Journal of Signal and Information Processing, 2015, 6, 203-216
Published Online August 2015 in SciRes. http://www.scirp.org/journal/jsip
http://dx.doi.org/10.4236/jsip.2015.63019

A Comparison of Integer Cosine and Tchebichef Transforms for Image Compression Using Variable Quantization

Soni Prattipati[1], M. N. S. Swamy[1], Pramod K. Meher[2]

[1]Department of Electrical and Computer Engineering, Concordia University, Montreal, Canada
[2]Institute of Infocomm Research, Singapore City, Singapore
Email: so_prat@encs.concordia.ca, swamy@ece.concordia.ca, pkmeher@i2r.a-star.edu.sg

Received 9 May 2015; accepted 17 July 2015; published 21 July 2015

Abstract

In the field of image and data compression, there are always new approaches being tried and tested to improve the quality of the reconstructed image and to reduce the computational complexity of the algorithm employed. However, there is no one perfect technique that can offer both maximum compression possible and best reconstruction quality, for any type of image. Depending on the level of compression desired and characteristics of the input image, a suitable choice must be made from the options available. For example in the field of video compression, the integer adaptation of discrete cosine transform (DCT) with fixed quantization is widely used in view of its ease of computation and adequate performance. There exist transforms like, discrete Tchebichef transform (DTT), which are suitable too, but are potentially unexploited. This work aims to bridge this gap and examine cases where DTT could be an alternative compression transform to DCT based on various image quality parameters. A multiplier-free fast implementation of integer DTT (ITT) of size 8 × 8 is also studied, for its low computational complexity. Due to the uneven spread of data across images, some areas might have intricate detail, whereas others might be rather plain. This prompts the use of a compression method that can be adapted according to the amount of detail. So, instead of fixed quantization this paper employs quantization that varies depending on the characteristics of the image block. This implementation is free from additional computational or transmission overhead. The image compression performance of ITT and ICT, using both variable and fixed quantization, is compared with a variety of images and the cases suitable for ITT-based image compression employing variable quantization are identified.

Keywords

Discrete Tchebichef Transform (DTT), Variable Quantization, Image Compression, Multiplier Free Implementation of ITT

How to cite this paper: Prattipati, S., Swamy, M.N.S. and Meher, P.K. (2015) A Comparison of Integer Cosine and Tchebichef Transforms for Image Compression Using Variable Quantization. *Journal of Signal and Information Processing*, **6**, 203-216. http://dx.doi.org/10.4236/jsip.2015.63019

1. Introduction

In the area of data communication and storage, there is always a struggle to convey the most with the least amount of representation. Image compression in particular, reduces the presence of inherent redundancy and irrelevant details in images to reduce the size of the data needed for representation. Lossy compression is preferred over Lossless compression for applications where a minor loss of quality is acceptable as they pledge high rates of compression. Applications like space research and defense often require huge volumes of video data to be transmitted, analyzed and a suitable response triggered instantaneously. This mandates the use of compression techniques to accelerate the transmission procedure, which consumes the major portion of the total response time.

In transform-coded compression techniques, the image undergoes transformation before being encoded. The discrete Tchebichef transform (DTT), based on a polynomial kernel derived from discrete Tchebichef polynomials (DTPs), was introduced in [1] and was developed later into an orthogonal transform in [2]. The discrete cosine transform (DCT), which originates from the trigonometric representation of the Tchebichef polynomials of the first kind, has its image compression potential exhaustively investigated. Numerous fast DCT algorithms [3]-[5] have been developed and multiple DCT-based compression techniques have been established. DTT exhibits energy compaction and de-correlation properties that are very much comparable with those of DCT. For a block size 8 × 8 and correlation coefficient 0.9, the de-correlation efficiencies of DTT and DCT are 97.34% and 97.82%, respectively [6] [7]. The first five diagonal elements of the transformation matrix result in energy packing efficiencies of 97.41% and 97.6% for DTT and DCT, respectively [6] [7]. DTT can be shown to conform easily to an integer representation without the need for approximation unlike the Integer counterpart of DCT. This reveals that DTT could be a potential competitor to DCT for certain class of images.

