FCO Sampling of Digital Video Using Perfect Reconstruction Filter Banks

Jelena Kovačević and Martin Vetterli

Abstract—**Three-dimensional nonseparable perfect reconstruction filter banks using three-dimensional nonseparable sampling by two, FCO, are proposed. Filter structures are derived and applied to digital video. Separation into two bands is obtained, and is shown to perform better from the perceptual point of view when compared to interlaced sequences resulting from the quincunx sampling of a progressively scanned signal in time-vertical dimensions.**

I. Introduction

Subsampling of video sequences is a common operation, and interlaced sampling has been used since the introduction of television. Both from a theoretical and practical point of view, it is of interest to investigate alternative sampling schemes having the same density. In particular, given an initial progressive video sequence (that is, a sequence sampled on a rectangular three-dimensional grid), one would like to investigate subsampling by two, and contrast the performance of a true three-dimensional subsampling pattern (FCO) with that of the traditional interlaced subsampling (or quincunx sampling in time and vertical dimensions).

We will pursue two goals: 1) to obtain a good subsampled version of the original sequence, and 2) to be able to reconstruct the original sequence perfectly from subsampled sequences. If only 1) were pursued, one would use standard low-pass filtering, possibly including motion, whereas if only 2) were desired, simple subsampling into the two cosets of the lattice would be sufficient. However, meeting both 1) and 2) requires the design of perfect reconstruction filter banks for the particular subsampling lattice. Such schemes are also known under the name of subband coding, and are well studied in speech and image compression [1]–[6]. In an earlier work [4], we had developed filter banks specifically for the progressive-to-interlaced conversion of video, and had demonstrated perfect reconstruction systems with certain attractive features (low complexity, orthogonality, linear phase). In this paper, we extend the work to FCO sampling, and compare it to the more traditional interlaced subsampling in terms of performance.

II. Quincunx and FCO Sampling

Assume that the original sampling lattice is rectangular, i.e., $\mathcal{Z}^n$. The quincunx sampling process is represented by the quincunx sublattice

$$\Lambda_Q = \{(n_1, n_2)^t \mid n_1 + n_2 = 2k,\ n_i,\ k \in \mathcal{Z}\}. \tag{1}$$

Take its three-dimensional counterpart, that is, the face-centered orthorhombic (FCO) lattice, described by the following set of points:

$$\Lambda_{FCO} = \{(n_1, n_2, n_3)^t \mid n_1 + n_2 + n_3 = 2k,\ n_i,\ k \in \mathcal{Z}\} \tag{2}$$

Manuscript received December 20, 1991; revised July 13, 1992. This work was supported in part by the National Science Foundation under Grants ECD-88-11111 and MIP-90-14189. The associate editor coordinating the review of this paper and approving it for publication was Dr. Hsueh-Ming Hang.

J. Kovačević is with AT&T Bell Laboratories, Murray Hill, NJ 07974.

M. Vetterli is with the Department of Electrical Engineering and the Center for Telecommunications Research, Columbia University, New York, NY 10027.

IEEE Log Number 9205209.

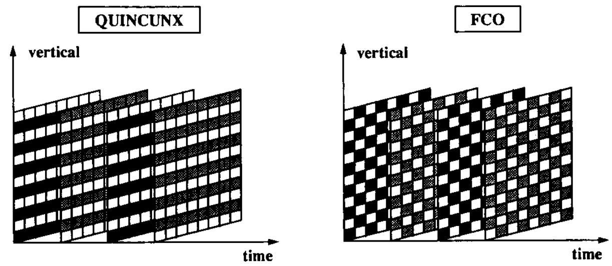


Fig. 1. Quincunx and FCO sampling performed in (time-vertical) dimensions.

as can be seen in Fig. 1. Both of these lattices can be generated by $\boldsymbol{Dk}$ where $\boldsymbol{D}$ is a nonunique matrix characterizing the sampling process and is an integer vector. For the two cases of interest, a possible choice is

$$D_Q = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad D_{FCO} \begin{pmatrix} 1 & 0 & 1 \\ -1 & -1 & 1 \\ 0 & -1 & 0 \end{pmatrix}. \tag{3}$$

It is worth noting here that the above choice of matrices was not arbitrary; rather, it was meant to ensure that their powers are diagonal matrices. This is a very useful feature if the same filter bank is to be cascaded (e.g., in the quincunx case, every other step would then lead to rectangular sampling). Thus, for example, $\boldsymbol{D}_Q^2 = 2\boldsymbol{I}$ and $\boldsymbol{D}_{FCO}^3 = 2\boldsymbol{I}$. It can be seen that, in both cases, the sampling process discards half of the samples (see Fig. 1). This sampling density can also be obtained from the determinant of the sampling matrix. Hence, the absolute value of the determinant, in both the quincunx and FCO cases equals two.

