THREE DIMENSIONAL DCT/IDCT ARCHITECTURE

Amar Aggoun
School of Engineering and Design, Brunel University, Uxbridge, UB8 3PH, UK

ABSTRACT

In this paper, the design and development of a new fully parallel architecture for the computation of the three-dimensional discrete cosine transform (3D DCT) is presented. It can be used for the computation of either the forward or the inverse 3D DCT and is suitable for real-time processing of 2D or multi-view video codecs. The computation of the 3D DCT is carried out using the row-column-frame (RCF) approach, where a 2D DCT is computed first followed by a final 1D DCT. The proposed 3D DCT architecture comprises three identical 1D DCT modules and two sets of transpose registers. .

KEYWORDS: *Digital Circuit, Computer architecture, Image compression*

I. INTRODUCTION

The discrete cosine transform (DCT) has been applied in a wide range of applications, due to its performance being very close to the statistically optimum Karhunen–Loeve transform [1]. These applications are mainly in data, image/video, and multimedia applications [2, 3], and as a result it has been adopted as part of several compression standards [4]. This has led to the development of a large number of architectures and implementations to compute the one-dimensional (1-D) and two-dimensional (2-D) DCTs.

Owing to the rapid growth in the 3-D applications based on the 3-D DCT, there is a greater need now to develop fast algorithms and dedicated hardware solutions for the 3-D DCT for such applications. Applications of 3D DCT have been widely reported in literature [5-15]. For example, it has been applied for compression of multi-spectral scanner data based on 4×4×4 cubes composed of 4×4 blocks from each of the four spectral bands [6]. It has also been previously used in face recognition [7] and video watermarking [8]. Another well-known application is the 3D DCT of 3D blocks, displaced by motion estimation, performed on each frame. These blocks are generated from non-interlaced, High Definition Television (HDTV) frames. Also the 3D DCT has been used to exploit the high degree of temporal correlation between successive frames in a video sequence [9-11]. In recent years, the 3D DCT has found another very important application, namely compression of 3D images [12-15].

VLSI architectures for the 1D DCT and 2D DCT are very popular [16-18]. This is not the case for 3D DCT. In the last few years, hardware architectures for the computation of the 3D DCT have been proposed [19-25]. The architectures use the row-column-frame approach for the computation of the 3D DCT, hence requiring three stages of 1D DCT computation and two stages to perform matrix and volume transpositions. Row-column approach has been extensively researched as it has been fundamental to most 2D DCT architectures reported in literature. Hence the major obstacle that is needed to be solved in 3D DCT computation is the design of the volume transposition to allow the computation of the final 1D DCT. Most of the 3D DCT architectures reported in literature follow one of the two methodologies reported by the author in [19] and [20]. The architecture proposed in [19] uses three 1D DCTs which accept the input data in a serial fashion i.e. one pixel per clock cycle. The outputs from the 2D DCT are fed into an $N^2 \times N$ memory, which is shuffled to allow the correct reading for the final N-point 1D DCT. The transposition operation of the $N^2 \times N$ memory required prior to the third 1D DCT cannot be performed in the conventional manner i.e. row-column transpose. This is due to the fact that this matrix is not square and that each element of the N-point data fed to the final 1D DCT are collected every N^2 cycles. In [19], this memory is divided into N distinct $N \times N$ memories and a switching network to enable a fast and simple read/write system. The advantage of the proposed

solution is that all matrix transpose are similar and carried out using the row-column transpose. However due to the serial nature of the architecture, its main disadvantage is that it is too slow for certain applications. The architecture has a throughput performance of one 3D DCT coefficient per clock cycle.

The architecture for the computation of the 3D DCT presented in [20] improves the area-time performance and decrease the initial delay by a factor of N when compared to the architecture in [19]. This is achieved by using N 2D DCT modules, one for each of the $N \times N$ block and a final 1D DCT architecture which performs parallel computation on the N 2D DCT coefficients. Although, the architecture is fast, it requires two types of 1D DCT architecture.

The work in [21] uses two architectures for the 3D DCT as those proposed in [19] and [20] except that distributed arithmetic is used for the computation of the 1D DCT. A third architecture is also proposed but it is much slower than the architecture in [19] as it uses a single 1D DCT plus a N^3-word transpose iteratively to compute the 3D DCT. The structure in [22] uses stored product to compute the respective matrix multiplication for 4×4×4 cubes. However this will become too computationally expensive when using higher sizes and no specific details about the architecture of the 3D DCT was provided. A context aware 3D DCT/IDCT hardware solution has been discussed in [23-24]. The architecture follows the same principle as that reported in [20]. Reduction in the computation is achieved at the algorithmic level where before each 1D DCT a pre-processing function that analyzes the statistics of the input samples to each computing stage and, based on heuristic rules that exploit the energy compacting properties of the DCT transform, decides if the DCT or IDCT computation has to be done or can be avoided. The drawback of this approach is that the 3D DCT/IDCT is specific the video coding application. In [25] a hardware implementation of a 3D DCT is proposed, where the architecture is very similar to that proposed in [19].

