Performance analysis of Integer DCT of different block sizes.

Aim: To investigate performance analysis of integer DCT of different block sizes.

Abstract:

Discrete cosine transform (DCT) has been serving as the main component of video coding systems. The integer discrete cosine transform (Int DCT) is an integer approximation of the discrete cosine transform. It can be implemented exclusively with integer arithmetic. It proves to be highly advantageous in cost and speed for hardware implementations. In particular, transforms of sizes larger than 4x4 or 8x8, especially 16x16 and 32x32 are proposed because of their increased applicability to the de-correlation of high resolution video signals. For example, order-16 integer transform is simple, low computational complexity transform but has high coding efficiency.

This project discusses how the use of larger transforms, especially in high resolution videos, can provide higher coding gain. The DCT-based systems have huge advantage to image applications because they provide a high compression ratio. However, their coding systems are limited to operating in only lossy coding because distortion of decoded image is unavoidable with these lossy algorithms. On the other hand, the integer transform, is becoming popular as a key technique to lossless and lossy unified waveform coding. Especially the integer DCT is attractive as the unified coding comparable to the conventional DCT-based algorithms.

Submitted by:

Suvinda Mudigere Srikantaiah  
UTA ID: 1000646539  
Email: Suvinda.mudigeresrikantaiah@mavs.uta.edu  
Date: 28th September 2010
Introduction:

In digital image processing, data compression is necessary to improve efficiency in storage and transmission. Transformation is one popular technique for data compression. By first transforming correlated pixels into weakly correlated ones, and after a ranking in their energy contents, for example, and retaining only the most significant components, high compression ratio is possible. Since inverse transformation is needed to reproduce the original image from the compressed data, it is important that the transform process be simple and fast. The family of orthogonal transforms is well suited for this application because the inverse of an orthogonal matrix is its transpose. The discrete cosine transform (DCT) is widely accepted as having a high efficiency [1].

The DCT matrix elements are real numbers and for a 16-order DCT, 8 bits are needed to represent these numbers in order to ensure perfectly negligible image reconstruction errors due to finite-length number representation. If the transform matrix elements are integers, then it may be possible to have a smaller number of bit representation and at the same time zero truncation errors. Using the principle of dyadic symmetry [2] order-8 integer cosine transform (ICT) which has zero truncation errors was introduced. This requires a small number, as little as 2 bit representation and comparable efficiency to the DCT [3].

Briefly, an ICT matrix is in the form \( I = KJ \) where \( I \) is the orthogonal ICT matrix, and \( K \) is a diagonal matrix whose elements take on values that serve to scale the rows of the matrix \( J \) so that the relative magnitudes of elements of the ICT matrix \( I \) are similar to those in the DCT matrix. The matrix \( J \) is orthogonal with elements that are all integers.

Transforms used in some standards [4]:

<table>
<thead>
<tr>
<th>Standard</th>
<th>Transform</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. MPEG-4 part 10/H.264</td>
<td>8 X 8, 4 X 4 integer DCT, 4 X 4, 2 X 2 Hadamard</td>
</tr>
<tr>
<td>2. WMV-9</td>
<td>8 X 8, 8 X 4, 4 X 8, 4 X 4 integer DCT</td>
</tr>
<tr>
<td>3. AVS China</td>
<td>Asymmetric 8 X 8 integer DCT</td>
</tr>
</tbody>
</table>

Table no.1: Transforms used in standards H.264 [4], WMV-9 [17] and AVS China [16].

Insight into the project:

As part of an image-compression system, the role of the ICT is to de-correlate the picture elements of image blocks for subsequent quantization and entropy encoding. The order-8 ICT was derived using the principle of dyadic symmetry. This concept gives a different development that leads to the order-16 ICT [5]. Equations relating the elements of the ICT matrix so as to satisfy the orthogonality conditions among the columns of the ICT matrix are first written. Then a search method is proposed to find integer solutions for these elements.
It is noticed that the coding performance of the Int-DCT is similar to that of the conventional lossy DCT in a low bit-rate but it is slightly worse than that of the conventional lossy DCT in a high bit-rate because of rounding errors.

I. DCT

Definition of DCT and IDCT [11]:

The forward discrete cosine transform (DCT) of $N$ samples is formulated by:

$$ F(u) = \sqrt{\frac{2}{N}} C(u) \sum_{x=0}^{N-1} f(x) \cos \left( \frac{\pi(2x + 1)u}{2N} \right) $$

for $u = 0, 1, \ldots, N - 1$, where

$$ C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u = 0 \\ 1 & \text{otherwise.} \end{cases} $$

The function $f(x)$ represents the value of the $x$th sample of the input signal. $F(u)$ represents a discrete cosine transformed coefficient for $u = 0, 1, \ldots, N - 1$

First of all this transform is applied to the rows, then to the columns of image data matrix.