Lately, integer approximation of DCT, integer cosine transform (ICT) is gaining popularity thanks to its reduced computational complexity and growing relevance of integer transforms in general. ICT is being used in high efficiency video coding (HEVC). HEVC provides better video compression performance than H.264/MPEG-4 AVC does. The implementation algorithm of ICT used in HEVC is in [8]. It can be observed that ICT is not a completely faithful adaptation of the DCT as scaling is involved during the estimation process; however, it has been shown that there is no approximation involved while developing ITT from DTT. Thus, it is of interest to compare the compression performance of ITT and ICT for various images and identify cases where ITT is a better choice. This work aims at identifying areas where ITT stands up to be a viable alternative to ICT for lossy image compression based on various image quality assessment parameters. Based on these findings, it would be possible to predict if an image is suited for ITT-based compression or ICT-based compression. An ITT for a block size of 8 × 8 is utilized here, whose computational complexity is compared with that of ICT and is completely multiplier-free.

It is advantageous to further enhance the lossy compression algorithm by including a variable quantization technique thereby ensuring better image quality even at higher compression rates. Quantization is a significant step during the lossy image compression procedure and optimizing it for better performance without an increase in computational complexity is highly desirable. It is unusual for an image to have a homogenous distribution of data all over. As a result, selecting a distinct quantization matrix optimized for each individual image block is appealing, but it would require transmitting the quantization information along with each block of the image. This significant transmission overhead defeats the very purpose of compression. In [9], a predefined collection of quantization matrices were maintained at the transmitter and the receiver to avoid transmitting the entire quantization matrix along with each image block. However, this seriously hinders the flexibility of operation as the user is restricted to a pre-determined set of quantization matrices. For instance, the image block might be robust enough to withstand higher quantization, which might not be supported at the receiver. Depending on the type of images being compressed, the receiver might need to be updated periodically to ensure the adequacy of the quantization set. This predicament can be avoided by making the procedure truly adaptive and allowing the user a wider choice of quantization matrices without any additional transmission overheadby using a special encoding technique. This work evaluates the performance of ITT with respect to that of ICT for compression of some special images using the novel variable quantization technique.

The rest of the paper is organized as follows: Section 2 defines DTT and its inverse, followed by Section 3 which presents ITT along with its multiplier free implementation. In Section 4, the variable quantization technique and the novel encoding method employed are discussed along with the image compression and decompression schemes using the transform. In Section V, the performance evaluation parameters used for comparison of the image compression using ITT with that of the ICT using variable and fixed quantization are presented.

The simulated results of compression for a variety of images are presented and interpreted. Finally, Section 5 contains the conclusions.

2. Discrete Tchebichef Polynomial and Transform

Transforms refer to conversion of signals between different domains like time and frequency. The representation of a signal in multiple domains makes it suitable for various applications like compression, analysis, detection, filtering, etc. There are different popular discrete transforms like discrete cosine, discrete Fourier, discrete wavelet and discrete Hartley transforms. Each of these transforms represents the signal in varied domains, bringing out specific innate characteristics which are suitable for different applications.

Discrete signals can be denoted using linear combination of sinusoidal or complex exponential sequences. Thus polynomials which are orthogonal in particular, are widely employed to represent the signals in view of their efficient data fitting and approximation abilities. Some of the popular discrete orthogonal polynomials are Hahn, Meixner, Krawtchouk and Charlier.

Discrete Tchebichef polynomials (DTP) [6] [7] [10] are the class of hyper geometric orthogonal polynomials associated with the distribution $d\alpha(n)$ of Stieltjes type, where $\alpha(n)$ is a step function with unit jumps at support points $n = 0,1,\cdots,N-1$. They are a special case of Hahn polynomial, which is a classic discrete orthogonal polynomial and a discrete analog of the popular Jacobi polynomial.