In what follows, we will need the notion of the Voronoi cell of the reciprocal lattice [7], [8]. It suffices to say at this point that it represents the set of points in the frequency domain closer to the origin than to any other point where the spectrum is replicated (due to sampling). For more details, refer to [7], [8]. It is obvious, then, that if the input signal is band-limited to that region, no overlapping of spectra will occur, and reconstruction of the signal from its samples will be possible. The Voronoi cell of the reciprocal lattice in the quincunx case is a diamond, while in the FCO case, it is a rhombic dodecahedron.

III. Energy Packing Argument for FCO Sampling

For a given subsampling factor, one approach in selecting a subsampling pattern is to maximize the energy retained in the subsampled version of the input signal (assuming that proper filtering is done to prevent aliasing).

For signals with flat spectra, any pattern will do, but for spectra with structure, different patterns will gather more or less of the original signal's energy. The amount of energy retained also depends on the particular low-pass filter chosen for a given subsampling pattern.

We will first consider subsampling by two and two-dimensional signals (sampled originally on a rectangular grid) with flat circular spectra (that is, $|S(\omega_1, \omega_2)| = 1$ for $\sqrt{\omega_1^2 + \omega_2^2} \leq \pi$). These signals have energy $\pi/4$.

For separable sampling by two, a perfect one-dimensional half-band filter is used. The support of the low-pass filter maximizing the energy retained is a rectangle (low-pass in one direction and all-pass in the other), and the energy gathered in the low band equals 60% of the input signal's energy.

If we modify the subsampling pattern to quincunx, then the support of the low-pass filter maximizing the energy will be diamond-shaped. The original signal's energy retained by this particular filter is 64%.

From a perceptual point of view, and assuming that two-dimensional signals are images, the difference is even more striking. The points of the spectrum lost in the quincunx case correspond to diagonals (to which the human eye is fairly insensitive), while in the separable case, horizontal high frequencies (above $\pi/2$) are lost, leading to blurred vertical edges (similarly for vertical high frequencies/blurred horizontal edges).

Going to three dimensions, one can now compare separable, quincunx, and FCO subsampling by two for signals with spherically uniform spectra (this model is of limited validity for motion video, but is an adequate compromise if a stationary model is needed). It can be verified that 73.7, 84.3, and 95.5% of the energy is gathered by separable, quincunx, and FCO subsampling, respectively, using the ideal filters depicted in Fig. 2.

Again, from a perceptual point of view, FCO subsampled video sequences present less objectionable artifacts (as will be seen later), and thus, both from the mean-square error and perceptual standpoints, FCO is the better way of subsampling by two.

IV. Analysis and Synthesis of Multidimensional Perfect Reconstruction Filter Banks

It is well known that a subband system, due to downsampling, is periodically shift-variant (see, for example, [2]). It is this shift variance that leads to aliased versions (or overlapping repeated spectra) of the input signal in the output. A convenient way to take care of the shift variance of such a multidimensional multirate system is to decompose both signals and filters into so-called *polyphase components,* each one corresponding to one of the cosets of the output lattice, and thus, a single-input linear periodically shift-variant system can be expressed as a multiinput linear shift-invariant system. For more details, see [5]. In both the quincunx and FCO cases, filters and signals are therefore decomposed into two polyphase components. Hence, signals at the output of the analysis bank can be represented in terms of the input signal, the forward polyphase transform $\boldsymbol{p}_f = (1, z_1^{-1})$, and the *analysis polyphase matrix* $\boldsymbol{H}_p(z)$ (that is, the matrix containing the polyphase components of the analysis filters), while the output signal can be represented in terms of the input channel signals, the *synthesis polyphase matrix* $\boldsymbol{G}_p(z)$ (that is, the matrix containing the polyphase components of the synthesis filters), and the inverse polyphase transform $\boldsymbol{p}_i = (1, z_1)$. Then, the output of the synthesis bank in the quincunx and FCO cases would be