In this paper, a new fully parallel 3D DCT/IDCT architecture using the linear systolic matrix-vector without the RAM based matrix transposition is presented. The main contribution of this paper is the new volume matrix transpose for the computation of the 3D DCT using three 1D DCTs. It is shown that the architecture is highly modular, parallel and can be used to compute both the forward and the inverse 3D DCT. Finally, it is shown that the proposed architecture enables the realisation of the 3D DCT/IDCT with a smaller area-time complexity when compared to the previously reported structures, and it also results in an extremely regular structure such that its realisation is very simple.

The rest of the paper is organized as follows. The sequential 3D DCT algorithm is presented in section 2. The overall architecture of the 3D DCT is discussed in section 3, where details of the 1D DCT and 2D DCT are described. Section 4 presents detailed description of the architecture of volume transposition including the experimental results. The conclusions and future work are discussed in section 5.

II. THE 3D DCT ALGORITHM

For a given 3D spatial data sequence $\{X_{ijk}; i, j, k = 0, 1, \ldots, N\text{-}1\}$, the 3D DCT data sequence $\{Y_{pqr}; p, q, r = 0, 1, \ldots, N\text{-}1\}$ is defined by

$$Y_{pqr} = E_p E_q E_r \sqrt{\frac{8}{N^3}} \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} X_{ijk} \cos\left[\frac{(2i+1)p\pi}{2N}\right] \cos\left[\frac{(2j+1)q\pi}{2N}\right] \cos\left[\frac{(2k+1)r\pi}{2N}\right] \quad (1)$$

where

$$E_x = \begin{cases} \frac{1}{\sqrt{2}}, & x = 0 \\ 1, & x \neq 0 \end{cases}$$

The forward and inverse transforms are merely mappings from the spatial domain to the transform domain and vice versa. The 3D DCT is a separable transform and as such, the row column frame (rcf)

decomposition can be used to evaluate equation (1). Denoting: $\cos\left[\frac{(2h+1)l\pi}{2N}\right]$ by c_{lh} and neglecting the scale factor $\sqrt{\frac{8}{N^3}}E_pE_qE_r$, the frame transform can be expressed as:

$$Y_{pqr}=\sum_{k=0}^{N-1}U_{pqk}c_{rk},\quad p,\ q,\ r=0,1,2,\ldots,N-1 \tag{2}$$

The column transform can be expressed as:

$$U_{pqk}=\sum_{j=0}^{N-1}V_{pjk}c_{qj},\quad p,\ q,\ k=0,1,2,\ldots,N-1 \tag{3}$$

and the row transform can be expressed as:

$$V_{pjk}=\sum_{i=0}^{N-1}X_{ijk}c_{pi},\quad p,\ j,\ k=0,1,2,\ldots,N-1 \tag{4}$$

In order to compute an $N\times N\times N$-point DCT (where N is even), N row transforms, N column transforms and N frame transforms need to be performed. However, by exploiting the symmetries of the cosine function, the number of multiplications can be reduced from N^2 to $N^2/2$. In this case, each row transform given by equation (4) can be written as matrix-vector multipliers via,

$$V_{pjk}=\sum_{i=0}^{N/2-1}[X_{ijk}+(-1)^pX_{(N-1-i)jk}]c_{pi} \tag{5}$$

Using a matrix notation, for N=8, equation (5) can written as

$$\begin{bmatrix}V_{0jk}\\V_{2jk}\\V_{4jk}\\V_{6jk}\end{bmatrix}=\begin{bmatrix}c_{00}&c_{01}&c_{02}&c_{03}\\c_{20}&c_{21}&c_{22}&c_{23}\\c_{40}&c_{41}&c_{42}&c_{43}\\c_{60}&c_{61}&c_{62}&c_{63}\end{bmatrix}\begin{bmatrix}X_{0jk}+X_{7jk}\\X_{1jk}+X_{6jk}\\X_{2jk}+X_{5jk}\\X_{3jk}+X_{4jk}\end{bmatrix} \tag{6}$$

$$\begin{bmatrix}V_{1jk}\\V_{3jk}\\V_{5jk}\\V_{7jk}\end{bmatrix}=\begin{bmatrix}c_{10}&c_{11}&c_{12}&c_{13}\\c_{30}&c_{31}&c_{32}&c_{33}\\c_{50}&c_{51}&c_{52}&c_{53}\\c_{70}&c_{71}&c_{72}&c_{73}\end{bmatrix}\begin{bmatrix}X_{0jk}-X_{7jk}\\X_{1jk}-X_{6jk}\\X_{2jk}-X_{5jk}\\X_{3jk}-X_{4jk}\end{bmatrix} \tag{7}$$

Equations (6) and (7) describe the computation of the even and odd coefficients, for the row transform for N=8, respectively. The computation for the second and third 1D DCTs i.e. the column and frame transforms described by equations (2) and (3) can also be computed using matrix-vector multipliers similar to that described by equation (5). Hence the row, column and frame transforms can be performed using the same architecture.

