AN IMPROVED ARCHITECTURE FOR THE ADAPTIVE DISCRETE COSINE TRANSFORM

François Martin and David R. Bull
Image Communications Group, Centre for Communications Research
University of Bristol, Queens Building, University Walk, Bristol BS8 1TR, UK
Dave.Bull@bristol.ac.uk

ABSTRACT

This paper presents a new approach to the efficient realisation of the discrete cosine transform for the specific case of interlaced image sequence coding. In such cases, the conventional approach of decomposing each frame or frame difference into 8x8 blocks is often no longer satisfactory and an adaptive architecture capable of processing either 8x8 or two 4x8 blocks is desirable. The approach described is based on the decomposition used by Madisetti, modified to maximise shared hardware resources and to exploit arithmetic redundancy using primitive operator methods. The resulting architecture is compared with alternative implementation options using an area-time metric with savings in excess of 50% having been observed.

1. INTRODUCTION

Since its definition by Ahmed, the Discrete Cosine Transform (DCT) has been widely used in many image and video signal processing systems and has been incorporated in most international standards including JPEG, H261, MPEG-1 and -2. The definition of the DCT is given below in its one dimensional (1) and two dimensional (2) forms.

$$X(k) = \sqrt{\frac{2}{N}}\varepsilon_k \sum_{m=0}^{N-1} x[m]\cos\left[(2m+1)\frac{k\pi}{2N}\right] \quad (1)$$

$$X(k,l) = \frac{2\varepsilon_k\varepsilon_l}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x[m,n]\cos\left[(2m+1)\frac{k\pi}{2N}\right]\cos\left[(2n+1)\frac{l\pi}{2N}\right] \quad (2)$$

$$\text{where } \varepsilon_k = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } k = 0 \\ 1 & \text{otherwise} \end{cases}$$

Equations (1) and (2) may also be represented in matrix form as given in equations (3) and (4) respectively.

$$\mathbf{X} = \mathbf{C}_N\mathbf{x} \quad (3)$$

$$\mathbf{X} = \mathbf{C}_N\mathbf{x}\mathbf{C}_N{}^T \quad (4)$$

$$\text{where } (\mathbf{C}_N)_{k,m} = \sqrt{\frac{2}{N}}\varepsilon_k \cos\left[(2m+1)\frac{k\pi}{2N}\right]$$

In the case of the 2D-DCT, the calculation complexity is $O(N^4)$. However, using equation (4) it can be readily shown that the 2D-DCT is separable. This implies that the 2D calculation can be performed by applying an N point 1D-DCT to the rows followed by an N point 1D-DCT on the resulting columns. Although the number of operations is reduced from $O(N^4)$ to $O(N^3)$ in the separable case, the 2D-DCT remains computationally intensive. For this reason, many algorithms have been developed to reduce its implementation complexity.

Early fast DCT algorithms were FFT based, but these are no longer popular due to their requirement for complex multiplications and additions. Alternative algorithms have been proposed that decompose the DCT itself; some of these decompose the DCT into lower order operations, while others rely on cyclic correlated structures [1] or on the DFT [2]. Other methods take into account the unitary property of the $\mathbf{C}_N$ matrix which allows it to be factorised into products of relatively sparse matrices [3]. Recursive methods have also been reported as alternative ways to calculate the DCT. Examples include a recursive form of equation (1) [4] and a recursive form of the matrix representation (equations (3) and (4)) [5]. For all such algorithms, the complexity is typically $O(N^2 \log_2 N)$. Many algorithms have been developed to exploit the properties of VLSI implementation [6]. These include the use of distributed arithmetic [7] and, more recently, systolic arrays [8]. The latter offer the possibility of high through-put rates especially for HDTV applications.

This paper presents a new approach to the efficient realisation of the 1-D DCT for the specific case of an interlaced image sequence. MPEG 2 for example, has an option for an interlaced field mode of operation. In such cases the conventional approach of decomposing each frame or frame difference into 8x8 blocks is often no longer satisfactory and an adaptive architecture is necessary. Section 2 of the paper justifies the need for such an

0-7803-3073-0/96/$5.00 ©1996 IEEE

adaptive architecture and section 3 presents a new and efficient solution to this problem based a modified form of Madisetti's algorithm [9] in conjunction with the use of primitive operator techniques [10].