$$Y_Q(z_1, z_2) = (z_1^{-1}\,1)\cdot \boldsymbol{G}_p(z^{\boldsymbol{D}_Q})\cdot \boldsymbol{H}_p(z^{\boldsymbol{D}_Q})\cdot \boldsymbol{x}_p(z^{\boldsymbol{D}_Q}) \tag{4}$$

$$Y_{FCO}(z_1, z_2, z_3) = (z_1^{-1}\,1)\cdot \boldsymbol{G}_p(z^{\boldsymbol{D}_{FCO}})\cdot \boldsymbol{H}_p(z^{\boldsymbol{D}_{FCO}})\cdot \boldsymbol{x}_p(z^{\boldsymbol{D}_{FCO}}) \tag{5}$$

where $z^{\boldsymbol{D}_Q} = (z_1 z_2, z_1 z_2^{-1})$, while $z^{\boldsymbol{D}_{FCO}} = (z_1 z_2^{-1}, z_2^{-1} z_3^{-1}, z_1 z_2)$. After having achieved the goal of obtaining perfect reconstruction, one might impose some other requirements on the filter bank, such as linear phase. Note that when compared to the one-dimensional case, the linear phase requirement is less constrained since linear phase in multiple dimensions means just centrosymmetry of the filter's impulse response.

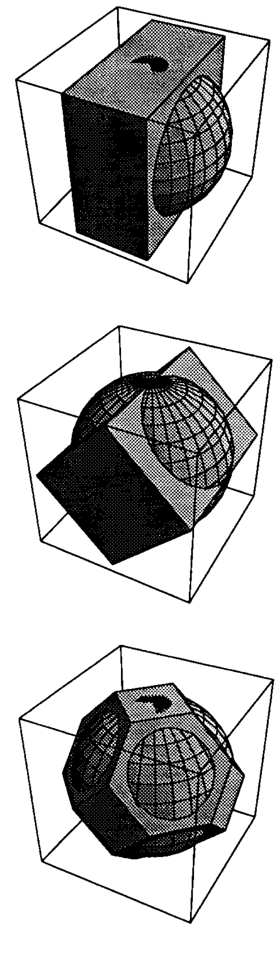


Fig. 2. (a) Separable, (b) quincunx, and (c) FCO sampling by two of three-dimensional signals with spherically uniform spectra. Shaded regions represent ideal low-pass filters.

When synthesizing filter banks, one of the most obvious approaches is to try to find cascade structures that would generate filters of the desired form, the reason being that cascade structures 1) usually have very low complexity, 2) higher order filters are easily derived from the lower order ones, and 3) the coefficients can be quantized without affecting the desired form. For building linear phase cascades, one has to make use of the linear phase testing condition given in [5]. In the next section, particular cascade structures from [5] will be used to construct three-dimensional perfect reconstruction linear phase filter sets. For orthogonal filters, see [9].

V. Three-Dimensional Perfect Reconstruction Filter Pairs

Let us now try to find a reasonably good linear phase set. To that end, a result from [5] will be used. For the sake of simplicity, it will be only summarized here. For more details, refer to [5]. It basically states that a perfect reconstruction linear phase filter set of sizes 3 and 5 in dimensions $(1, \cdots, n)$ can be generated from a perfect reconstruction linear phase set with the same sizes in dimensions $(1, \cdots, n-1)$. Moreover, this solution is general for linear phase filters of that size. Higher order filters can be obtained by cascading basic building blocks while retaining perfect reconstruction property.