Each 1D DCT unit uses N fully parallel vector inner product (VIP) which exploits the symmetries of the cosine function, to reduce the number of multiplications from N^2 to $N^2/2$ as described by equations (6) and (7), for N = 8. The architecture for computing an 8-point DCT is depicted in figure 1 and has been reported in [15]. It consists of $N/2$ adder/subtractor cells for summing and subtracting the inputs to the 1D DCT block as required by equation (4). The pair of inputs X_{ijk} and $X_{(N\text{-}1\text{-}i)jk}$ enters the $(i+1)^{th}$ adder/subtractor cell. All the pairs of input data enter the adder/subtractor cells at the same time.

The design of the VIP units is based on a systematic design methodology using radix-2^n arithmetic which allows partitioning of the operands to n-bit digits each and hence providing the designer with more flexibility between throughput rate and hardware cost, by varying the digit-size n, the pipelining level and also the type of architecture [18].

III. 3D DCT ARCHITECTURE

The new fully parallel architecture for the computation of the 3D DCT is based on the row-column-frame decomposition technique and is shown in figure 1. The architecture uses the same 1D DCT module for all three dimensions and is divided into three main stages. Stages one and three compute the 2D DCT and 1D DCT, respectively.

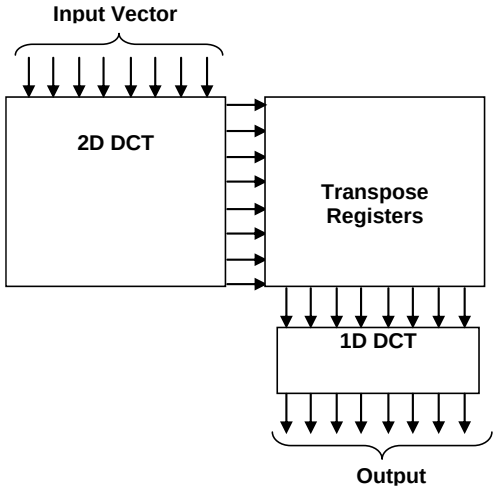

Figure 1. Block diagram of the 3D DCT/IDCT architecture

3.1. 1D DCT Architecture

The architecture for computing the 1D DCT, for N = 8, is depicted in figure 2. It is based on Step 1 of the systolic array implementation proposed by Chang and Wang [17]. It consists of N/2 adder/subtractor cells for summing and subtracting the inputs to the 1D DCT block as required by equation (5). The pair of inputs X_{ijk} and $X_{(N-1-i)jk}$ enters the $(i+1)^{th}$ adder/subtractor cell.

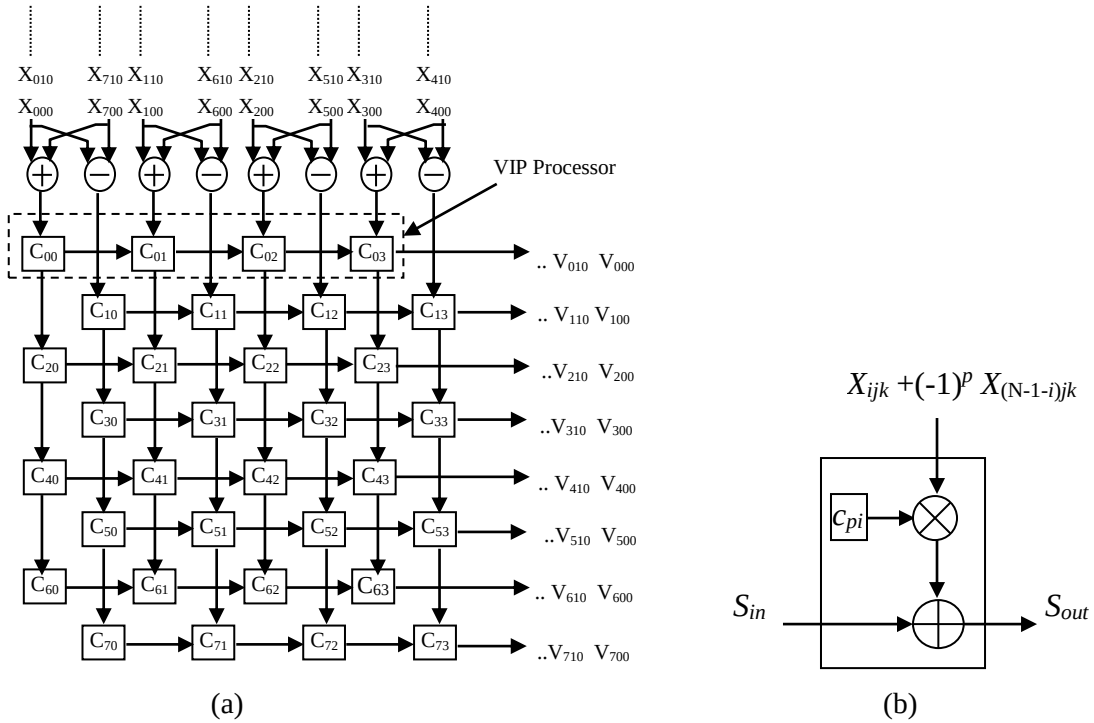

Figure 2. (a) 1D DCT architecture for N=8. (b) basic cell [18].