The inverse discrete cosine transform (IDCT) of $N$ samples is formulated by:

$$ f(x) = \sqrt{\frac{2}{N}} \sum_{u=0}^{N-1} C(u) F(u) \cos \left( \frac{\pi(2x + 1)u}{2N} \right) $$

for $x = 0, 1, \ldots, N - 1$, where

$$ C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u = 0 \\ 1 & \text{otherwise.} \end{cases} $$

This is used for image decompression.
The **DCT-II** is probably the most commonly used form, and is often simply referred to as "the DCT" [6].

Given an input function $f(i,j)$ over two integer variables $i$ and $j$ (a piece of an image), the 2D DCT transforms it into a new function $F(u,v)$, with integer $u$ and $v$ running over the same range as $i$ and $j$. The general definition of the transform is:

$$F(u,v) = \frac{2 C(u) C(v)}{\sqrt{MN}} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \cos \left(\frac{2i + 1}{2M}\cdot \pi\right) \cdot \cos \left(\frac{2j + 1}{2N}\cdot \pi\right) \cdot f(i,j)$$

where $i,u = 0,1,…,M - 1; \quad j,v = 0,1,…,N - 1; \quad$ the constants $C(u)$ (or $C(v)$) are defined as

$$C(l) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } l = 0 \\ 1 & \text{otherwise} \end{cases}$$

where $l = u,v$

**II. ORDER-4 INTEGER TRANSFORM**

Considering the example of transform used in H.264 [4], H.264 uses an adaptive transform block size, 4 X 4 and 8 X 8 (high profiles only). For improved compression efficiency, H.264 also employs a hierarchical transform structure. The DC coefficients of neighboring 4 X 4 transforms for the luma signals are grouped into 4 X 4 blocks and transformed again by the Hadamard transform. For blocks with mostly flat pel values, there is significant correlation among transform DC coefficients of neighboring blocks. Therefore, the standard specifies the 4 X 4 Hadamard transform for luma DC coefficients for 16 X 16 Intra-mode only, and 2 X 2 Hadamard transform for chroma DC coefficients.

The 4X4 IntDCT matrix is obtained using matrix $H$,

$$H = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix}$$
The variables a, b, c are as follows:

\[ a = \frac{1}{2}, \quad b = \sqrt{\frac{1}{2}} \cos \left( \frac{\pi}{8} \right), \quad c = \sqrt{\frac{1}{2}} \cos \left( \frac{3\pi}{8} \right) \]

However, to simplify the implementation of the transform, c is approximated by 0.5. For ensuring orthogonality, b also needs to be modified so that

\[ a = \frac{1}{2}, \quad b = \sqrt{\frac{2}{5}}, \quad c = \frac{1}{2}. \]

Thus order-4 integer transform,

\[
\begin{bmatrix}
1 & 1 & 1 & 1 \\
2 & 1 & -1 & -2 \\
1 & -1 & -1 & 1 \\
1 & -2 & 2 & -1 \\
\end{bmatrix}
\]

III. ORDER-8 INTEGER TRANSFORM

Consider the ICT of a one-dimensional data vector X of size 8 [7]. This ICT is implemented by premultiplying X by the orthogonal matrix C, given by

\[
C = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
5 & 3 & 2 & 1 & -1 & -2 & -3 & -5 \\
3 & 1 & -1 & -3 & -3 & -1 & 1 & 3 \\
3 & -1 & -5 & -2 & 2 & 5 & 1 & 3 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
2 & -5 & 1 & 3 & -3 & -1 & 5 & -2 \\
1 & -3 & 3 & -1 & -1 & 3 & -3 & 1 \\
1 & -2 & 3 & -5 & 5 & -3 & 2 & -1 \\
\end{bmatrix}
\]

One attractive characteristic of the ICT is that the absolute values of its coefficients are equal to powers of two, or powers of two, plus one.
Denoting the ICT vector by $Y$,

Which can be written as $Y = CX$.
as $C$ is an orthogonal matrix, $D = CC^T$, where $D$ is diagonal.
Now, let $\Delta$ denote the inverse of $D$.
Clearly, factoring the identity matrix $I$ as

$$I = DD^{-1}$$
$$I = D\Delta$$
$$I = \sqrt{\Delta} D \sqrt{\Delta}$$

using, $D = CC^T$, $I = \sqrt{\Delta} C \sqrt{\Delta}$. Thus, it can be identified that $M = \sqrt{\Delta} C$ is an orthonormal matrix,
so that $M^T = C^T \sqrt{\Delta} = M^{-1}$. The matrix $M$ represents the normalized ICT.