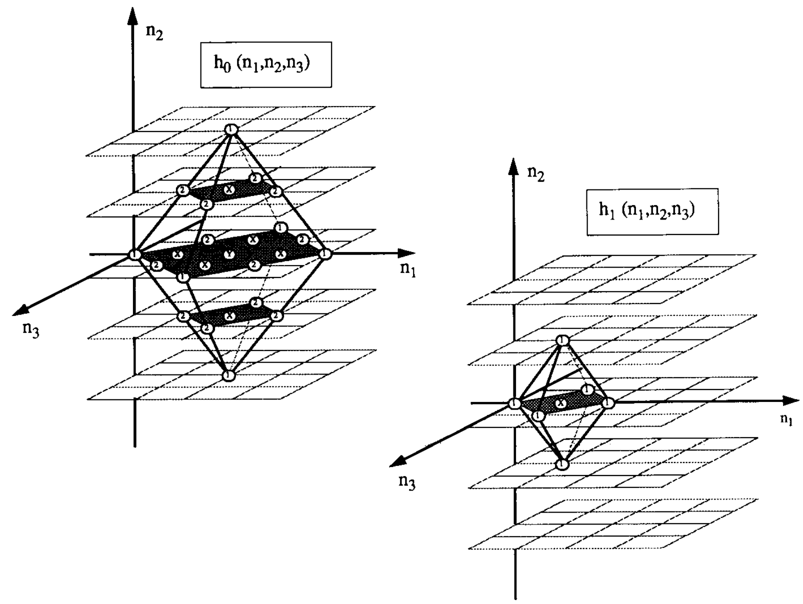


Fig. 3. Impulse responses of the linear phase three-dimensional 3/5 set, where $X = -6$ and $Y = -66$.

However, for the purposes of this paper, the smallest size filters will be used.

Thus, starting from a general one-dimensional solution, linear phase filters of sizes 3 and 5, one can obtain the two-dimensional solution generating diamond-shaped filters from [4]

$$h_0(n_1, n_2) = \begin{pmatrix} & & -1 & & \\ & -2 & 4 & -2 & \\ -1 & 4 & 28 & 4 & -1 \\ & -2 & 4 & -2 & \\ & & -1 & & \end{pmatrix}$$

$$h_1(n_1, n_2) = \begin{pmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{pmatrix}. \tag{6}$$

Using this filter set as a starting point, one can construct the three-dimensional filters using the technique from [5], yielding the following polyphase components of the two filters:

$$\begin{aligned} H_{00}\left(z^{D_{\mathrm{FCO}}}\right) &= 1 + a_2 z_1^{-2} + z_1^{-4} + \left(a + \frac{d}{a_1}\right)\left(1 + z_1^{-2}\right) \\ &\quad \cdot z_1^{-1}\left(z_2^{-1} + z_2\right) \\ &\quad + \frac{ad}{a_1} \cdot z_1^{-2}\left(z_2^{-2} + z_2^{2}\right) + \left(A + \frac{D}{a_1}\right) \\ &\quad \cdot \left(1 + z_1^{-2}\right) z_1^{-1}\left(z_3^{-1} + z_3\right) \\ &\quad + \left(\frac{aD + Ad}{a_1}\right) z_1^{-2}\left(z_2^{-1} + z_2\right)\left(z_3^{-1} + z_3\right) \\ &\quad + \frac{AD}{a_1} z_1^{-2}\left(z_3^{-2} + z_3^{2}\right) \end{aligned} \tag{7}$$

$$\begin{aligned} H_{01}\left(z^{D_{\mathrm{FCO}}}\right) &= a_1\left(1 + z_1^{-2}\right) + d \cdot z_1^{-1}\left(z_2^{-1} + z_2\right) \\ &\quad + D \cdot z_1^{-1}\left(z_3^{-1} + z_3\right) \end{aligned} \tag{8}$$

$$\begin{aligned} H_{10}\left(z^{D_{\mathrm{FCO}}}\right) &= 1 + z_1^{-2} + a \cdot z_1^{-1}\left(z_2^{-1} + z_2\right) \\ &\quad + A \cdot z_1^{-1}\left(z_3^{-1} + z_3\right) \end{aligned} \tag{9}$$

