A Generalized Output Pruning Algorithm for Matrix-Vector Multiplication and Its Application to Compute Pruning Discrete Cosine Transform

Yuh-Ming Huang, Ja-Ling Wu, and Chi-Lun Chang

***Abstract*—In this correspondence, a generalized output pruning algorithm for matrix-vector multiplication is proposed. It is shown that for a given decomposition of the matrix of the transform kernel and the pruning pattern, the unnecessary operations for computing an output pruning discrete cosine transform (DCT) can be eliminated thoroughly by using the proposed algorithm**

I. Introduction

Recently, a lot of one-dimensional (1-D) and two-dimensional (2-D) fast pruning DCT algorithms for computing only the lower frequency components have been proposed in [1]–[3]. However, to the best of our knowledge, no known generalized pruning method can be directly applied to any orthogonal discrete transform (ODT), such as the DCT, the discrete Fourier transform (DFT), the discrete Hartley transform (DHT), etc. In this correspondence, a generalized output pruning algorithm for computing matrix-vector multiplication of any order is presented. It is shown that for a given decomposition of the matrix, the unnecessary operations can be eliminated thoroughly. An efficient pruning DCT algorithm can then be derived based on the prescribed pruning algorithm. Of course, the applicability of the proposed output pruning algorithm is not limited to the DCT; actually, it can be applied to all well-known discrete orthogonal transforms, such as the DFT, the DHT, and the discrete sine transform (DST). However, in this work, the pruning DCT algorithm is our only focus.

II. Generalized Output Pruning Algorithm for Matrix-Vector Multiplication

Consider the operation of a general matrix-vector multiplication of order N, say, $D_N = A_{N\times N} \times B_N$, and assume only partial multiplication outputs $D_N[j]$ (where $D_N[j]$ is the jth entry of the vector D_N, $1 \le j \le N$) are required. It follows that we can speed up the aforecited computation by pruning the unnecessary operations.

To reduce the computational complexity, we decompose the matrix $A_{N\times N}$ into a product of a sequence of more-sparse matrices of the same order, that is, $A_{N\times N} = \Pi_{i=0}^{k-1} C_{N\times N}^{i}$. By the associative property of matrix-vector multiplication, D_N can be computed recursively as

$$\begin{cases} B_N^0 = B_N \\ B_N^i = C_{N\times N}^{k-i} \times B_N^{i-1}, \qquad 1 \le i \le k. \\ D_N = B_N^k \end{cases} \tag{1}$$

Since there are k stages of matrix-vector multiplication of order N in (1), no matter what kind of output pruning pattern is, $k \times N$ bits are required to record whether each $B_N^{i,j}$ has to be computed or not, where $B_N^{i,j}$ is the inner product of the jth row vector of $C_{N\times N}^{k-i}$ and the output vector B_N^{i-1} of the previous stage.

Manuscript received January 19, 1999; revised June 15, 1999. The associate editor coordinating the review of this paper and approving it for publication was Dr. Xiang-Gen Xia.

Y.-M. Huang is with the Department of Information Engineering, National Chi-Nan University, Puli, Taiwan, R.O.C.

J.-L. Wu and C.-L. Chang are with Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.

Publisher Item Identifier S 1053-587X(00)00970-3.

In this section, a more efficient algorithm for computing output-pruning matrix-vector multiplication is presented. In this algorithm, only $\lceil \log(k+1) \rceil \times N$ bits are required to record whether the partial results $B_N^{i,j}$ has to be computed or not. In other words, we need an array, say, M of order N with each entry of $\lceil \log(k+1) \rceil$ bits in size, to record which operations are required or unnecessary.

If the computation of $D_N[j]$ is necessary, then initially, let $M[j] = 0$; otherwise, let $M[j] = 255$ or a large integer. The final value of each entry of M will evolve gradually through the computation of $C_{N\times N}^{0}$ to that of the $C_{N\times N}^{k-1}$ and will be precomputed and stored with respect to the characteristics of the concerned matrix $C_{N\times N}^{i}$ described as follows.

A. Encoding Processes

Let T be a control or threshold parameter and its value is set to be zero initially.