2. JUSTIFICATION FOR AN ADAPTIVE DCT

In the case of an interlaced image sequence, if motion occurs between the scanning of odd and even lines, large valued spurious DCT coefficients may be created at high frequencies. These will be coarsely quantised during compression, reducing the quality of the decoded image sequence. This is demonstrated in figures 1 to 4 where an artificial 16 pixel shift has been introduced between even and odd lines of the block. In such situations it would be desirable to have the option of computing 4x8 DCTs on odd and even lines independently for comparison with the conventional 8x8 transform. The solution yielding the best reconstructed image (according to some predefined performance criterion) would then be selected for the block in question. Although desirable, this approach has the disadvantage that the hardware must simultaneously provide both 4x8 and 8x8 transforms. The algorithm and architecture presented in this paper represents an efficient method of performing this task.

3. A BLOCK ADAPTIVE ARCHITECTURE

The computational complexity of the DCT can be reduced by applying a decomposition where the calculation is divided on the basis of its even and odd rows (5):

$$\begin{bmatrix} X(0) \\ X(2) \\ X(4) \\ X(6) \end{bmatrix} = \begin{bmatrix} a_4 & a_4 & a_4 & a_4 \\ a_2 & a_6 & -a_6 & -a_2 \\ a_4 & -a_4 & -a_4 & a_4 \\ a_6 & -a_2 & a_2 & -a_6 \end{bmatrix} \begin{bmatrix} x[0]+x[7] \\ x[1]+x[6] \\ x[2]+x[5] \\ x[3]+x[4] \end{bmatrix} \text{ and}$$

$$\begin{bmatrix} X(1) \\ X(3) \\ X(5) \\ X(7) \end{bmatrix} = \begin{bmatrix} a_1 & a_3 & a_5 & a_7 \\ a_3 & -a_7 & -a_1 & -a_5 \\ a_5 & -a_1 & a_7 & a_3 \\ a_7 & -a_5 & a_3 & -a_1 \end{bmatrix} \begin{bmatrix} x[0]-x[7] \\ x[1]-x[6] \\ x[2]-x[5] \\ x[3]-x[4] \end{bmatrix}$$

$$\text{where } a_i = \cos\left(i\frac{\pi}{16}\right). \tag{5}$$

All columns of each matrix in (5) have the same set of coefficient magnitudes (a_2, a_4, a_6 for the first matrix and a_1, a_3, a_5, a_7 for the second). Thus, for each set of inputs, all operations of the form: $a_i(x[n]+x[7-n])$ and $a_i(x[n]-x[7-n])$ may be performed using shared hardware.

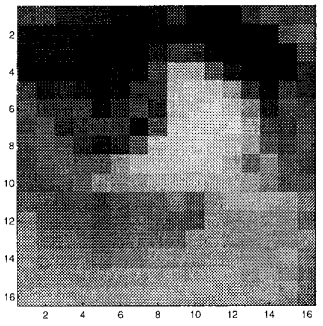

Figure 1: correlated 16x16 image block

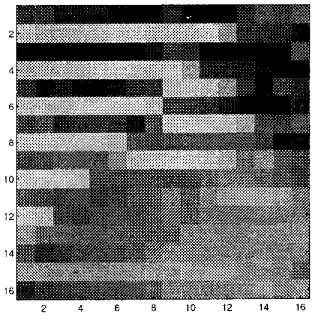

Figure 3: 16x16 image block with a shift of 16 pixels between even and odd lines.

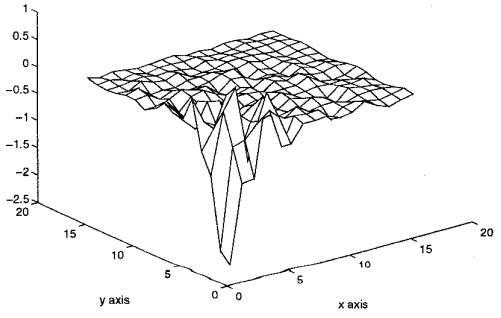

Figure 2: 16x16 DCT corresponding to figure 1.