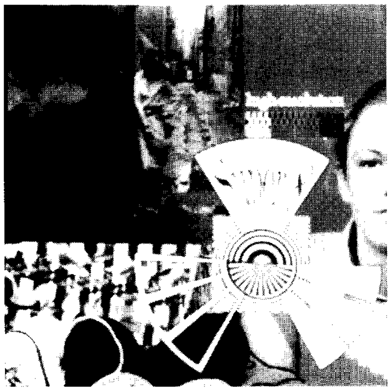
$$H_{11}\left(z^{D_{\mathrm{FCO}}}\right) = a_1 \tag{10}$$

where $z^{D_{\mathrm{FCO}}} = \left(z_1 z_2^{-1}, z_2^{-1} z_3^{-1}, z_1 z_2\right)$. As already pointed out, this solution is general. If further symmetry is required, one can impose $a = A = 1, ad/a_1 = AD/a_1 = 1$. To obtain a good low-pass filter, we will impose a zero at (π, π, π) on the filter H_0. The same will be done for the high-pass, i.e., a zero at (0,0,0) will be imposed on the filter H_1. The impulse responses of the linear phase three-dimensional 3/5 set are given in Fig. 3 with $X = -6$ and $Y = -66$.

VI. Possible Schemes and Experiments

Let us first briefly review the scheme proposed in [4]. The input progressive sequence is split into two interlaced subsequences by means of filtering and quincunx sampling over time-vertical dimensions. This kind of decomposition can be useful for compatible representation and coding since, for example, a present NTSC receiver could use just the interlaced sequence, while a more sophisticated HDTV one could perfectly recover the original progressive sequence with the help of the "deinterlacing" channel. The same scheme can be used to go from interlaced to progressively scanned sequences and back, a process that could be useful for motion estimation/compensation purposes. These steps can be combined, resulting in a two-step splitting of the original sequence. This last decomposition could be viewed as a very rough wavelet decomposition of the progressive sequence. The filters used in this scheme are the ones given in (6).

As pointed out, the aim here is to evaluate the gain of processing the video signal using a true three-dimensional scheme when a data rate reduction of two is needed. Hence, in a similar fashion as in the quincunx case, one can start with a progressive sequence and split it into two FCO sequences. Thus, the sequence is passed through the low-pass filter $h_0(n_1, n_2, n_3)$ from Fig. 3. To be able to compare the subband obtained in this fashion with the interlaced one, the two-channel scheme from the last section is used. However, due to the fact that there do not exist "FCO-interlaced" screens, the FCO subband cannot be observed directly. To that end, both the FCO and



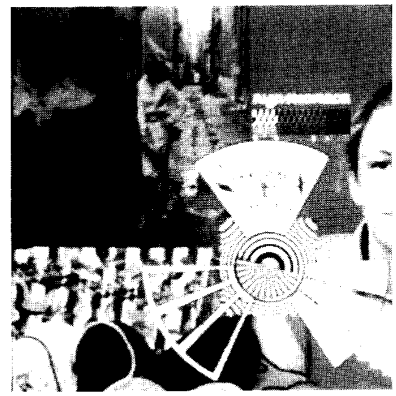
(a)

(b)

Fig. 4. Frame from the (a) original progressive sequence, and reinterpolated subchannel in the (b) quincunx scheme, (c) FCO scheme.

the interlaced subbands are reinterpolated to the original, rectangular grid. For the FCO subband, the interpolation is performed using a simple, six-neighbor averaging filter, while the interlaced subband is reinterpolated using a four-neighbor averaging filter.

Upon observing the result, it was obvious that using the FCO sampling results in a visually more pleasing sequence. Since most natural moving scenes have their energy concentrated in the low-frequency band (see Section III), this result was to be expected due to the fact that in the FCO scheme, the high band contains the high frequencies in all three dimensions, as opposed to the quincunx scheme, where the high subband contains high frequencies only in the time-vertical dimension. It is, therefore, not unexpected that the artifacts occurring with the FCO scheme are visually less disturbing than the ones observed in the quincunx scheme. Fig. 4(a)–(c) shows representative frames from the original progressive sequence, the reinterpolated quincunx sequence, and the reinterpolated FCO



(c)

Fig. 4. (*Continued*)

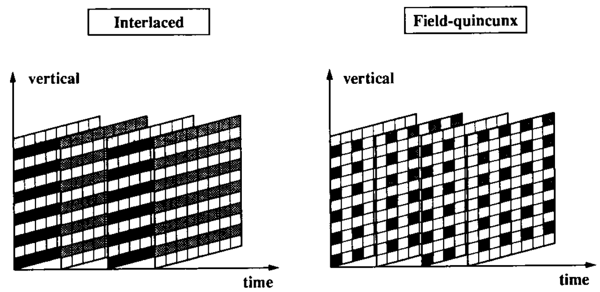


Fig. 5. Interlaced input produces a field-quincunx output when the sampling used is FCO.