If Δ is the forward difference operator, the DTP is defined by the difference formula

$$t_k(n) = k!\Delta^k\left[\binom{n}{k}\binom{n-N}{k}\right] \text{ for } k = 0,1,\cdots,N-1 \tag{1}$$

where $t_k(n)$ is the k^{th} Tchebichef polynomial.

DTP exhibits important properties like orthogonality, recurrence and normalization which can be used to define a related transfer function $\tau(k,n)$ where k and n correspond to the indices of the frequency and time components, respectively, as shown in [9] [10].

Properties of DTP

***A. Orthogonality*:** DTPs are orthogonal with respect to a unit weight function

$$\sum_{n=0}^{N-1} t_k(n)t_m(n) = \delta_{km}d(k,N) \tag{2}$$

where $t_k(n)$ and $t_m(n)$ are the k^{th} and m^{th} Tchebichef polynomials, respectively, and $d(k,N)$ is the squared norm of the DTP defined as

$$d(k,N) = \sum_{n=0}^{N-1}\left[t_k(n)\right]^2 \tag{3}$$

The squared norm of the DTP can also be written as

$$d(k,N) = \frac{N\left(N^2-1\right)\left(N^2-2^2\right)\cdots\left(N^2-k^2\right)}{2k+1} \tag{4}$$

***B. Direct Representation*:** DTP can be evaluated using the closed form expression

$$t_k(n) = k!\sum_{i=0}^{k}(-1)^{k-i}\binom{N-1-i}{k-i}\binom{k+i}{k}\binom{n}{i} \tag{5}$$

***C. Recurrence Relation*:** DTPs satisfy a three-term recurrence relation given by

$$t_k(n) = \left[\left(\frac{2k-1}{k}\right)t_1(n)\right]t_{k-1}(n) - \left[\left(\frac{k-1}{k}\right)\left(N^2-(k-1)^2\right)\right]t_{k-2}(n) \tag{6}$$

for $k \ge 2$ with $t_0(n) = 1$, $t_1(n) = 2n - N + 1$. This formula is useful for a fast evaluation of polynomials of higher degrees.

***D. Symmetry*:** The DTP satisfies the symmetry property

$$t_k(N-1-n) = (-1)^k t_k(n) \tag{7}$$

E. Normalization of the DTP: The system defined in (1) is orthogonal. Every orthogonal system can be normalized by replacing $t_m(x)$ by $\langle t_m \cdot t_m \rangle^{-\frac{1}{2}} t_m(x)$ [10], where $\langle . \rangle$ is the scalar product. In view of (3), orthonormal DTP, denoted by $\tau_k(n)$, is given by

$$\tau_k(n) = \frac{t_k(n)}{\sqrt{d(k,N)}} \tag{8}$$

As a consequence, the recurrence relation in (6) may be rewritten for orthonormal DTPs in the form [2]

$$\tau_k(n) = (a_1 n + a_2)\tau_{k-1}(n) + a_3 \tau_{k-2}(n) \tag{9}$$

With $\tau_0(n) = \frac{1}{\sqrt{N}}, \tau_1(n) = (2n+1-N)\sqrt{\frac{3}{N(N^2-1)}}$ where

$$a_1 = \frac{2}{3}\sqrt{\frac{4k^2-1}{N^2-k^2}}, a_2 = \frac{1-N}{k}\sqrt{\frac{4k^2-1}{N^2-k^2}}, a_3 = \frac{1-k}{k}\cdot\frac{2k+1}{2k-3}\sqrt{\frac{N^2-(k-1)^2}{N^2-k^2}}$$

From the discrete Tchebichef polynomial and its various properties, a discrete Tchebichef transform (DTT) is derived.

In the DTT domain, the transformed sequence, *Y*(*k*), of an input data sequence, *x*(*n*), is given by

$$Y(k) = \sum_{n=0}^{N-1} \tau(k,n)\, x(n) \quad \text{for } k,n = 0,1,\cdots,N-1 \tag{10}$$

where $\tau(k,n)$ is the orthogonal basis of DTP denoted as $\tau_k(n)$ in (8) over an interval $[0, N-1]$ and whose values can be obtained from the recurrence relation in (9). In the above equation, *k* and *n* denote frequency and time indices, respectively.

