



Fast approximate DCT with GPU implementation for image compression [☆]



Reem T. Haweel ^{a,*}, Wail S. El-Kilani ^b, Hassan H. Ramadan ^a

^a Basic Science Department, Ain-Shams University, Faculty of Computer and Information Science, Abbasia, Cairo 11566, Egypt

^b Computer Systems Department, Ain-Shams University, Faculty of Computer and Information Science, Abbasia, Cairo 11566, Egypt

ARTICLE INFO

Article history:

Received 2 March 2015

Revised 18 December 2015

Accepted 7 July 2016

Available online 7 July 2016

Keywords:

Approximate DCT transforms

Fast algorithms

Image compression

GPU

ABSTRACT

Recent developments in image and video processing employed in multimedia and communication systems require fast 2-D Discrete Cosine Transforms (DCT). The DCT is widely employed in image compression for its high power compaction property. Approximate DCT transforms have been developed to proceed faster than the original DCT while maintaining comparative levels of power compaction. This paper introduces a multiplierless efficient and low complexity 8-point approximate DCT. A flow diagram is provided for the fast implementation of the proposed transform. Only 17 additions are required for both forward and backward transformations. A fast and efficient Graphics Processing Unit (GPU) implementation for the proposed transform is provided. Performance evaluation shows that the proposed transform outperforms other approximate DCT transforms in JPEG-like image compression.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

Image and video compression are essential in many recent systems. These include digital video in multimedia devices [1], geospatial remote sensing [2], automatic surveillance [3], traffic cameras [4], and homeland security [5]. Image sizes and videos are increasing massively and the need to accelerate the compression standards is becoming more demanding. Transform methods using orthogonal kernel functions are commonly used in image compression. The Discrete Cosine Transform (DCT) [6] incorporates real sinusoids and possesses many interesting features. In addition to its orthogonal structure, the DCT has good power compaction properties. The DCT is the best substitute for the Karhunen Loeve transform, which is considered to be statistically optimal for power concentration [7]. For this reason, the DCT is the core of image coding [8] and video compression techniques such as JPEG, MPEG1, MPEG2, H.261, and H.263 [9]. In spite of the existence of many fast algorithms [10] which reduce the total number of operations required to compute such transforms, multiplication operations may be inevitable. To increase the speed of transformation while keeping the compaction properties of the DCT, the Signed Discrete Cosine Transform (SDCT) [11] was suggested simply by applying

the signum function operator to the DCT elements. All the elements of the transform are 0 or ± 1 , that is no multiplication operations or transcendental expressions are required. Unfortunately, the SDCT transformation matrix is not orthogonal. The computation of the SDCT requires 24 additions [11]. Following the introduction of the SDCT, a stream of research papers has followed such as the Bouguezel-Ahmed-Swamy (BAS) series of algorithms [12–16]. The target is to modify the SDCT to further reduce the computational complexity and to achieve orthogonality. The strategy is to change some of the SDCT matrix elements and to clear others. An interesting modification to the SDCT is introduced in [17] with a transform, T_1

$$T_1 = \frac{D_1}{2\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 0.5 & -0.5 & -1 & -1 & -0.5 & 0.5 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0.5 & 0 & 0 & -0.5 & -0.5 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (1)$$

where

$$D_1 = \text{diag}(1, \sqrt{2}, 2\sqrt{2/5}, 2, 1, \sqrt{2}, 2, 2\sqrt{2/5}, 2) \quad (2)$$

[☆] This paper has been recommended for acceptance by M.T. Sun.

* Corresponding author.

E-mail addresses: reem_tarek@hotmail.com (R.T. Haweel), wail.elkilani@gmail.com (W.S. El-Kilani), hramadan@eun.eg (H.H. Ramadan).

In T_1 , Some of the SDCT elements have been changed to ± 0.5 and 24 elements have been cleared (turned to zero). It has been shown in [17] that the power compaction of the transform, and consequently the compression capabilities, is high. The computation of the transform requires 17 additions and two shifts. The transform in [17] is quasi orthogonal. There are two nonzero off-diagonal elements in $T_1 T_1^T$. However, the effect of these two nonzero elements on the compression is negligible and the approximation of the transform transpose as the transform inverse can be performed.

