Fast Conversion of H.264/AVC Integer Transform Coefficients into DTT Coefficients

Muhammed Yusuf Khan¹ and Tanveer Hasan²

Faculty of Engineering & Technology, Aligarh Muslim University, Aligarh, India.

Abstract

In this paper we propose a fast method to convert H.264/AVC Integer Transform (IT) coefficients to Discrete Tchebichef Transform (DTT) coefficients for applications in video transcoding. We derive the transform kernel matrix for converting, in the transform domain, four IT coefficient blocks into one 8x8 DTT block of coefficients. Using the symmetry of this matrix, we show that the proposed conversion method requires lesser computations than its equivalent in the pixel domain.

Keywords: Transform conversion, video transcoding.

1. INTRODUCTION

It is observed that about half of the bitrate can be saved by using H.264 [1]. When a new standard is adopted, it gives rise to interoperability problems with older systems. In the case of H.264, interoperability with MEPG-2 systems is significant. It is achieved through video transcoding for conversion between different standards [2]. MPEG 2 video codecs use the 8x8 DCT to produce transform coefficients, H.264 use the 4x4 IT. The Discrete Tchebichef Transform (DTT) [3] which is derived from Orthonormal Tchebichef Ploynomials has similar energy compaction properties like DCT. In paper [4] the conversion of DTT coefficients to IT coefficients is compared with DCT coefficients to IT coefficients conversion. The fast conversion of H.264/AVC Integer Transform Coefficients into DCT coefficients is computed by Marques et al[5].This paper goes step further by developing fast algorithm for converting H.264/AVC Integer Transform (IT) coefficients to Discrete Tchebichef
Transform (DTT) coefficients for video transcoding applications. The organization of this paper is as follows. In section 2, we describe the transform domain IT-to-DCT conversion. In section 3 the fast conversion algorithm & its utility is described.

2. IT-TO-DCT CONVERSION

The IT-to-DTT conversion in pixel domain is shown in figure 1. The input consists of four 4x4 blocks \((X_1, X_2, X_3, X_4)\) of IT coefficients. The inverse IT is applied to each of the four blocks in order to obtain the pixel domain blocks \((x_1, x_2, x_3, x_4)\). Then the four pixel domain blocks are grouped to form a single 8x8 block \((x)\) to which the DTT is applied, such that an 8x8 block of transform coefficients \((Y)\) is obtained. The transform domain conversion is better because complete decoding up to the pixel domain is not needed. The transform domain IT-to-DTT conversion is based on the so-called S-matrix \([4][6]\). It is applied to an 8x8 block \((X)\) comprised of four 4x4 blocks \((X_1, X_2, X_3, X_4)\) of IT coefficients to produce the corresponding 8x8 block \((Y)\) of DTT coefficients. The conversion is given by the following operation,

\[
Y = SXS^T \quad (1)
\]

where \(S\) is the transform kernel matrix and \(S^T\) is its transpose. To obtain \(S\), we have to consider the inverse IT of blocks \(X_1, X_2, X_3, X_4\) which results in pixel blocks \(x_1, x_2, x_3, x_4\), each one given by \(x_i = J X_i J^T\), (i=1,2,3,4) where \(J\) is the matrix given by equation 2[7]

\[
J =
\begin{bmatrix}
1 & 1 & 1 & 0.5 \\
1 & 0.5 & -1 & -1 \\
1 & -0.5 & -1 & 1 \\
1 & -1 & 1 & -0.5
\end{bmatrix}
\]  

(2)

Take

\[
K =
\begin{bmatrix}
J & 0 \\
0 & J
\end{bmatrix}
\]

we can compute \(y\) in a single step as given by,

\[
y = KXK^T \quad (3)
\]

DTT of an 8x8 block is defined as

\[
Y = TyT^T \quad (4)
\]

\(T[3]\) is the DTT kernel matrix, then it follows that,

\[
Y = TKX K^T T^T \quad (5)
\]