In the proposed architecture, all the pairs of input data enter the adder/subtractor cells at the same time. Figure 2 shows that the architecture also consists of N vector inner products, where half are used for the added pairs as described by equation (6) and the other half for the subtracted pairs as

described by equation (7). Each vector inner product consists of N/2 multiplier/accumulator cells. Each cell stores one coefficient c_{pi} in a register and evaluates one specific term over the summation in (5). The multiplications of the terms c_{pi} with the corresponding data are performed simultaneously and then the resulting products are added together in parallel. This addition is carried out using carry save arithmetic modules incorporated within the multipliers structure as reported in [18].

3.2. 2D DCT Architecture:

Stage one is achieved using the parallel 2D DCT architecture reported in [18] and shown in figure 3 for N=8. It consists of two 1D DCT modules (shown in figure 2) and a transposition matrix which is implemented using two sets of $(N^2 + N)/2$ skewed registers and the data is fed from one set to the other via N (N:1) multiplexers (see figure 3). This allows the reading of N output coefficients from the 1st 1D DCT and the feeding of N coefficients to the 2nd 1D DCT in a pipelined fashion [18].
The 1D DCT unit accepts input vectors in parallel and produces the N DCT coefficients in parallel. The N outputs from the row transform are fed into an array of skewed shift registers as shown in figure 3 to enable the reading of only one coefficient from the same output vector at any one time. This achieves the appropriate reordering of the data into the second array of skewed registers. The skewed registers are made of N shift registers with lengths varying from 1 to N, respectively.
For N=8, a 3-bit control signal is required to enable the 8:1 multiplexers to select one of the eight input words. During the first cycle of the transfer of data from the first array of skewed register to second array, the first multiplexer from the right will select its port 1 as its output. In the second cycle, the second multiplexer on the right will select port 1 as its output while the first multiplexer from the right will select port 2 as its output and so on. Hence, the control signal is connected to all multiplexers through delay elements as shown in figure 3.

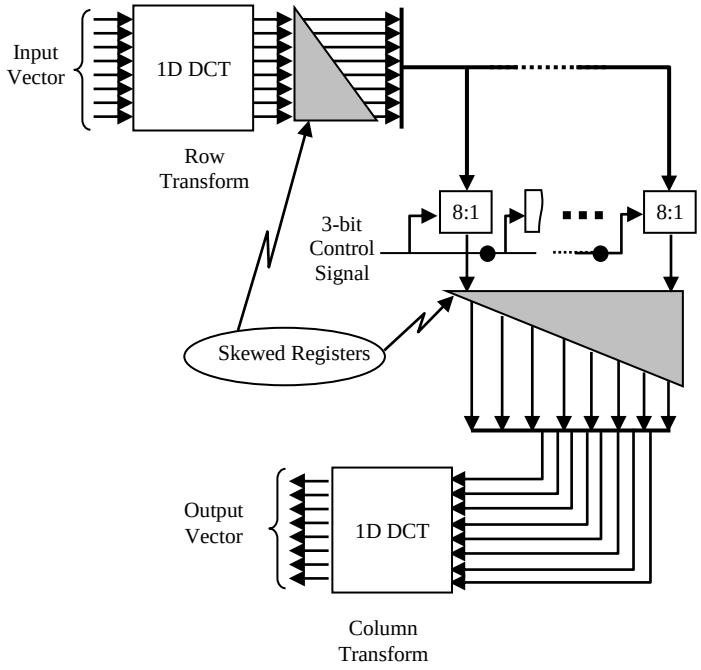

Figure 3. 2D DCT/IDCT architecture for N=8 [18]

IV. ARCHITECTURE FOR THE VOLUME TRANSPOSITION

Stage two of the proposed 3D DCT performs the volume transposition, i.e. the transposition of the 2D DCT coefficients, required to rearrange the coefficients in the correct sequence for the computation of the final 1D DCT in order to yield the desired 3D DCT coefficients. This is achieved using registers and multiplexers. Stage three computes the final 1D DCT and is based on the 1D DCT module used in the 2D DCT architecture and shown in figure 1. It accepts N samples of the 2D DCT coefficients in parallel and produces N 3D DCT coefficients in parallel in one clock cycle. The 3D DCT architecture can be used for the computation of either the forward or the inverse DCT and it possesses features of

regularity and modularity.

