equivalently, our problem is to find vectors r_m, s_m, u_m, v_m for $m = 1, \ldots, K$ such that

$$\left|\left|D - \sum_{m=1}^{K} \tilde{r}_m \cdot \tilde{s}_m \cdot \tilde{u}_m \cdot \tilde{v}_m\right|\right|_F$$

is minimized.

B. An Optimal Solution to the 4-D LRA Problem

Define for the given $M_1 \times M_2 \times N_1 \times N_2$ array D the error function

$$J_K(r, s, u, v) = \frac{1}{2}\left|\left|D - \sum_{m=1}^{K} r_m \cdot s_m \cdot u_m \cdot v_m\right|\right|_F^2 \tag{3}$$

where $r_m \in R^{M_1}$, $s_m \in R^{M_2}$, $u_m \in R^{N_1}$, $v_m \in R^{N_1}$, and

$$r = \begin{bmatrix} r_1 \\ \vdots \\ r_K \end{bmatrix}, \; s = \begin{bmatrix} s_1 \\ \vdots \\ s_K \end{bmatrix}, \; u = \begin{bmatrix} u_1 \\ \vdots \\ u_K \end{bmatrix}, \; v = \begin{bmatrix} v_1 \\ \vdots \\ v_K \end{bmatrix}$$

In what follows we shall use $r_m(i)$, $s_m(j)$, $u_m(k)$, and $v_m(l)$ to denote the ith, jth, kth and lth entry of r_m, s_m, u_m, and v_m, respectively. For fixed r and s, we use (2) and elementary properties of matrix trace to compute

$$\begin{aligned} J_K &= \frac{1}{2}\sum_{i=1}^{M_1}\sum_{j=1}^{M_2}\left|\left|D_{ij} - \sum_{m=1}^{K} r_m(i)s_m(j)u_m v_m^T\right|\right|_F^2 \\ &= \frac{1}{2}\sum_{m=1}^{K}\sum_{n=1}^{K} a_{mn}b_{mn}c_{mn}d_{mn} - \sum_{m=1}^{K} u_m^T S_{wx}^{(m)} v_m + \mathrm{c} \end{aligned} \tag{4}$$

where

$$a_{mn} = r_m^T r_n, \; b_{mn} = s_m^T s_n, \; c_{mn} = u_m^T u_n, \; d_{mn} = v_m^T v_n,$$

$$S_{wx}^{(m)} = \sum_{i=1}^{M_1}\sum_{j=1}^{M_2} r_m(i)s_m(j)D_{ij}$$

and c is a constant independent of u and v.

Using (4) to set $\partial J_K/\partial u_m = 0$ and $\partial J_K/\partial v_m = 0$ for $m = 1, \ldots, K$, we obtain

$$(D \otimes I_{N_1})u = S_{wx}v \tag{5}$$

$$(C \otimes I_{N_2})v = S_{wx}^T u \tag{6}$$

where

$$D = \begin{bmatrix} a_{11}b_{11}d_{11} & \cdots & a_{1K}b_{1K}d_{1K} \\ a_{21}b_{21}d_{21} & \cdots & a_{2K}b_{2K}d_{2K} \\ \vdots & & \vdots \\ a_{K1}b_{K1}d_{K1} & \cdots & a_{KK}b_{KK}d_{KK} \end{bmatrix} \tag{7}$$

$$C = \begin{bmatrix} a_{11}b_{11}c_{11} & \cdots & a_{1K}b_{1K}c_{1K} \\ a_{21}b_{21}c_{21} & \cdots & a_{2K}b_{2K}c_{2K} \\ \vdots & & \vdots \\ a_{K1}b_{K1}c_{K1} & \cdots & a_{KK}b_{KK}c_{KK} \end{bmatrix} \tag{8}$$

$$S_{wx} = \text{diag}\{S_{wx}^{(1)}, \ldots, S_{wx}^{(K)}\} \tag{9}$$

Note that matrices D and C are symmetric and nonsingular, hence (5), (6) can be written as

$$u = [D^{-1}(v) \otimes I_{N_1}]S_{wx}v \tag{10}$$

$$v = [C^{-1}(u) \otimes I_{N_2}]S_{wx}^T u \tag{11}$$

which suggests an iterative scheme for solving nonlinear system of equations (10), (11) as follows:

$$\tilde{u}^{(i)} = [D^{-1}(v^{(i)}) \otimes I_{N_1}]S_{wx}v^{(i)} \tag{12}$$

$$\tilde{v}^{(i)} = [C^{-1}(u^{(i)}) \otimes I_{N_2}] S_{wx}^T u^{(i)} \quad (13)$$

$$u^{(i+1)} = \alpha \tilde{u}^{(i)} + (1-\alpha) u^{(i)} \quad (14)$$

$$v^{(i+1)} = \alpha \tilde{v}^{(i)} + (1-\alpha) v^{(i)} \quad (15)$$

for $i = 0, 1, \ldots$, and $0 < \alpha < 1$. With the vectors u and v fixed, we end up with a problem requesting to find vectors r and s that minimize J_K. Obviously, iterations of the same type are needed, and equations similar to (7)–(15) can be derived. These equations are given below without derivation details.