A prominent member of the BAS series is the parametric transform [16] which is an 8-point orthogonal transform containing a single parameter in its transformation matrix, T_2 given by

$$T_2 = D_2 \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & \alpha & -\alpha & -1 & -1 & -\alpha & \alpha & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ \alpha & -1 & 1 & -\alpha & -\alpha & 1 & -1 & \alpha \end{bmatrix} \quad (3)$$

where

$$D_2 = \text{diag}\left(1/\sqrt{8}, 1/2, 1/\sqrt{4+4\alpha^2}, 1/\sqrt{2}, 1/\sqrt{8}, 1/\sqrt{2}, 1/2, 1/\sqrt{4+4\alpha^2}\right) \quad (4)$$

The parameter, α is selected as a small integer in order to minimize the complexity of T_2 [16].

A collection of approximations for the 8-point DCT based on integer functions is presented in [18]. Considered functions include: the floor, ceiling, truncation, and rounding-off functions. Such approximations are orthogonal or quasi-orthogonal with low arithmetic complexity and low complexity inversion. The approximations in [18] are multiplierless requiring only additions and bit-shifting operations. Computational complexity ranged from 18 to 24 additions. A prominent orthogonal member of this collection with no bit-shifting is based on the rounding to zero function and is given by

$$T_3 = D_3 \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix} \quad (5)$$

where

$$D_3 = \text{diag}\left(1/\sqrt{8}, 1/\sqrt{6}, 1/2, 1/\sqrt{6}, 1/\sqrt{8}, 1/\sqrt{6}, 1/2, 1/\sqrt{6}\right) \quad (6)$$

Another interesting DCT approximation requiring only 14 additions is presented in [19]. It is based on a search over the 8×8 matrix space in order to find candidate matrices that possess low computation cost. The search is restricted to produce orthogonal matrices with elements in the range $\in \{0, 1, 2\}$. The transforming matrix is given by

$$T_4 = D_4 \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 1 & -1 & 0 & 0 \end{bmatrix} \quad (7)$$

where

$$D_4 = \text{diag}\left(1/\sqrt{8}, 1/\sqrt{2}, 1/2, 1/\sqrt{2}, 1/\sqrt{8}, 1/\sqrt{2}, 1/2, 1/\sqrt{2}\right) \quad (8)$$

Another transform with 26 inserted zero elements has been introduced in [20]. The other elements in this transform are ± 1 . Due to the large number of inserted zeroes, the computational requirements are 17 additions only. Unfortunately, the power compaction of this transform is relatively low due to the random insertion of zeros. Moreover, the transform is not orthogonal and the transform transpose cannot substitute the transform inverse which offsets the advantage of the reduced computational requirements.

Alternate approximation to the DCT has been obtained by scaling then rounding-off the scaled DCT matrix [21,22]. The resulting transform has ± 1 elements and 16 zeroes. The transform is orthogonal and the computational requirements are 22 additions which is relatively high.

This paper is twofold. In the first part of the paper, the proposed algorithm is introduced in Section 2. Section 3 demonstrates the improved performance of the proposed transform in image compression. In the second part of the paper, an efficient GPU implementation for the proposed transform is illustrated in Section 4. Finally, Section 5 concludes the work.

2. Proposed approximate DCT transform

The aim of this research is to introduce an efficient approximate DCT transform attaining both high energy compaction properties and low computational complexity. Referring to the literature review summarized in the introduction of Section 1 before, the algorithm of [17] has high compaction properties and a comparative low complexity. However, there are shift operations that may cause problems. This is because, in hardware, a shift operation may be represented as a right-shift which may incur in computational errors. The proposed approximate DCT (PADCT) matrix, referred to as T_p , results from applying the signum function operator to the transform in [17] given in Eq. (1). The signum function retains the overall spectral properties and consequently retains the high compaction properties. Moreover, the problematic shift operations in [17] have now been waved leading to an enhancement in the overall computational complexity. The diagonal matrix in Eq. (2) is modified to normalize the main diagonal elements in $T_p T_p^T$ to ones for a uniform computation [34]. That is the proposed transform; T_p and its associated diagonal D_p are given by

$$T = D_p T_p \quad (9)$$

where

$$T_p = \text{sign}\{T_1\} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (10)$$

Download English Version:

<https://daneshyari.com/en/article/528516>

Download Persian Version:

<https://daneshyari.com/article/528516>

[Daneshyari.com](https://daneshyari.com)