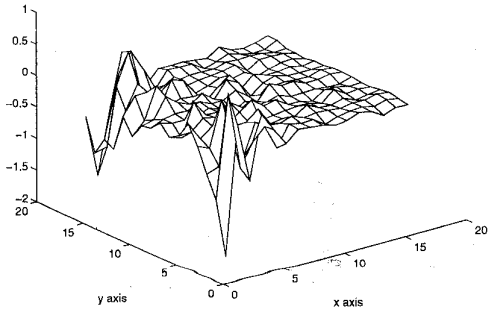

Figure 4: 16x16 DCT corresponding to figure 3.

and the result accumulated (possibly after reordering and sign changing) to produce the outputs $X(0),\cdots,X(7)$. Madisetti [9] proposed an implementation for the DCT and the inverse DCT based on this decomposition and demonstrated its use in a 100 MHz 2-D 8x8 DCT/IDCT processor suitable for HDTV applications.

To facilitate a block-adaptive structure, the DCT algorithm must be modified to efficiently produce 4 point transforms on both the even and odd components of an 8 point input vector as well as the original 8 point DCT. The 4 point DCTs required are thus:

$$\begin{bmatrix} X_e(0) \\ X_e(1) \\ X_e(2) \\ X_e(3) \end{bmatrix} = \begin{bmatrix} a_4 & a_4 & a_4 & a_4 \\ a_2 & a_6 & -a_6 & -a_2 \\ a_4 & -a_4 & -a_4 & a_4 \\ a_6 & -a_2 & a_2 & -a_6 \end{bmatrix} \begin{bmatrix} x[0] \\ x[2] \\ x[4] \\ x[6] \end{bmatrix} \text{ and}$$

$$\begin{bmatrix} X_o(0) \\ X_o(1) \\ X_o(2) \\ X_o(3) \end{bmatrix} = \begin{bmatrix} a_4 & a_4 & a_4 & a_4 \\ a_2 & a_6 & -a_6 & -a_2 \\ a_4 & -a_4 & -a_4 & a_4 \\ a_6 & -a_2 & a_2 & -a_6 \end{bmatrix} \begin{bmatrix} x[1] \\ x[3] \\ x[5] \\ x[7] \end{bmatrix} \quad (6)$$

where $\mathbf{X}_e$ is the DCT of the even rows and $\mathbf{X}_o$ is the DCT of the odd rows of the input vector **x**. Using an approach similar to that of Madisetti, the 8-point DCT can be readily extended to efficiently implement the adaptive structure. For example, the DCT of the even components in **x** can be rewritten as (7):

$$\begin{bmatrix} X_e(0) \\ X_e(1) \\ X_e(2) \\ X_e(3) \end{bmatrix} = \frac{1}{2}\begin{bmatrix} a_4 & a_4 & a_4 & a_4 \\ a_2 & -a_2 & a_6 & -a_6 \\ a_4 & a_4 & -a_4 & -a_4 \\ a_6 & -a_6 & -a_2 & a_2 \end{bmatrix} \begin{bmatrix} x[0]+x[7] \\ x[1]+x[6] \\ x[2]+x[5] \\ x[3]+x[4] \end{bmatrix} + \frac{1}{2}\begin{bmatrix} a_4 & -a_4 & a_4 & -a_4 \\ a_2 & a_2 & a_6 & a_6 \\ a_4 & -a_4 & -a_4 & a_4 \\ a_6 & a_6 & -a_2 & -a_2 \end{bmatrix} \begin{bmatrix} x[0]-x[7] \\ x[1]-x[6] \\ x[2]-x[5] \\ x[3]-x[4] \end{bmatrix} \quad (7)$$

A similar decomposition can be applied to the odd rows of **x**. Comparing (5) and (7) an architecture which allows some sharing of resources between the 8x8 and 4x8 transforms is now possible. Consider for example, the computation of $X(2)$ and $X(6)$. This can achieved using the accumulator-based architecture of figure 5, where the R block performs a permutation on the data streams according to sample index and the S blocks perform a sign change. It can be shown that with alternative permutations and sign changes, the a2 and a6 products can be reused in the formation of other outputs: $X_o(1)$, $X_o(3)$, $X_e(1)$ and $X_e(3)$. Similar configurations and their reuse can be devised for other product terms resulting in the architecture of figure 6. In this figure, the B1 and B2 blocks form the required products of input samples and DCT coefficients. Although these may be implemented using conventional fixed function multipliers, further complexity savings can be made if these are replaced with primitive operator sections [10].