sequence, respectively. It can be seen that in the FCO scheme [Fig. 4(c)], the contours are sharper [compare, for example, "Super 8" in Fig. 4(b) and (c)]. Also, the concentric circles appearing in the quincunx case [see the middle of the "turning wheel" in Fig. 4(b)] appear in the corners in the FCO case. When observing the sequence, it can also be noted that there is some "line crawling" in the background of the natural scene (upper left-hand side of the frame). This disturbing effect is completely eliminated using FCO sampling.

In the above splitting, the input sequence is progressive. An interesting question is: What do we obtain if an interlaced sequence is sampled with the FCO lattice? The answer turns out to be a field-quincunx sequence, that is, a sequence that could serve as one of the MUSE signals (MUSE is the NHK scheme developed for HDTV satellite broadcase [10]). Fig. 5 shows the input, the interlaced sequence, and the obtained subsequence.

VII. Conclusion

In this paper, a scheme has been proposed as an alternative to the quincunx scheme, which achieves a data reduction of two by interlacing. It uses the only true nonseparable sampling by two in three dimensions—FCO, and both from an energy-packing and a perceptual point of view, performs better than the corresponding two-dimensional splitting—quincunx.

REFERENCES

[1] R. Ansari, H. Gaggioni, and D. J. LeGall, "HDTV coding using a nonrectangular subband decomposition," in *Proc. SPIE Conf. Visual Commun. Image Processing.*, (Cambridge, MA), Nov. 1988, pp. 821–824.

[2] M. Vetterli, "Multidimensional subband coding: Some theory and algorithms," *Signal Processing.*, vol. 6, pp. 97–112, Feb. 1984.

[3] F. Bosveld, R. L. Lagendijk, and J. Biemond, "Three-dimensional subband decompositions for hierarchical video coding," in *Proc. SPIE Conf. Visual Commun. Image Processing,* (Boston, MA), Nov. 1991.

[4] M. Vetterli, J. Kovačević, and D. J. LeGall, "Perfect reconstruction filter banks for HDTV representation and coding," *Image Commun.*, vol. 2, pp. 349–364, Oct. 1990.

[5] J. Kovačević and M. Vetterli, "Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for $\mathcal{R}^n$," *IEEE Trans. Inform. Theory,* (Special Issue on Wavelet Transforms and Multiresolution Signal Analysis), vol. 38, pp. 533–555, Mar. 1992.

[6] E. Viscito and J. P. Allebach, "The analysis and design of multidimensional FIR perfect reconstruction filter banks for arbitrary sampling lattices," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 29–42, Jan. 1991.

[7] J. W. Cassels, *An Introduction to the Geometry of Numbers.* Berlin: Springer-Verlag, 1971.

[8] E. Dubois, "The sampling and reconstruction of time-varying imagery with application in video systems," *Proc. IEEE,* vol. 73, pp. 502–522, Apr. 1985.

[9] J. Kovačević and M. Vetterli, "Design of multidimensional nonseparable regular filter banks and wavelets," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Processing* (San Francisco, CA), Mar. 1992, pp. IV:389–IV:392.

[10] Y. Ninomiya, Y. Ohtsuka, Y. Izumi, S. Gohshi, and Y. Iwadate, "An HDTV broadcast system utilizing a bandwith compression technique—MUSE," *IEEE Trans. Broadcast.*, vol. BC-33, pp. 130–155, Dec. 1987.

An Efficient Method for Transposing Large Matrices and Its Application to Separable Processing of Two-Dimensional Signals

Michael R. Portnoff

***Abstract*—Our goal is to transpose an arbitrary matrix when the total number of matrix elements is too large to store them all in random-access memory. This problem is often a computational bottleneck in large computed-imaging problems. We derive a simple algorithm for obtaining the transposed matrix using only two read/write passes over the data. This algorithm is efficient for a wide range of practical problems. We observe that the first step of the algorithm reorders the data in a form that permits efficient access to the data *either by row or by column*. Thus if the only reason for constructing the transpose is to provide efficient access to the data for processing along the "slow" dimension of a two-dimensional data set, the matrix transpose can be eliminated simply by storing the data in this intermediate form. Furthermore, this reordering can be performed *in place* with a single read/write pass over the data.**