The transposition module acts as storage for the 3D DCT coefficients and also achieves volume transposition. However, this transposition unit is more complicated than that of the 2D DCT architecture as it handles a lot more data. This is due to the fact that the *N* parallel inputs into the final 1D DCT should consist of corresponding points in the volume data. In order to rearrange the 2D DCT coefficients to accomplish this, all the $N \times N \times N$ coefficients needs to be processed and stored appropriately. The transposition is achieved using $3N^2(N\text{-}1)/2$ registers and N (N:1) multiplexers as shown in figure 4. The $3N^2(N\text{-}1)/2$ registers are divided into two sets. The first set contains $N(N\text{-}1)/2$ skewed registers each made of N shift registers. The second set of registers is made up of N arrays of line delays.

Each array of line delays, shown in figure 5, consists of N line delays, each containing N registers except for the 1st one which contains one register only. Each of the registers in the first set (array of skewed registers) is connected to a corresponding array of line delay in the second set of registers as shown in figure 4. The outputs from the N arrays of line delays are connected to the final 1D DCT module via the N (N:1) multiplexers

For convenience, the shift registers in the array of skewed registers are termed $R_1, R_2, \ldots, R_{N\text{-}1}$, where the index specifies the length of the register (in terms of the number of N word registers). Also the arrays of line delays are termed L_0^i, L_1^i, …, L_{N-1}^i, where the lower index specifies the line delay within the array of line delays, with L_0^i representing the bottom line delay and L_{N-1}^i the top line delay. The upper index, i, specifies the arrays of line delays for i = 0,1, 2, …, N-1, where i = 0 represents the topmost array of line delays and $i = N$-1 the bottom. For N=8, the subimages which form the input data to the 3D DCT architecture are termed S_0 to S_7, after the first transpose the 1D DCT coefficient matrices achieved are termed S_0' to S_7'. After the second transpose the 2D DCT coefficient matrices are termed S_0'' to S_7''. Figure 6 shows the pixel sequence of the 2D DCT coefficient matrices S_0'' to S_7'' after the 2D DCT computation. The top 2D DCT coefficients are fed directly into the top array of line delays. The remaining 2D DCT coefficients are fed into their corresponding skewed shift registers R_1 to $R_{N\text{-}1}$ as they leave the 2D DCT module. As the skewed shift registers overflows, the data is clocked into the corresponding array of line delays. The transposition module has an initial delay of (N^2-N+1) cycles. After this delay, the data from the first row of the S_0'' matrix will be in L_7^0, the data from the first row of the S_1'' will be in L_6^0, the data from the first row of the S_2'' matrix will be in L_5^0 and so on. Thus all the data in position *'00'* of each of the 2D DCT matrices will be clocked into their corresponding multiplexers at the same time, enabling the final 1D DCT to be computed on corresponding points of the volume data as required.

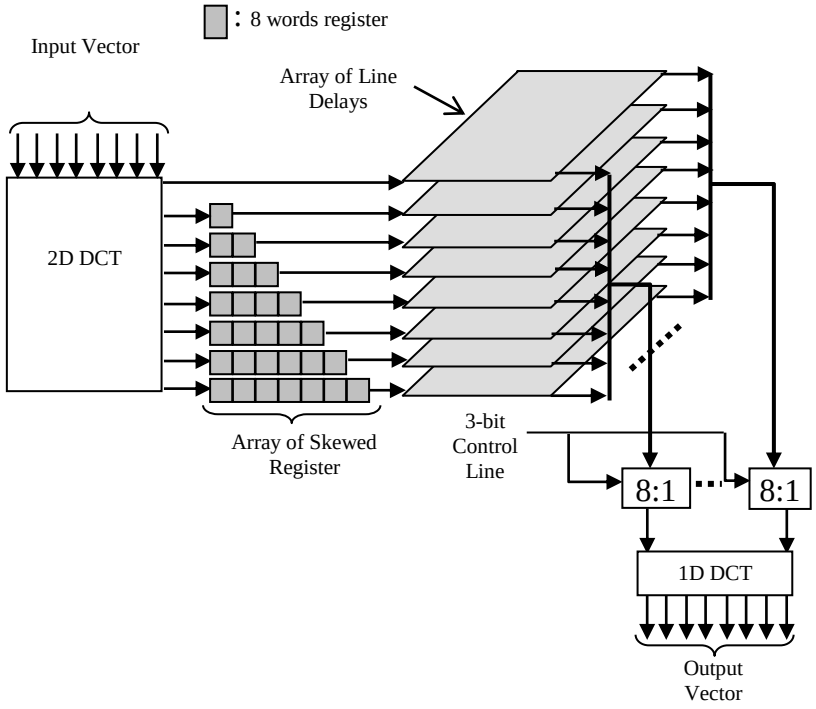

Figure 4. Architecture for the 3D DCT/IDCT (N=8).