From the above equation the transform kernel matrix \(S\) can defined as

\[
S = TK \quad (6)
\]
where the transform kernel matrix $S[4]$ is given by,

\[
S = \begin{bmatrix}
  a & -b & 0 & c & -d & -e & 0 & f \\
  0 & -g & h & -i & -j & -k & -l & -m \\
  0 & 0 & n & -o & p & -q & -r & s \\
  0 & -t & -u & v & w & -x & -y & -z \\
  a & b & 0 & -c & -d & e & 0 & -f \\
  0 & -g & -h & -i & j & k & l & -m \\
  0 & 0 & n & o & p & q & -r & -s \\
  0 & -t & -u & v & -w & -x & -y & -z
\end{bmatrix}
\]

and each value of the $S$ matrix as follows,

\[
\begin{align*}
a &= 1.4144; & b &= 1.2346; & c &= 0.4923; & d &= 0.0001; & e &= 0.3424; & f &= 0.3413; & g &= 1.0801; & h &= 2.1605; & i &= 1.3539; \\
& & j &= 0.5641; & k &= 1.1984; & l &= 0.1232; & m &= 0.7510; & n &= 0.3086; & o &= 0.9847; & p &= 1.2893; & q &= 0.5992; \\
r &= 0.4924; & s &= 0.8193; & t &= 0.1543; & u &= 0.3085; & v &= 0.3692; & w &= 0.7252; & x &= 1.5408; & y &= 2.0924; & z &= 1.5705;
\end{align*}
\]

3. FAST ALGORITHM

The fast IT-to-DTT algorithm is based on exploiting the symmetry properties of the $S$ matrix. As the transform defined by (1) is separable, it can be computed by column transforms followed by row transforms. If we define $p$ as an 8 point row vector and $P$ its 1D transform, then by using the vertical symmetry of the $S$ matrix, we can map the combinations defined in each line of $S$ to 8 variables. The 1D transform $P$ is obtained from linear operations of these variables. We have derived the following algorithm in order to achieve an efficient method to obtain $P$.

\[
\begin{align*}
\end{align*}
\]

\[
\begin{align*}
P[1] &= a m_1; \\
P[2] &= -b m_2 - g m_3 - t m_4; \\
P[3] &= h m_5 + n m_6 + u m_7; \\
P[4] &= c m_2 + i m_3 + o m_8 - v m_4; \\
P[5] &= -d m_1 - j m_5 + p m_6 + w m_7; \\
P[6] &= -e m_2 + k m_3 + q m_8 - x m_4; \\
P[7] &= -l m_5 - r m_6 + y m_7; P[8] &= f m_2 - n m_3 - s m_8 - z m_4;
\end{align*}
\]

This algorithm requires 26 multiplications and 26 additions, i.e., a total of 52 operations to perform the 1D transform. The 2D $S$-transform needs 8 columns 1D
transform (8x52=416 operations) and 8 rows 1D transforms (416 operations). Hence the proposed fast conversion algorithm needs a total of 832 operations. The pixel domain approach needs four inverse IT (320 operations) and one DTT (672 operations) which is equal to 992 operations. The proposed transform domain fast algorithm significantly reduces (16%) the number of operations when compared to the pixel domain conversion. The proposed method is much faster than the pixel domain approach. The method is useful for video transcoding applications where fast processing is needed.

![Diagram of Pixel domain IT-to-DTT conversion](image)

**Figure 1** Pixel domain IT-to-DTT conversion

REFERENCES


[3] Sujata Eshwar Discrete Tchebichef Transform and its Application to Image/Video Compression, A Thesis, Concordia University Montreal, Quebec, Canada

Fast Conversion of H.264/AVC Integer Transform Coefficients into DIT Coefficients

Engineering Technology Science and Management Innovation (ICETSMI-2017) at Indian Federation of United Nations Associations, New Delhi, India.


Muhammed Yusuf Khan and Tanveer Hasan,