The *inverse DTT* restores the input data when applied to the transform coefficients and is given by

$$x(n) = \sum_{k=0}^{N-1} \tau(k,n)\, Y(k) \quad \text{for } k,n = 0,1,\cdots,N-1 \tag{11}$$

For a 2-D input sequence *x* (n_1,n_2), the *2-D DTT of order N* × *M* is defined as

$$Y(k_1,k_2) = \sum_{n_1=0}^{N-1}\sum_{n_2=0}^{M-1} \tau(k_1,n_1)\,\tau(k_2,n_2)\, x(n_1,n_2) \tag{12}$$

for $k_1, n_1 = 0,1,\cdots,N-1$ and $k_2, n_2 = 0,1,\cdots,M-1$.

The *inverse 2-D DTT* is given by

$$x(n_1,n_2) = \sum_{k_1=0}^{N-1}\sum_{k_2=0}^{M-1} \tau(k_1,n_1)\,\tau(k_2,n_2)\, Y(k_1,k_2) \tag{13}$$

for $k_1, n_1 = 0,1,\cdots,N-1$ and $k_2, n_2 = 0,1,\cdots,M-1$

All the properties of DTP discussed are applicable to DTT as well.

3. Integer Tchebichef Transform

Floating-point operations preserve full-precision value, but at the cost of increased resource consumption. Moreover, floating-point operations are not defined in all processors. Integer operations offer increased computational speed as well as decreased complexity, cost and power consumption. This sub-section, discusses an integer Tchebichef transform of size 8 × 8, which greatly reduces the computational complexity.

Framework for Integer Transform Representation: The 2-D DTT of input data *X*, of size *N* × *N* given by (12), can also be expressed in the form below

$$\boldsymbol{Y} = \boldsymbol{\tau X \tau'} \tag{14}$$

where $\boldsymbol{\tau}$ is the "*N* × *N*" transform and can be factorized as shown below to separate the integer and float numbers. Consider a variable

$$q(k)=\frac{p(k)}{\sqrt{d(k,N)}} \tag{15}$$

where

$$p(k)=\begin{cases} k!k & \text{for } k=1 \\ k!(k+1) & \text{for } k=0,2,4,6 \\ k!(N-k) & \text{for } k=3,5,7 \end{cases}$$

and $d(k, N)$ is the squared norm of the DTP given by

$$d(k,N)=\frac{N\left(N^2-1\right)\left(N^2-2^2\right)\cdots\left(N^2-k^2\right)}{2k+1} \tag{16}$$

Let $\boldsymbol{Q}$ be a "$N \times N$" diagonal matrix with $q(k)$ given in (15) as the diagonal element of the k^{th} row. It is possible to express the transformation matrix $\boldsymbol{\tau}$ as

$$\boldsymbol{\tau}=\boldsymbol{Q}\hat{\boldsymbol{\tau}} \tag{17}$$

$$\hat{\boldsymbol{\tau}}=\boldsymbol{Q}^{-1}\boldsymbol{\tau} \tag{18}$$

Hence evaluating (18) by substituting the values of $\boldsymbol{\tau}$ from (9) and $\boldsymbol{Q}$ for $N = 8$ results in

$$\hat{\boldsymbol{\tau}}=\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -7 & -5 & -3 & -1 & 1 & 3 & 5 & 7 \\ 7 & 1 & -3 & -5 & -5 & -3 & 1 & 7 \\ -7 & 5 & 7 & 3 & -3 & -7 & -5 & 7 \\ 7 & -13 & -3 & 9 & 9 & -3 & -13 & 7 \\ -7 & 23 & -17 & -15 & 15 & 17 & -23 & 7 \\ 1 & -5 & 9 & -5 & -5 & 9 & -5 & 1 \\ -1 & 7 & -21 & 35 & -35 & 21 & -7 & 1 \end{pmatrix} \tag{19}$$