$$(B \otimes I_{M_1}) r = S_{yz} s \quad (16)$$

$$(A \otimes I_{M_2}) s = S_{yz}^T r \quad (17)$$

where

$$B = \begin{bmatrix} b_{11}c_{11}d_{11} & \cdots & b_{1K}c_{1K}d_{1K} \\ b_{21}c_{21}d_{21} & \cdots & b_{2K}c_{2K}d_{2K} \\ \vdots & & \vdots \\ b_{K1}c_{K1}d_{K1} & \cdots & b_{KK}c_{KK}d_{KK} \end{bmatrix} \quad (18)$$

$$A = \begin{bmatrix} a_{11}c_{11}d_{11} & \cdots & a_{1K}c_{1K}d_{1K} \\ a_{21}c_{21}d_{21} & \cdots & a_{2K}c_{2K}d_{2K} \\ \vdots & & \vdots \\ a_{K1}c_{K1}d_{K1} & \cdots & a_{KK}c_{KK}d_{KK} \end{bmatrix} \quad (19)$$

$$S_{yz} = \mathrm{diag}\{S_{yz}^{(1)}, \ldots, S_{yz}^{(K)}\} \quad (20)$$

$$S_{yz}^{(m)} = \sum_{k=1}^{N_1} \sum_{l=1}^{N_2} u_m(k) v_m(l) D_{kl} \quad (21)$$

$$r = [B^{-1}(s) \otimes I_{M_1}] S_{yz} s \quad (22)$$

$$s = [A^{-1}(r) \otimes I_{M_2}] S_{yz}^T r \quad (23)$$

$$\tilde{r}^{(i)} = [B^{-1}(s^{(i)}) \otimes I_{M_1}] S_{yz} s^{(i)} \quad (24)$$

$$\tilde{s}^{(i)} = [A^{-1}(r^{(i)}) \otimes I_{M_2}] S_{yz}^T r^{(i)} \quad (25)$$

$$r^{(i+1)} = \alpha \tilde{r}^{(i)} + (1-\alpha) r^{(i)} \quad (26)$$

$$s^{(i+1)} = \alpha \tilde{s}^{(i)} + (1-\alpha) s^{(i)} \quad (27)$$

A second level iteration is needed between the above two sets of iterations. The iteration continues until the new quadruple $\{r,\ s,\ u,\ v\}$ shows no difference from the preceding one up to a prescribed tolerance. This solution procedure will be referred to as Algorithm 1.

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

The $K = 1$ case corresponds to the problem of finding vectors r_1, s_1, u_1, and v_1 such that $J_1 = ||D - r_1 \cdot s_1 \cdot u_1 \cdot v_1||_F^2$ is minimized. Using (4), it can readily be shown that with fixed r_1 and s_1, the vectors u_1 and v_1 that minimize J_1 can be obtained by the SVD of matrix $\hat{S}_{wx}$ which is defined by

$$\hat{S}_{wx} = \frac{1}{||r_1||^2 ||s_1||^2} \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} r_1(i) s_1(j) D_{ij} \quad (28)$$

Specifically we have

$$u_1 = \sigma_1^{1/2} u^* \quad (29)$$

$$v_1 = \sigma_1^{1/2} v^* \quad (30)$$

where $\{u^*,\ v^*\}$ is the first Schmidt pair of $\hat{S}_{wx}$ and σ_1 is the largest singular value of $\hat{S}_{wx}$. Having computed u_1 and v_1, the vectors r_1 and s_1 that minimize J_1 for fixed u_1 and v_1 are given by

$$r_1 = \mu_1^{1/2} r^* \quad (31)$$

$$s_1 = \mu_1^{1/2} s^* \quad (32)$$

where $\{r^*,\ s^*\}$ is the first Schmidt pair of matrix $\hat{S}_{yz}$ which is defined by

$$\hat{S}_{yz} = \frac{1}{||u_1||^2 \, ||v_1||^2} \sum_{k=1}^{N_1} \sum_{l=1}^{N_2} u_1(k) v_1(l) D_{kl} \quad (33)$$

and μ_1 is the largest singular value of $\hat{S}_{yz}$. Thus we see that an optimal solution for $K = 1$ can be obtained by switching back and forth between the SVD of $\hat{S}_{wx}$ and the SVD of $\hat{S}_{yz}$. This iterative procedure is referred to as Algorithm 2.