**IV. ORDER-16 INTEGER TRANSFORM**

Order-16 ICT has been shown to be very close to order-16 DCT [8]. A simple integer transform for video coding is considered and its test is based on a set of CIF sequences. The order-16 transform considered is an extended version of the order-8 ICT adopted in AVS. As shown in (1), $T_8$ is the order-8 transform matrix. Without significant increase in complexity, $T_8$ can be extended to order-16 transform $T_{16}$ as shown in (2). The normalized basis vectors of $T_{16}$ have the waveforms similar to that of DCT.

$$T_8 = \begin{bmatrix}
8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
10 & 9 & 6 & 2 & -2 & -6 & -9 & -10 \\
10 & 4 & -4 & -10 & -10 & -4 & 4 & 10 \\
9 & -2 & -10 & -6 & 6 & 10 & 2 & -9 \\
8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\
6 & -10 & 2 & 9 & -9 & -2 & 10 & -6 \\
4 & -10 & 10 & -4 & -4 & 10 & -10 & 4 \\
2 & -6 & 9 & -10 & 10 & -9 & 6 & -2
\end{bmatrix}$$

(1) $T_8$: Order 8 transform matrix [5].
$T_{16} =$
\[
\begin{bmatrix}
8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
10 & 10 & 9 & 9 & 6 & 6 & 2 & 2 & -2 & -2 & -6 & -6 & -9 & -9 & -10 & -10 \\
10 & 10 & 4 & 4 & -4 & -4 & -10 & -10 & -10 & -10 & -4 & -4 & 4 & 4 & 10 & 10 \\
9 & 9 & -2 & -2 & -10 & -10 & -6 & -6 & 6 & 6 & 10 & 10 & 2 & 2 & -9 & -9 \\
6 & 6 & -10 & -10 & 2 & 2 & 9 & 9 & -9 & -9 & -2 & -2 & 10 & 10 & -6 & -6 \\
4 & 4 & -10 & -10 & 10 & 10 & -4 & -4 & -4 & -4 & 10 & 10 & -10 & -10 & 4 & 4 \\
2 & 2 & -6 & -6 & 9 & 9 & -10 & -10 & 10 & 10 & -9 & -9 & 6 & 6 & -2 & -2 \\
4 & -4 & -4 & 10 & 10 & -10 & -10 & 4 & -4 & 10 & 10 & -10 & -10 & 4 & 4 & -4 \\
6 & -6 & -6 & 10 & 2 & -2 & 9 & -9 & -9 & 9 & -2 & 2 & 10 & -6 & -6 & 6 \\
9 & -9 & -2 & 2 & -10 & 10 & -6 & 6 & 6 & -6 & 10 & -10 & 2 & -2 & -9 & 9 \\
10 & -10 & 4 & -4 & -4 & 4 & -10 & 10 & -10 & 10 & -4 & 4 & 4 & -4 & 10 & -10 \\
10 & -10 & 9 & -9 & 6 & -6 & 2 & -2 & -2 & 2 & -6 & 6 & -9 & 9 & -10 & 10 \\
\end{bmatrix}
\]

(2) $T_{16}$: Order 16 transform matrix [5].

**Performance Evaluation:**

The efficiency of a transform is generally defined as its ability to decorrelate a vector or random elements. In finding efficiency of integer DCT, standard images are applied as an input signal. Transforms considered will be DCT, Integer DCT of different block sizes.

The following operations are performed:

a) Variance distribution for 1 order Markov process, $\rho = 0.9$ (Plot and Tabulate)
b) Normalized basis restriction error vs. # of basis function (Plot and Tabulate)
c) Obtain transform coding gains
d) Plot fractional correlation ($0 < \rho < 1$)

The entropy of the transformed data to assess the lossless coding results and the Peak signal-to-noise ratio, PSNR are computed to evaluate lossy coding experiments [9]. It is found that, since the different information encoded, the lossless coding with float DCT (DCT in its real number form, with a fractional part) performs worse than integer DCT, which is the very advantage of integer DCT. It is supposed to follow the expectation that coding results of lossy coding with float DCT and integer DCT are very similar, since the entropy (a measure of the uncertainty associated with a random variable) of integer DCT is close to that of rounded float DCT [10].
References:

5. W. Cham and C. Fong “Simple order-16 integer transform for video coding” ICIP, Hong Kong, Sept.2010.