IMPACT: International Journal of Research in Engineering & Technology (IMPACT: IJRET)
ISSN(E): 2321-8843; ISSN(P): 2347-4599
Vol. 3, Issue 6, Jun 2015, 77-80

DCT AND SYSTOLIC ARRAY IN IMAGE COMPRESSION

EKTA AGRAWAL, KUMAR MANU, RUCHI VARSHNEY & ANUPAM YADAV

Department of Electronics & Communication Engineering, Moradabad Institute of Technology, Moradabad, Uttar Pradesh, India

ABSTRACT

Systolic arrays are a family of parallel computer architectures capable of using a very large number of processors simultaneously for important computations in applications such as scientific computing and signal processing. A discrete cosine transform (DCT) expresses a sequence of finitely many data points in terms of a sum of cosine functions at different frequencies. DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry.

KEYWORDS: Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), Systolic Array, VLSI

INTRODUCTION

Systolic arrays are most widely used in parallel computing applications such as computer arithmetic and signal processing. Systolic arrays are best suited for repetitive computation. These types of computation require highly regular and parallelism architecture. In a systolic array, processing elements are called, systolic cells, which perform computation simultaneously [1].

Discrete Cosine Transform (DCT) expresses a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies. Basically, DCT is a Fourier- related transform using only real numbers [2]. The DCT helps in separate the image into parts. DCT transforms a signal or image from the spatial domain to the frequency domain. Because of the wide-spread use of DCTs, research into fast algorithms for their implementation has been rather active and also, since the DCT is computation intensive, the development of high speed hardware and real-time DCT processor design have been object of research [3].

The 1-D DCT array is constructed by using the Chebyshev polynomial to generate the transform kernel values recursively. The 2- D DCT array is based on the row-column decomposition but involves no matrix transposition problems, where the row and column transforms are evaluated similarly to the 1-D DCT [4].

2-D DCT algorithms are the most typical for image compression, as the number of applications that require higher-dimensional DCT algorithms are developing in case of higher dimensional images, videos and audio. The JPEG standard has been around since the late 1980's and has been an effective first solution to the standardization of image compression. JPEG has some very useful strategies for DCT quantization and compression. DCT was only developed for low compressions [5]. The 8 x 8 DCT block size was chosen for speed (which is less of an issue now, with the advent of faster processors) not for performance.

Like other transforms, the Discrete Cosine Transform (DCT) attempts to decorrelate the image data. After decorrelation each transform coefficient can be encoded independently without losing compression efficiency [6]. This section describes the DCT and some of its important properties.

THE ONE-DIMENSIONAL DCT

$$C(u)=\alpha(u)\sum_{n=0}^{N-1} f(x)\cos\left[\frac{\pi}{N}(n+\frac{1}{2})u\right] \qquad (1)$$

Where u=0, 1, 2, N-1

$$\alpha(u)=\begin{cases}\sqrt{\frac{1}{N}} & for \quad u=0\\ \sqrt{\frac{2}{N}} & for \quad u\neq 0\end{cases}$$

This equation represents 1-D DCT Type II. The DCT-II is probably the most commonly used form, and is often simply referred to as "the DCT". This transform is exactly equivalent (up to an overall scale factor of 2) to a DFT of 4N real inputs of even symmetry where the even-indexed elements are zero. That is, it is half of the DFT of the *4N* inputs y_n, where $y_{2n}=0$, $y_{2n+1}= x_n$ *for* $0 \leq n < N$, *and* $y_{4n-n}= y_n$ *for* $0 < n < 2N$. Some authors further multiply the X_0 term by 1/√2 and multiply the resulting matrix by an overall scale factor of $\sqrt{\frac{2}{N}}$ [7]. This makes the DCT-II matrix orthogonal. Orthogonality the inverse transformation matrix of *A* is equal to its transpose i.e. $A^{-1} = A^T$. Therefore, and in addition to its decorrelation characteristics, this property renders some reduction in the pre computation complexity, but breaks the direct correspondence with a real-even DFT of half-shifted input [10]. The DCT-II implies the boundary conditions: x_n is even around n=-1/2 and even around $n=N$-1/2; *C(u)* is even around k=0 and odd around $k=N$. DCT-II is normally called 1- D DCT.