1) If $C_{N\times N}^{i}$ is a permutation matrix, that is, for any vector V_N of order N, the result of the matrix-vector multiplication $C_{N\times N}^{i} \times V_N$ is just a position swapping of V_N. In this case, the entries of M are unchanged in value but permuted according to the inverse permutation matrix $(C_{N\times N}^{i})^{-1}$., and the value of T is unchanged.
2) $C_{N\times N}^{i}$ is a diagonal matrix, that is, all the entries of $C_{N\times N}^{i}$ are equal to zero except the diagonal components. In this case, the values of each entry of M and T will be unchanged.
3) $C_{N\times N}^{i}$ is a general diagonal matrix, that is, all the diagonal components of it are not equal to zero, and no constraint is set to the nondiagonal components. In this case, the value of T will be increased by one. The value of T (which is denoted as T_t) is used as a threshold for indicating the fact that in the matrix-vector multiplication stage, say, $C_{N\times N}^{i} \times B_N^{k-i-1} = B_N^{k-i}$, some output entry $B_N^{k-i,s}$ is unnecessary (i.e. $M[s] = 255$), whereas the entry $B_N^{k-i-1,s}$ of the input vector B_N^{k-i-1} is required to compute some output entry $B_N^{k-i,r}$. That is, the sth input entry $B_N^{k-i-1,s}$ has to be computed correctly before dealing with the matrix-vector multiplication $C_{N\times N}^{i} \times B_N^{k-i-1}$, but after that, the sth output entry $B_N^{k-i,s}$ is of no use for later stages. In other words, if $M[r] < T_t$, and $M[s] = 255$, then we set $M[s] = T_t$.
4) $C_{N\times N}^{i}$ can be decomposed into a product of a general diagonal matrix and a permutation matrix, or vice versa. In this case, the array M will be processed by using the merged methodologies presented in 1) and 3).
5) The other matrix forms that do not belong to those of the above four types are categorized as type 5). Notice that those matrices discussed in 1)–3) are special subsets of 4). Hence, by definition, those matrices of type 5) cannot be decomposed into a product of a general diagonal matrix and a permutation matrix. Moreover, according to the following corollary, we will deduce that each matrix of type 5) is a linearly dependent matrix.

Corollary 1: If a matrix of size $N \times N$ cannot be decomposed into a product of a general diagonal matrix and a permutation matrix of the same size, then its determinant is equal to 0.

By corollary 1, we know that for any well-defined discrete transform matrix $A_{N\times N}$, which is linearly independent, it will never be categorized as a type 5) matrix.

For the sake of convenience, those sets composed of the matrices discussed in 1)–4) are, respectively, denoted by P, D, GD, and PGD.

As we have obtained the final values for each entry of M through the computation of $C_{N\times N}^{0}$ to that of the $C_{N\times N}^{k-1}$, then with the help of M, all the unnecessary operations for the computation of $C_{N\times N}^{0} \times$

TABLE I
NUMBERS OF REQUIRED MULTIPLICATIONS AND ADDITIONS FOR THE RESULTANT PRUNING DCT ALGORITHMS WITH RESPECT TO DIFFERENT MATRIX DECOMPOSITIONS AND DIFFERENT PRUNING PATTERNS

The referred matrix decompositions / No. of mults / No. of adds / Pruning Patterns	Z. Wang [1]	Winograd [6]	B. G. Lee [5]
Only first 2 outputs are required	33/126	52/207	64/188
Only first 4 outputs are required	65/204	85/298	110/279
Only first 8 outputs are required	97/288	109/364	146/351
Only first 16 outputs are required	129/372	133/425	174/411
Only first 32 outputs are required	161/450	161/481	192/463

$C^1_{N\times N}\times\cdots\times C^{k-1}_{N\times N}\times B_N$ can be eliminated thoroughly. Let the final accumulated value of T be denoted by T_f. This means, among those matrices $C^i_{N\times N}$ $0\le i\le k-1$, there are T_f matrices that belong to GD or PGD. In other words, after the matrix decomposition, the value T_f is precomputable.

Now, let us show how the unnecessary operations for the computation of $C^0_{N\times N}\times C^1_{N\times N}\times\cdots\times C^{k-1}_{N\times N}\times B_N$ can be eliminated thoroughly with the aid of M and T.

B. Decoding Process

First, let $T=T_f$. The elimination processes for unnecessary operations are deduced gradually through the computation of $C^{k-1}_{N\times N}\times B_N$ to that of the $C^0_{N\times N}\times B^{k-1}_N$.