I. INTRODUCTION

Transposing a large matrix on a computer is straight forward so long as there is sufficient random access memory (RAM) to contain the entire matrix. If there is sufficient RAM to contain a copy of the matrix *and its transpose*, then the transpose is constructed, trivially, by copying each element of the matrix to its new location in the transpose. If there is only enough RAM to contain the matrix itself, then a square matrix can be transposed in place, simply, by swapping its elements about the main diagonal, and a rectangular matrix can be transposed in place with a less simple algorithm [1].

Manuscript received May 18, 1992; revised July 30, 1992. This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405–Eng-48. The associate editor coordinating the review of this paper and approving it for publication was Dr. Neal Gallagher.

The author is with Lawrence Livermore National Laboratory, Livermore, CA 94550.

IEEE Log Number 9205205.

Our goal is to transpose an arbitrary $M \times N$ matrix $\boldsymbol{A}$ when the number of elements in $\boldsymbol{A}$ is too large to store them all, simultaneously, in RAM. The widely used algorithms for solving this problem are recursive in nature and based on a *perfect shuffle* [2], its inverse, a *perfect deal* [2], or Eklundh's *divide-and-conquer strategy* [3]–[5]. Indeed, these are the methods of choice when only a small amount of RAM is available. Each of these approaches requires that at least one dimension of the matrix be an integer power of 2 (or a highly composite number) and the last requires that the matrix be square. For M or N a power of 2, the perfect shuffle or perfect deal method requires negligible RAM and makes $\log_2 M$ or $\log_2 N$ read/write passes over the data. None of these passes is in place. For $M = N$ a power of 2, Eklundh's method requires that a minimum of two rows of $\boldsymbol{A}$ be stored in RAM and will transpose $\boldsymbol{A}$ in $\log_2 N$ in-place read/write passes. If more RAM is available, so that 2^ρ rows of $\boldsymbol{A}$ (ρ an integer) can be stored simultaneously in RAM, then the number of passes for Eklundh's algorithm can be reduced to the smallest integer greater than $(\log_2 N)/\rho$ [4].

Our method is not recursive. It will transpose an *arbitrary-size rectangular matrix* in two read/write passes and the first pass can process the data in place. Moreover, during each pass, each matrix element is moved only once and the indexing is extremely simple. We developed this method with the premise that "modest" amounts of RAM are available on contemporary computers. To be efficient, our method will require sufficient RAM to store on the order of ten to a few hundred rows of $\boldsymbol{A}$. We will show that the amount of RAM required is proportional to the *square root* of the number of elements in the matrix.

II. METHOD

Assume that the elements of $\boldsymbol{A}$ are stored row-wise on disk and that there is sufficient RAM to store, simultaneously, p rows of $\boldsymbol{A} (p \geq 2)$. Imagine grouping the rows of $\boldsymbol{A}$ into m submatrices, $\boldsymbol{A}_i$ the first $m-1$ of which are $p \times N$ and the last is $r \times N$ where $0 < r \leq p$. Thus,

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \\ \vdots \\ \boldsymbol{A}_m \end{bmatrix}. \tag{1}$$

We will now transpose $\boldsymbol{A}$ in two steps.

Step One: We read, successively, each $\boldsymbol{A}_i$ from the disk, transpose it and write $\boldsymbol{A}_i^T$ back to the disk, either to a temporary file, or overwriting the original data, as desired. The resulting disk file will contain the *sequence of submatrices*:

$$\left\{ \boldsymbol{A}_1^T, \boldsymbol{A}_2^T, \cdots, \boldsymbol{A}_m^T \right\} \tag{2}$$

where each individual $\boldsymbol{A}_i^T$ is stored row wise.[1] Notice that for the trivial case of sufficient memory to store the entire matrix $\boldsymbol{A}$ (i.e., $p = M$), Step One, alone, produces the desired transpose.

[1] The file contains a sequence of submatrices rather than a single rectangular matrix because the last submatrix, $\boldsymbol{A}_m^T$, will, in general, have dimensions $N \times r (0 < r \leq p)$ rather than $N \times p$.