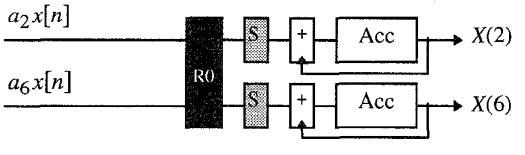

Figure 5: Calculation of $X(2)$ and $X(6)$

To compare the resulting structure with alternative implementation options, a complexity metric of the form: *Gate count* x *delay per sample* has been used. Several implementation alternatives were investigated, based on existing algorithms and/or implementations [5-7, 9]. It was found that a pipelined version of the proposed architecture incorporating primitive operator graph multipliers was the most efficient solution offering a throughput of approximately 10ns per pixel using approximately 20 000 gate equivalents. This represents a 50% reduction in complexity when compared to solutions based on Lee's[5] or McGoverns [6] algorithms.

4. CONCLUSIONS

The possibility of adaptively selecting the size of a DCT could be a major asset in the design of future video codecs especially in the case of interlaced images. This paper has proposed a structure based on a pipelined modified Madisetti algorithm incorporating primitive operator graph based multipliers. In an extensive comparative study, this was found to offer the most efficient solution to this problem offering a 50% saving over comparative.

ACKNOWLEDGEMENTS

The authors wish to thank Sony Broadcast and Professional Europe and the Centre for Communications Research at Bristol University for their support of this work.

REFERENCES

[1] Chan Y.-H., Siu W.-C., *"A Cyclic Correlated Structure for the Realisation of Discrete Cosine Transform"*, IEEE Trans. Circuits and Systems II, Vol. 39, No 2, Feb. 1992, pp 109-113.

[2] Vetterli M., Ligtenberg A., *"A Discrete Fourier Cosine Transform"*, IEEE J. Selected Areas Commun., Vol. 4, No 1, Jan. 1986, pp 49-61.

[3] Chen W.-H., Smith C.H., Fralick S.C., *"A Fast Computational Algorithm for the Discrete Cosine Transform"*, IEEE Trans. on Communications, Vol. 25, No 9, Sept. 1977, pp 1004-1009.

[4] Chau L.-P., Siu W.-C., *"Recursive Algorithm For the Discrete Cosine Transform with General Lengths"*, Electronics Letters, Vol. 30, No 3, Feb. 1994, pp 197-198.

[5] Lee P., Huang F.-Y., *"Restructured Recursive DCT and DST Algorithms"*, IEEE Trans. Signal Processing, Vol. 24, NO. 7, July 1994, pp 1600-1609.

[6] McGovern F.A., Woods R.F., Yan M., *"Novel VLSI implementation of (8x8) point 2-D DCT"*, Electronics Letters, Vol. 30, No. 8, 14 Apr. 1994, pp 624-626.

[7] White S.A., *"Application of Distributed Arithmetic to Digital Signal Processing. A Tutorial Review"*, IEEE ASSP Mag., Vol. 6, July 1989, pp 4-19.

[8] Chan Y,-T., Wang C.-L., *"New Systolic Array Implementation of the 2D Discrete Cosine Transform and its Inverse"*, IEEE Trans. Circuits and Systems for Video Tech., Vol. 5, No. 2, Apr. 1995, pp 150-157.

[9] Madisetti A., Wilson Jr. A.N., *"A 100MHz 2-D 8x8 DCT/IDCT Processor for HDTV Applications"*, IEEE Trans. Circuits and Systems for Video Tech., Vol. 5, No. 2, Apr. 1995, pp 158-165.