Consequently, from (17), the transform of (14) may be written as

$$\boldsymbol{Y}=\left(\boldsymbol{Q}\hat{\boldsymbol{\tau}}\right)\boldsymbol{X}\left(\boldsymbol{Q}\hat{\boldsymbol{\tau}}\right)'=\boldsymbol{Q}\left(\hat{\boldsymbol{\tau}}\boldsymbol{X}\hat{\boldsymbol{\tau}}'\right)\boldsymbol{Q} \tag{20}$$

By utilizing the properties of a diagonal matrix, the above equation can be expressed as

$$\boldsymbol{Y}=\left(\hat{\boldsymbol{\tau}}\boldsymbol{X}\hat{\boldsymbol{\tau}}'\right)\odot\hat{\boldsymbol{Q}} \tag{21}$$

where $\odot$ represents element-by-element multiplication and $\hat{\boldsymbol{Q}}$.

$\hat{\boldsymbol{Q}}$ can be obtained by multiplying as

$$\hat{\boldsymbol{Q}}=\left(\text{diagonal elements of } \boldsymbol{Q}\right)_{N\times 1}\times\left(\text{diagonal elements of } \boldsymbol{Q}\right)_{1\times N} \tag{22}$$

For $N = 8$, $\hat{\boldsymbol{Q}}$ is given by

$$\hat{\boldsymbol{Q}}=\begin{pmatrix} q_0^2 & q_0q_1 & q_0q_2 & q_0q_3 & q_0q_4 & q_0q_5 & q_0q_6 & q_0q_7 \\ q_0q_1 & q_1^2 & q_1q_2 & q_1q_3 & q_1q_4 & q_1q_5 & q_1q_6 & q_1q_7 \\ q_0q_2 & q_1q_2 & q_2^2 & q_2q_3 & q_2q_4 & q_2q_5 & q_2q_6 & q_2q_7 \\ q_0q_3 & q_1q_3 & q_2q_3 & q_3^2 & q_3q_4 & q_3q_5 & q_3q_6 & q_3q_7 \\ q_0q_4 & q_1q_4 & q_2q_4 & q_3q_4 & q_4^2 & q_4q_5 & q_4q_6 & q_4q_7 \\ q_0q_5 & q_1q_5 & q_2q_5 & q_3q_5 & q_4q_5 & q_5^2 & q_5q_6 & q_5q_7 \\ q_0q_6 & q_1q_6 & q_2q_6 & q_3q_6 & q_4q_6 & q_5q_6 & q_6^2 & q_6q_7 \\ q_0q_7 & q_1q_7 & q_2q_7 & q_3q_7 & q_4q_7 & q_5q_7 & q_6q_7 & q_7^2 \end{pmatrix}$$

In (21), $\left(\hat{\tau} \boldsymbol{X} \hat{\tau}'\right)$ represents the ITT, as the transformation matrix $\hat{\tau}$ is entirely composed of integers as seen in (19). The matrix multiplication with the input must be carried out in a manner as computationally less intensive as possible. The 1D ITT of input "$\boldsymbol{X}$" is given by $\left(\hat{\tau} \boldsymbol{X}\right)$. The elements of matrix "$\hat{\tau}$" for size $N = 8$ are given by (19). The corresponding 1D transformation $\boldsymbol{Y}$ for an input $\boldsymbol{X}$ of size 8 can be obtained by following the steps in **Table 1**.

It can be observed that the process is carried out predominantly by add and subtract operations, thereby reducing the number of multiplication operations which are computationally more intensive. The only multiplying coefficients that appear in the proposed implementation are 2, 3, 4, 6 and 8. The computational complexity can be further decreased by realizing the remaining multiplication operations by a series of shift and add operations as shown in **Table 2**.

Figure 1 shows a 5-stage architectural structure for ID ITT designed following the multiplier-free implementation described in this Section.