Figure 7 shows a snapshot at that particular instant, of the coefficient sequence of the arrays of line delays, for the first, second and last arrays of line delays. As the data is outputted, they are also shifted into the next shift register, creating a chain-like movement and enabling the next data to be clocked out. The next set of data to be clocked into the multiplexers will be the data in position *'01'* of each of the 2D DCT matrices and so on.

After *N* clock cycles, all the data in L_i^0, i.e. the first row of each 2D DCT matrix, would have been processed, at the same time, the data in L_i^1 will be ready, with each line delay L_1^1 to L_7^1 holding the second row of each of the 2D DCT matrix. The same process of moving data to the 1D DCT module via the multiplexers will now initiate in L_i^1. This process continues until all the data from all the arrays of line delays are processed. After N^2 cycles the process starts all over again by processing data from L_i^0.

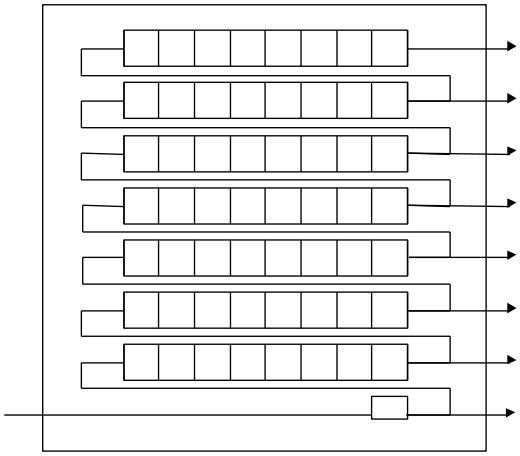

Figure 5. Array of delay lines for N=8 (R^i).

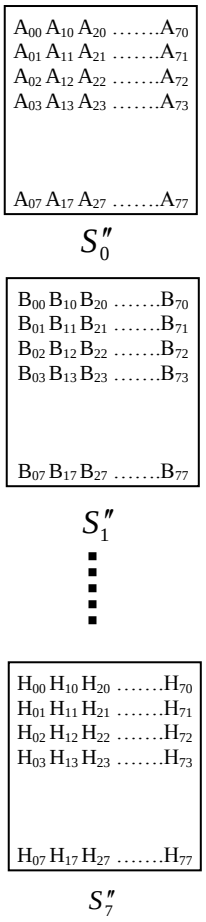

Figure 6. Data sequence of the eight 2D DCT coefficient matrices.

It can be seen from figure 7 that the data continues to flow through the arrays of line delays as data is outputted. The processing of the data from the array of line delays, *i*, starts *N* cycle after the beginning of the processing of data from the array of line delays, *i*-1. Hence the data entering the array of line delays, *i*, is delayed by *N* cycles when compared to the data entering the array of line delays, *i*-1. This delay is achieved by the array of skewed registers.