The Two-Dimensional DCT

The objective of this paper is to study the efficacy of DCT on images. This necessitates the extension of ideas presented in the last section to a two-dimensional space. The 2-D DCT is a direct extension of the 1-D case and is given by

$$C(u,v)=\frac{1}{4}\alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\frac{\cos(2x+1)u\pi}{2N}\frac{\cos(2y+1)v\pi}{2N} \qquad (2)$$

for u,v = 0,1,2,….,N — 1 and *a(u)* and *a(v)* are defined in (1)

Implementation of the 2-D DCT directly from the theoretical equation results in 1024 multiplications and 896 additions. Fast algorithms exploit the symmetry within the DCT to achieve dramatic computational savings.

There are three basic categories of approach for computation of the 2-D DCT. The first category of 2-D DCT implementation is indirect computation through other transforms—most commonly, the Discrete Hartley Transform (DHT) and the Discrete Fourier Transform (DFT) [8]. The DHT based algorithm of shows increased performance in throughput, latency, and turnaround time. Optimization with respect to these parameters is not the focus of the proposed project. A DFT approach calculates the odd-length DCT. The second style of algorithms computes the 2-D DCT by row-column decomposition. In this approach, the separability property of the DCT is exploited. An 8 point, 1-D DCT is applied to each of the 8 rows, and then again to each of the 8 columns [8]. The 1-D algorithm that is applied to both the rows and columns is the same. Therefore, it could be possible to use identical pieces of hardware to do the row computation as well as the column computation. The bulk of the design and computation is in the 8 point 1-D DCT block, which can potentially be reused 16 times—8 times for each row, and 8 times for each column. Therefore, the fast algorithm for computing the 1-D DCT is usually selected. The high regularity of this approach is very attractive for reduced cell count and easy very large scale integration (VLSI) implementation.

The third approach to computation of the 2-D DCT is by a direct method using the results of a polynomial transform. Computational complexity is greatly reduced, but regularity is sacrificed. Instead of the 16 1-D DCTs used in the conventional row-column decomposition, [6] uses all real arithmetic including 8 1-D DCTs, and stages of pre-adds and post-adds (a total of 234 additions) to compute the 2-D DCT. Thus, the number of multiplications for most implementations should be halved as multiplication only appears within the 1-D DCT. Although this direct method of extension into two dimensions creates an irregular relationship between inputs and outputs of the system, the savings in computational power may be significant with the use of certain 1-D DCT algorithms [9]. With this direct approach, large chunks of the design cannot be reused to the same extent as in the conventional row-column decomposition approach [10]. Thus, the direct approach will lead to more hardware, more complex control, and much more intensive debugging.

CONCLUSIONS

Systolic array architecture follows parallel processing concept in their working and DCT has the main application in image processing. It is very convenient to transform DCT from DFT, DHT or by polynomial transform. Image can be compressed, corrected in terms of burliness, contrast, brightness, illumines and motion blurriness using DCT. Combination of systolic architecture and DCT makes less computation time. Systolic Array and DCT can be implemented using VLSI.

REFERENCES

1. Kung, Hsiang-Tsung. "Why systolic architectures?." *IEEE computer* 15.1 (1982): 37-46.
2. Britanak, Vladimir, and Kamisetty Ramamohan Rao. "The fast generalized discrete Fourier transforms: A unified approach to the discrete sinusoidal transforms computation." *Signal Processing* 79.2 (1999): 135-150.
3. Rao, K. Ramamohan, and Ping Yip. *Discrete cosine transform: algorithms, advantages, applications*. Academic press, 2014.
4. Wang, Chin-Liang, and Chang-Yu Chen. "High-throughput VLSI architectures for the 1-D and 2-D discrete cosine transforms." *Circuits and Systems for Video Technology, IEEE Transactions on* 5.1 (1995): 31-40.
5. Gersho, Allen, and Robert M. Gray. *Vector quantization and signal compression*. Vol. 159. Springer Science & Business Media, 2012.
6. Khayam, Syed Ali. "The discrete cosine transform (DCT): theory and application." *Michigan State University* (2003).
7. Wang, Zhongde. "Fast algorithms for the discrete W transform and for the discrete Fourier transform." *Acoustics, Speech and Signal Processing, IEEE Transactions on* 32.4 (1984): 803-816.

8. Watson, Andrew B. "Image compression using the discrete cosine transform." *Mathematica journal* 4.1 (1994): 81.
9. Hsu, Chung-hsing, and Wu-chun Feng. "A power-aware run-time system for high-performance computing." *Proceedings of the 2005 ACM/IEEE conference on Supercomputing*. IEEE Computer Society, 2005.
10. Merhav, Neri, and Vasudev Bhaskaran. "Fast algorithms for DCT-domain image downsampling and for inverse motion compensation." *Circuits and Systems for Video Technology, IEEE Transactions on* 7.3 (1997): 468-476.