[10] Bull D.R., Horrocks D.H., *"Primitive operator digital filters"*, IEE Proc., Vol. 138, No. 3, June 1991, pp 401-412.

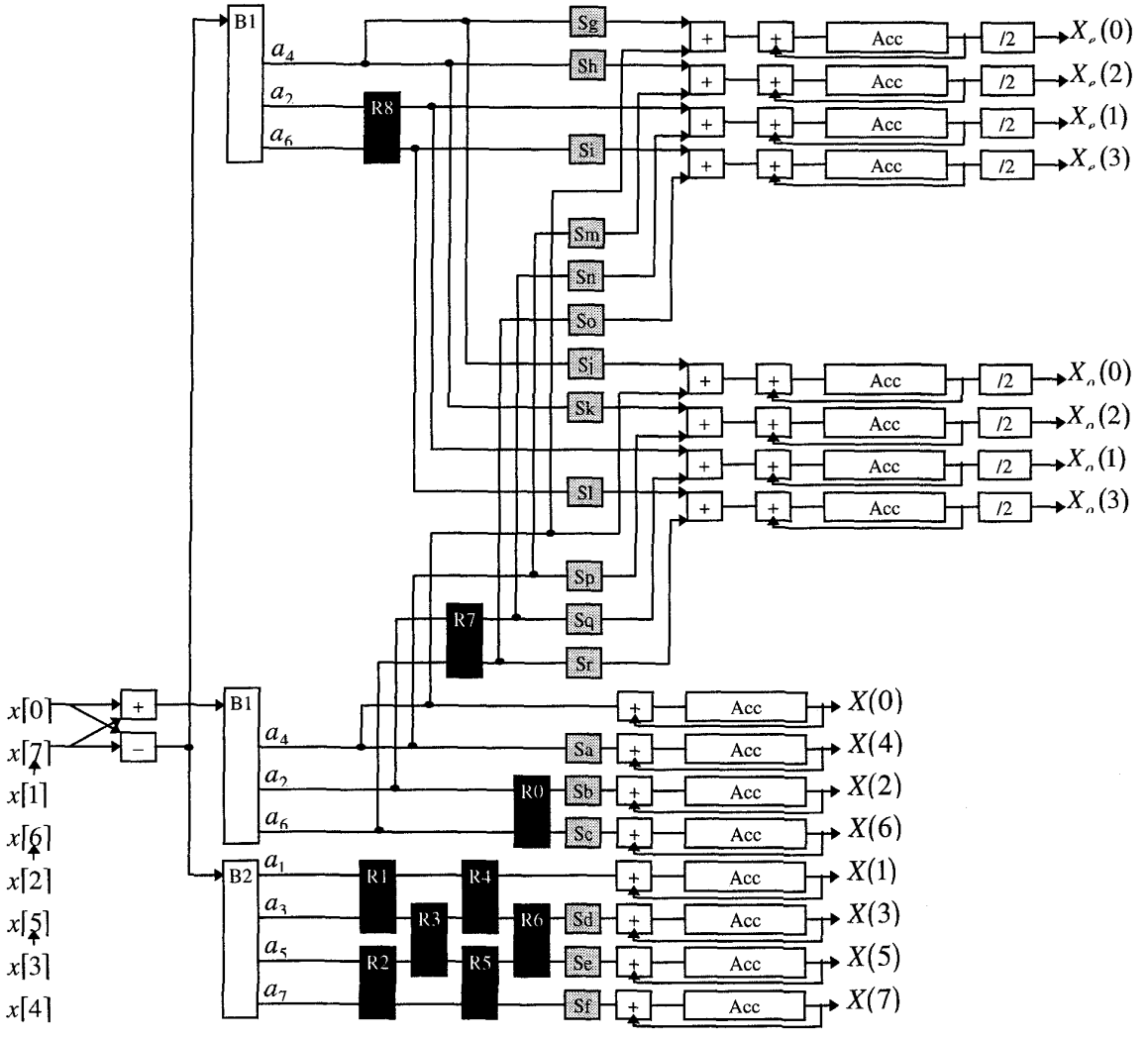

Figure 6: Adaptive DCT architecture