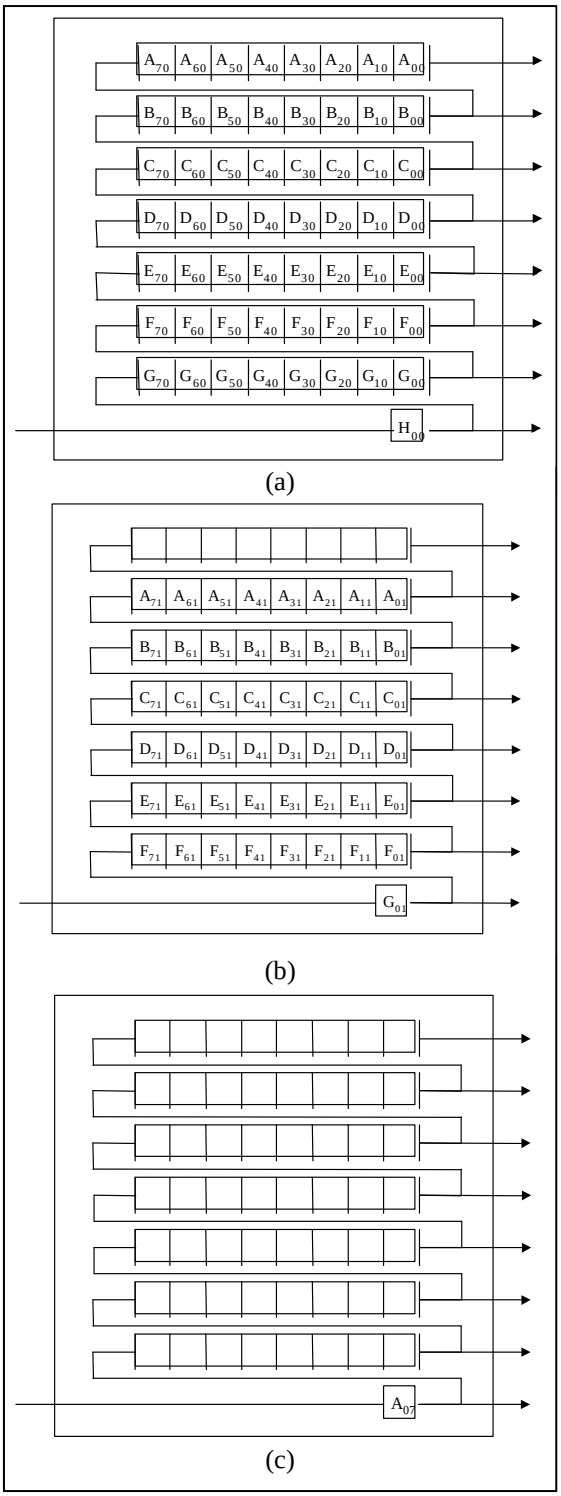

Figure 7. 2D DCT coefficient sequence through the array line delays for N=8. a) L_i^0 . b) L_i^1 . c) L_i^7 **.**

To allow the reading of the data from the different arrays of line delays in successive manner, an array of *N (N:1)* multiplexers is used. The switching between the arrays of line delays is carried out every *N* cycle. Hence, the *N* multiplexers are switched simultaneously every *N* cycles and controlled by a control line with $log_2(N)$ bits. For N=8, a 3-bit control signal is required to enable the 8:1 multiplexers to select one of the eight input words. During the first *N* cycles, the multiplexers will select port 1 as their output. At the $(N+1)^{th}$ cycle, the multiplexer will switch to port 2 as their outputs and so on. Hence, the control signal is broadcasted to all multiplexers as shown in figure 4.

The 3D DCT has been implemented using the Virtex V400 device of the Xilinx Field Programmable Gate Arrays (FPGAs). From the timing verification results of the fully parallel 3D DCT the

processing frequency achieved is 50MHz. Since this architecture computes *N* 3D DCT coefficients per clock cycle, the processing frequency per pixel is 400 MHz (for *N=8*).
Similarly, for the 3D DCT architectures reported in [19] and [20] have been implemented on the same device for comparison and processing frequency of 360MHz has been achieved. Table 1 shows the hardware cost in terms of the number of gates and the processing frequency for the three architectures. From table 1, it can be seen the proposed 3D DCT architecture outperforms previously reported architectures in terms of area-time complexity. The area-time complexity is reduced by at least 80% and 43% when compared to the architectures reported in [19] and [20], respectively.

Table 1. Comparisons of hardware requirements and processing frequency (n=8).

	Proposed 3D DCT Architecture	N×Serial Architecture [19]	Fully Parallel 3D DCT Architecture [20]
Processing Frequency	400MHz	360MHz	360MHz
No. of gates	103497	464248	163912
AT (Gates x ns)	259	1290	456

V. CONCLUSIONS AND FUTURE WORK

In this paper the design of a novel fully parallel architecture based on the row-column-frame decomposition, without RAM based memory transposition, for the computation of the 3D DCT is presented. The proposed 3D DCT architecture is suitable for real-time applications such the 2D or multi-view video coding. The architecture utilises highly parallel structures to achieve high-speed performance. Due to the highly regular and modular structure, design time is minimised, the architecture is relatively easy to implement and highly suitable for VLSI implementation. It can be seen that the new architecture not only achieves lower communication complexities, but it also achieves a much lower area-time complexity.
It is worth noting that the proposed 3D DCT architecture can be implemented using different 1D DCT blocks to the one proposed in this paper. 1D DCT architectures using techniques such as distribute arithmetic or stored product have been widely reported and most can be used as an alternative 1D DCT block in the proposed 3D DCT. An interesting investigation is to look into the power consumption of the proposed architecture in comparison to a distributed arithmetic based architecture.

REFERENCES

[1]. S. Lee, Y. Kim, and S. W. Choi, "Fast scene change detection using direct feature extraction from MPEG compressed video" *IEEE trans. Multimedia* , vol. 2, pp. 240-254, 2000.
[2]. D. LeGall., "A Video Compression Standard for Multimedia Applications*" Commun., ACM*, vol. 34 (4), pp. 46 – 58, 1991.
[3]. G. K. Wallace, "The JPEG Still Picture Compression Standard," *Commun., ACM*, vol. 34 (4), pp. 30–40, 1991.
[4]. ISO/IEC DIS 13818 – 2, *MPEG-2 Video.* Draft Int. Standard.
[5]. S. Boussakta and H. O. Alshibami, "Fast algorithm for the 3D DCT-II" IEEE Trans. On Signal Processing, Vol. 52 (4), pp. 992-1001, 2004.
[6]. S.C. Tai et al., "An adaptive 3-D discrete cosine transform coder for medical image compression", IEEE Trans. Infor. Tech. in Biomed., 4 (3) pp. 259–263, 2000.
[7]. V.K. Padmaja and B. Chandrasekhar , "Literature review of image compression effects on face recognition" International Multidisciplinary Research Journal, 2(8), pp. 17-20, 2012.
[8]. H.Y. Huang et al., "A video watermarking algorithm based on pseudo 3D DCT", IEEE CIIP'09, pp. 76-81, 2009.
[9]. M. Servais and G. De Jager, "Video Compression Using the Three Dimensional Discrete Cosine Transform," *Proc. of the South African Symposium on Communications and Signal Processing*, pp. 27-32, 1997.
[10]. J. K. Sunkara, *et. al*., "A New Video Compression Method using DCT/DWT and SPIHT based on Accordion Representation" *I.J. Image, Graphics and Signal Processing*, vol. 4, pp. 28-34, 2012.
[11]. D. A. Adjeroh, S.D. Sawant, "Error-Resilient Transmission for 3D DCT Coded Video", IEEE Transactions on Broadcasting, vol. 55 (2), pp. 178-189, 2009.