1) If $C^i_{N\times N}\in P$, then the entries of M will be swapped according to the permutation pattern defined by $C^i_{N\times N}$.
2) If $C^i_{N\times N}\in E$, then the jth output $B^{k-i,j}_N$ needs to be computed only when $M[j]\le T$; otherwise, it can be left out for pruning the unnecessary operations.
3) If $C^i_{N\times N}\in$ GD, then the jth output $B^{k-i,j}_N$ needs to be computed only when $M[j]<T$; otherwise, it can be left out for pruning the unnecessary operations. Furthermore, the value of T will be decreased by one in this case.
4) If $C^i_{N\times N}\in$ PGD, it follows that $C^i_{N\times N}$ can be decomposed into a product of a general diagonal matrix D_g and a permutation matrix P_p (or vice versa). Then, the entries of M will be permuted according to the P_p first, and the jth output $B^{k-i,j}_N$ has to be computed only when $M[j]<T$; otherwise, it can be left out for pruning. Of course, the value of T will also be decreased by one in this case.

The above statements described the detailed procedures of the proposed pruning algorithm. Because the final value of T, i.e., T_f, will not be greater than k. Only $\lceil\log(k+1)\rceil\times N$ bits are required to record the evolution process of M.

Corollary 2: Let $A_{N\times N}(=\Pi^{k-1}_{i=0}\ C^i_{N\times N})$ be a linearly independent matrix. From the proposed algorithm, it can be deduced that in the computation of $A_{N\times N}\times B_N$, $B_N[j]$ is necessary only when $M[j]\le T_f$.

Lemma 1: Let $A_{N\times N}(=\Pi^{k-1}_{i=0}\ C^i_{N\times N})$ be a linearly independent matrix. Then, all the unnecessary operations can be eliminated thoroughly by the above proposed pruning algorithm.

From Lemma 1, for a linearly independent matrix $A_{N\times N}$, we know that the unnecessary operations can be eliminated thoroughly when only the partial outputs of the matrix-vector multiplication $A_{N\times N}\times B_N$ are required. However, for a special pruning pattern, does there exist another scheme that can be used to further reduce the number of required operations. In the next corollary, we show that the number of required operations cannot be reduced by just utilizing the permutation technique. Moreover, for $C^i_{N\times N}\in$ PGD, the gain of pruning will not be changed even if we apply a different decomposition to $C^i_{N\times N}$. his will be shown in Lemma 2.

Corollary 3: Let $A_{N\times N}$ be a linearly independent matrix and P_C be a permutation matrix. For any pruning pattern, on computing of the following expressions $A_{N\times N}\times B_N$, $(A_{N\times N}P_c)\,(P_c^{-1}B_N)$, and $P_c(P_c^{-1}A_{N\times N})B_N$, the simplification gains obtained from pruning the unnecessary operations will be the same.

Lemma 2: Let $A_{N\times N}$ be a linearly independent matrix. Based on the proposed pruning algorithm, the simplification gain will keep unchanged, even though we apply a different decomposition to the matrix $C^i_{N\times N}$ ($\in$ PGD).

Therefore, more effective decomposition of the matrix $A_{N\times N}$ is necessary if we want to obtain better simplification gain.

III. APPLICATION OF THE PROPOSED OUTPUT PRUNING ALGORITHM TO THE COMPUTATION OF PRUNING DCT

Since DCT is an orthogonal discrete transform, its transform kernel matrix must be a linearly independent matrix. That is, the pruning algorithm presented in Section II can be directly applied to derive efficient pruning DCT algorithms. Moreover, all well-known DCT algorithms (such as [4]–[6]) and pruning DCT algorithms (such as [1]–[3]) can be modeled as a matrix-vector multiplication with known decompositions of the DCT transform kernel matrix.

Since the optimism of the proposed pruning algorithm is decomposition dependent, we cannot only derive effective pruning DCT algorithms but also compare the effectiveness of matrix decomposition cor-

responding to each existing fast algorithm by checking the complexities of the so-obtained pruning algorithms.

The following data are obtained by applying the proposed output pruning algorithm to derive efficient pruning DCT algorithms, based on the matrix decompositions presented in [1], [5], and [6]. For the 1-D DCT of length 64, Table I lists the numbers of required multiplications and additions for the corresponding pruning DCT algorithms with respect to different pruning patterns.

The most well-known pruning DCT algorithm presented in [1] gives the same complexities as listed in the first column of Table I. This fact verifies the correctness and effectiveness of the proposed pruning algorithm. As for the other two algorithms (or matrix decompositions), the gain obtained from pruning is less significant. The number of pruned multiplications is larger in Winograd's approach, whereas the number of pruned additions is larger in Lee's approach. In fact, these characteristics can be observed and explained from their corresponding algorithm structures. In Winograd's DCT algorithm, the required multiplications are post-processing oriented, whereas in Lee's DCT algorithm, the most post-processing oriented operations are additions. That is, if the complexity of multiplication is the major concern, then the pruning gain will be more significant when the required multiplications of the algorithm are nearly post-processing oriented.