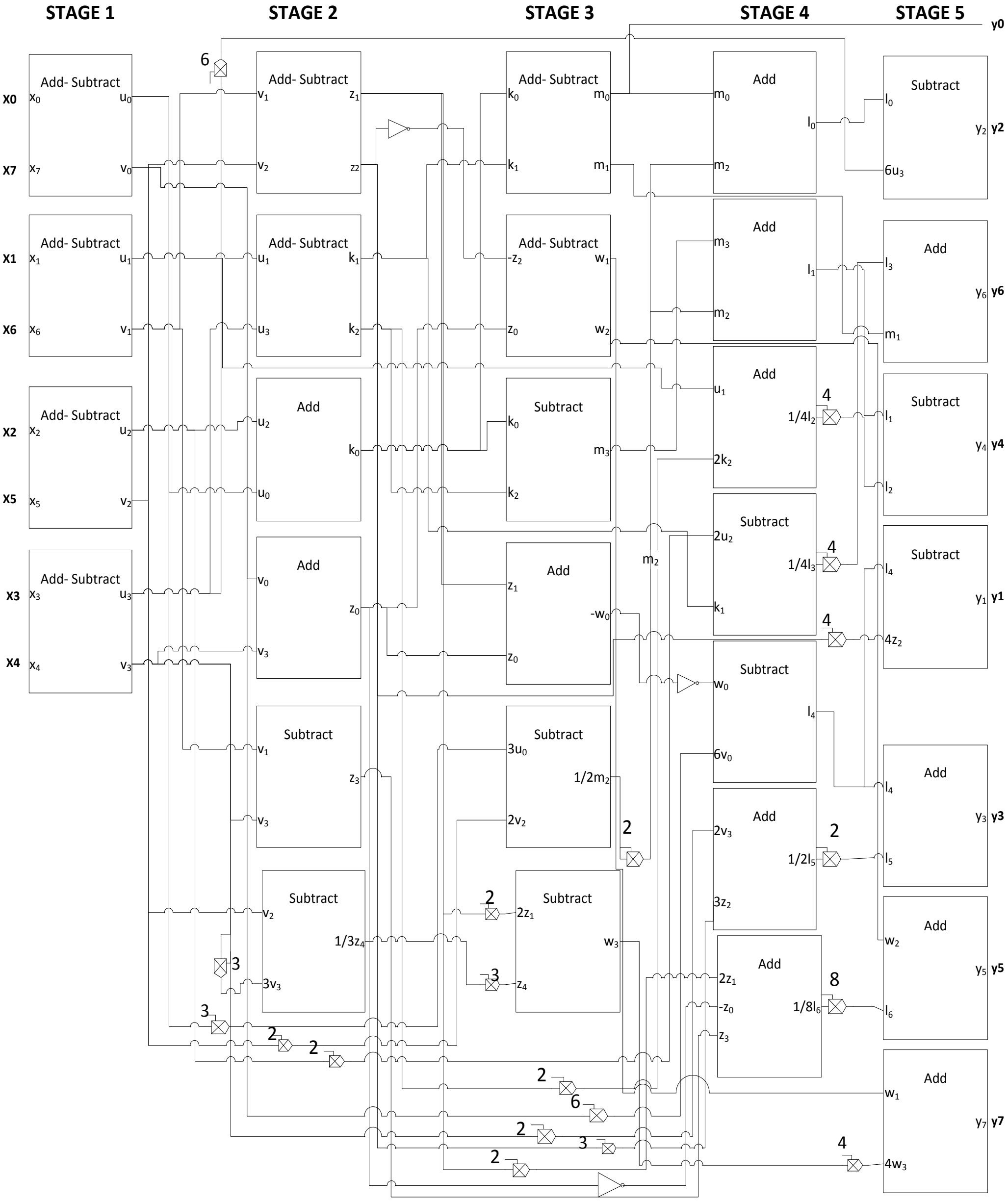


Figure 1. Architectural diagram of 1D integer DTT.