[12]. N. P. Sgouros, *et.al*., "Use of an adaptive3D DCT scheme for coding multiview stereo images", Proc. of the IEEE int. Symposium on Signal Processing and Information technology, pp. 180-185, 2005.

[13]. M. Zamarin, *et. al*., "A novel multi-view image coding scheme based on view-warping and 3D-DCT" Journal of Visual Communication and Image Representation, vol. 21(5-6), pp. 462-473, 2010.

[14]. R. Zaharia, A. Aggoun and M. McCormick: "Adaptive 3D-DCT compression algorithm for continuous parallax 3D integral imaging" *Journal of Signal Processing: Image Communications***,** vol. 17(3), pp. 231-242, 2002.

[15]. A. Aggoun, "A 3D DCT Compression Algorithm for Omnidirectional Integral Images" Proceedings of IEEE ICASSP 2006, vol. 2, pp.517 - 520, 2006.

[16]. J. S. Chiang, Y. F. Chui and T. H. Chang, "A High Throughput 2-Dimensional DCT/IDCT Architecture for Real-Time and Video System," *IEEE Proc. of Int'l Conf. on Electronic Circuits and Systems*, vol. 2, pp. 867-870, Sept. 2001.

[17]. Y. T. Chang and C. L. Wang, "New Systolic Array Implementation of the 2D Discrete Cosine Transform" *IEEE Trans. on Circuits and Systems for Video Tech*., vol. 5, no. 2, Apr. 1995.

[18]. A. Aggoun and I. Jalloh, "Two-dimensional DCT/IDCT architecture" *IEE Proceedings Computers and digital techniques*, Vol. 150(1), pp. 2-10, 2003.

[19]. I. Jollah, A. Aggoun and M. McCormick, "A 3D DCT Architecture for Compression Of Integral 3D Images" *Proceedings of IEEE workshop on Signal Processing Systems*, SiPS, pp. 238-244, 2000.

[20]. I. Jalloh and A. Aggoun, "A Parallel 3D DCT Architecture for the compression of Integral 3D Images," *Proceedings of the 8th IEEE International conference on Electronics, Circuits and Systems*, pp. 229-232, 2001.

[21]. S. Saponara, L. Fanucci and P. Terreni, "Low-power VLSI architectures for 3D discrete cosine transform (DCT)" Proceedings of the 46th IEEE International Midwest Symposium on Circuits and Systems, 2003, Vol. 3, Dec. 2003, pp. 1567-1570.

[22]. M.B. Abdelhalim and A.E. Salama, "Implementation of 3D-DCT based video encoder/decoder system" International Symposium on Signals, Circuits and Systems, 2003, Vol. 2, July 2003, pp. 389-392.

[23]. S. Saponara, and L. Fanucci. "Context-Aware Fast 3D DCT/IDCT Algorithm for Low Power Video Codec in Mobile Embedded Systems" Streaming *Day 2010*.

[24]. S. Saponara, "Real-time and low-power processing of 3D direct/inverse discrete cosine transform for low-complexity video codec" J Real-Time Image Proc., vol. 7, pp. 43–53, 2012.

[25]. Y.C. Cheng, et.al. "3D-DCT Chip Design for 3D Multi-view Video Compression" Appl. Math. Inf. Sci., Vol. 6(2S), pp. 567S-572S, 2012.

AUTHORS

A. Aggoun is a Reader in information and communication technologies in the School of Engineering and Design at Brunel University. He received the "Ingenieur d'Etat" degree in electronic engineering from Ecole Nationale Polytechnique of Algiers (ENPA) Algeria and the PhD degree in compressed video signal processing from Nottingham University, UK. He has more than 25 years experience in the development of imaging and computer vision systems. He has led and contributed to a number of research projects on 3D Imaging Technologies. He is currently the principle coordinator of the EU-funded project "3D Live Immersive Video-Audio Interactive Multimedia" (3D VIVANT, a project funded by the European Union which has made a number of advances in the field of 3D imaging technology for capture, representation, manipulation and display of 3D content.