By applying Algorithm 2 K times, a suboptimal solution to the 4-D LRA problem can be obtained. At the kth step of the procedure, the $(k-1)$th residual array

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

is constructed, to which Algorithm 2 is applied to obtain the kth quadruple $\{r_k,\ s_k,\ u_k,\ v_k\}$, and the suboptimal rank-K approximation of D is finally given by

$$D_K = \sum_{i=1}^{K} r_i \cdot s_i \cdot u_i \cdot v_i \quad (34)$$

III. APPROXIMATION OF A STILL IMAGE

As was addressed in the introduction, the approximation of a still image can be carried out via a 2D-to-4D mapping followed by an optimal LRA of the mapped 4-D array. In this section we present a simulation study of this approach. The test image, denoted by A, is a 64×64 image shown in Fig. 1a. An $8 \times 8 \times 8 \times 8$ 4-D array is obtained using the mapping

$$D(i,j,k,l) = A[i + 8(j-1), l + 8(k-1)]$$

to which the optimal 4-D LRA algorithm (Algorithm 1) is applied with $K = 4$, and 8, respectively. On the other hand, the SVD of A is computed and A is approximated by the first 1 and 2 singular images. Note that in terms of

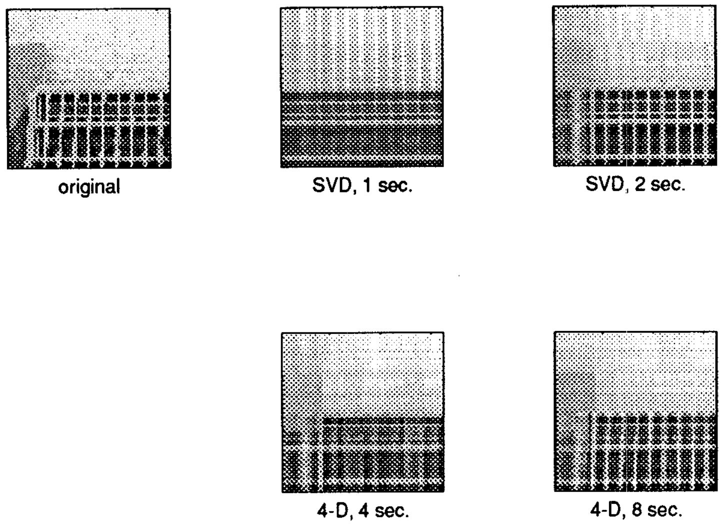

FIGURE 1

entries used, using 1 and 2 singular images in the approximation is equivalent to using $K = 4$ and 8 sections in the optimal 4-D LRA. The ratio of the total number of image pixels to the total number of entries used in the rank-K 4-D LRA is $\eta = 128/K$, where $K = 4$ and 8. The F-norm errors of the three sets of approximations are listed in Table I. It is observed that with same η the 4-D LRA is consistently smaller approximation errors compared to their SVD counterparts. Images obtained from the 4-D LRA with $K = 4$ and 8 and their SVD counterparts are shown in Figure 1.

Table I. Errors in the 4-D LRA and SVD approximations

SVD Approximation		Optimal 4-D LRA		
number of singular images	F-norm error	K	F-norm error	η
1	8.7164	4	7.8645	32
2	5.6134	8	5.5149	16

REFERENCE

[1] L. L. Scharf, "The SVD and reduced rank signal processing," *Signal Processing*, 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. 24, pp. 26-53, 1976.

[3] H. C. Andrews and C. L. Patterson, "Outer product expansions and their use in digital image processing," *IEEE TRans. Comput.,* vol. 25, pp. 140-148, 1976.

[4] N. Garguir, "Comparative performance of SVD and adaptive cosine transform in coding images," *IEEE Trans. Commu.*, vol. 27, pp. 1230-1234, Aug. 1979.

[5] D. P. O'Leary and S. Peleg, "Digital image compression by outer product expansion," *IEEE Trans. Commu.*, vol. 31, 00. 441-444, March 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 in Communications and Control Systems*, 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] G. W. Stewart, *Introduction to Matrix Computations*, New York: Academic Press, 1974.

[10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd edition, Baltimore MD: Johns Hopkins University Press, 1989.