IV. Conclusions

In this correspondence, an index-registration technique is presented to establish an effective framework for developing efficient pruning algorithms for various ODT's. Moreover, with the aid of the proposed technique, an automatic optimal output pruning ODT program generator can be developed. This is currently under investigation.

References

[1] Z. Wang, "Pruning the fast discrete cosine transform," *IEEE Trans. Commun.*, vol. 39, pp. 640–643, May 1991.

[2] A. N. Skodras, "Fast discrete cosine transform pruning," *IEEE Trans. Signal Processing*, vol. 42, pp. 1833–1837, July 1994.

[3] C. A. Christopoulos, J. Bormans, J. Cornelis, and A. N. Skodras, "The vector-radix fast cosine transform: Pruning and complexity analysis," *Signal Process.*, vol. 43, pp. 197–205, May 1998.

[4] H. S. Hou, "A fast recursive algorithm for computing the discrete cosine transform," *IEEE Trans. Acoust., Speech Signal Processing*, vol. ASSP-35, pp. 1455–1461, Oct. 1987.

[5] B. G. Lee, "A new algorithm to compute the discrete cosine transform," *IEEE Trans. Acoust., Speech Signal Processing*, vol. ASSP-32, pp. 1243–1245, 1984.

[6] E. Feig and S. Winograd, "Fast algorithms for the discrete cosine transform," *IEEE Trans. Signal Processing*, vol. 40, pp. 2174–2193, Sept. 1992.

[7] I. W. Selesnick and C. S. Burrus, "Automatic generation of prime length FFT programs," *IEEE Trans. Signal Processing*, vol. 44, pp. 14–24, Jan. 1996.

A Novel Design Technique for Biorthogonal Filterbank Systems

Youhong Lu and Joel M. Morris

***Abstract*—In this correspondence, we present a design technique for the cosine-modulated FIR biorthogonal filter bank systems. The system achieves perfect reconstruction with a given analysis or synthesis prototype filter. In particular, if the analysis filter is a good approximation of an ideal lowpass filter, then so is the synthesis filter, and the difference is a measure of ideality of the lowpass analysis filter. The advantage of the technique is that we have more freedom in the choice of prototype filters.**

***Index Terms*—Biorthogonal, cosine modulation, filterbank, Gabor expansion, perfect reconstruction condition.**

I. Introduction

Multirate analysis and the synthesis filter systems are useful in signal analysis and representation [1]. There are many techniques for this kind of system in which the system is designed to satisfy the perfect reconstruction property, for example, the halfband filter-based technique, the power complementary-based technique, the lapped lattice-based technique, and the paraunitary-based technique [1]. The most efficient techniques for implementation of this system, we believe, are the tree-structured filter bank system [1].

The cosine-modulated analysis and the synthesis filter system have been studied in depth by many researchers because the design is simpler and more realizable than that of a general filter bank system [1]–[14]. Most past and current design techniques set a fixed relationship between analysis and synthesis filter banks, for example, $h_m(k) = f_m(N-k-1)$, where $h_m(k)$ and $f_m(k)$ are mth band analysis and synthesis filters, respectively, and N is the length of the filters. The system design problem, therefore, becomes a set of analysis or synthesis filter design problems. This design usually requires us to solve a nonlinear equation, and consequently, nonlinear optimization methods have to be used.

In many applications, a set of desired analysis or synthesis filters might be required. For example, in echo cancellation based on time-frequency techniques for telecommunication systems, the analysis filters have to be designed for maximum performance, and the synthesis filters are then designed based on the designed analysis filters to maintain smallest distortion [13]; in image processing, modulated-Gaussian filters are frequently used to extract image features such as edges and textures. In this work, we mainly discuss the design of the set of synthesis filters for a given desired set of analysis filters. Since the filter bank system still holds if we exchange the set of analysis filters with the set of synthesis filters in the system based on our biorthogonal-like sequence concept [12], this is equivalent to the design of the set of analysis filters for a given desired set of synthesis filters. We will denote this filter bank system the biorthogonal filter bank.

Manuscript received October 7, 1995; revised May 12, 1999. The associate editor coordinating the review of this paper and approving it for publication was Editor-in-Chief Prof. José M. F. Moura.

Y. Lu is with the DSP Department, 3Com Corporation, Mount Prospect, IL 60056 USA.

J. M. Morris is with the Department of Computer Science and Electrical Engineering, University of Maryland-Baltimore County, Catonsville, MD 21228 USA.

Publisher Item Identifier S 1053-587X(00)00992-2.