Table 1. 1D multiplier-free fast ITT.

Step 1	$u_o = x_0 + x_7$ $u_1 = x_1 + x_6$ $u_2 = x_2 + x_5$ $u_3 = x_3 + x_4$	$v_o = x_0 - x_7$ $v_1 = x_1 - x_6$ $v_2 = x_2 - x_5$ $v_3 = x_3 - x_4$
Step 2	$k_o = u_0 + u_2$ $k_1 = u_1 + u_3$ $k_2 = u_1 - u_3$ $z_4 = 3(v_2 - 3v_3)$	$z_o = v_0 + v_3$ $z_1 = v_1 - v_2$ $z_2 = v_1 + v_2$ $z_3 = v_1 - v_3$
Step 3	$m_o = k_0 + k_1$ $m_1 = k_0 - k_1$ $m_2 = 2(3u_0 - 2u_2)$ $m_3 = k_0 - k_2$	$w_0 = -(z_1 + z_0)$ $w_1 = -(z_0 + z_2)$ $w_2 = z_0 - z_2$ $w_3 = 2z_1 - z_4$
Step 4	$l_0 = m_0 + m_2$ $l_1 = m_3 + m_2$ $l_2 = 4(2k_2 + u_1)$ $l_3 = 4(2u_2 - k_1)$ $l_4 = w_0 - 6v_0$	$l_5 = 2(2v_3 + 3z_2)$ $l_6 = 8(2z_1 - z_0 + z_3)$
Step 5	$y_0 = m_0$ $y_1 = l_4 - 4z_2$ $y_2 = l_0 - 6u_3$ $y_3 = l_4 + l_5$	$y_4 = l_1 - l_2$ $y_5 = l_6 + w_2$ $y_6 = m_1 + l_3$ $y_7 = w_1 + 4\, w_3$

Table 2. Multiplication through shift and add operations.

***m* × 2 =**	**= <<*m***
***m* × 3 =**	**= <<*m* + *m***
***m* × 4 =**	**= <<<<*m***
***m* × 6 =**	**= << (<<*m* + *m*)**
***m* × 8 =**	**= <<<<<<*m***

The 2-D DTT can be computed efficiently by using the proposed 1-D implementation first along the rows and then along the columns in two steps by using the separability property of the DTT as shown below.

$$\hat{\boldsymbol{\tau}} \boldsymbol{X} \hat{\boldsymbol{\tau}}' = \left(\hat{\boldsymbol{\tau}} \left(\hat{\boldsymbol{\tau}} \boldsymbol{X} \right)' \right)' \tag{23}$$

The element-by-element multiplication between the ITT, namely, $\left(\hat{\boldsymbol{\tau}} \boldsymbol{X} \hat{\boldsymbol{\tau}}'\right)$ and factorization matrix $\hat{\boldsymbol{Q}}$ in (22), is incorporated into the quantization stage.

4. Variable Quantization

The transformation of the image data is followed by the crucial step of "quantization" in lossy image compression during which each block of the transformed image is divided by a quantization matrix and the result is rounded off to nearest integer leading to better compression, but loss of accuracy as well. The quantization matrix employed in image compression using JPEG standard with cosine transform of size 8 × 8 is fixed and is given by [10]. Depending upon the level of compression desired, the fixed quantization matrix is multiplied by a quantization factor. Higher the quantization factor, higher is the amount of compression achieved. In conventional JPEG compression the entire image is quantized with the resulting matrix. As a result, all the blocks of the image have the same accuracy level which might not always be the best choice.

For variable quantization, the image blocks need to be differentiated as robust or sensitive to quantization based on their characteristics. The sensitive blocks are quantized with a smaller quantization matrix, whereas the resilient ones are quantized with a strong matrix,thus improving the quality of the image.

One of the simplest ways of discriminating between the image blocks is based on their last non-zero element position (nzp) [9] after transformation followed by the standard quantization procedure. This position when compared with a threshold can help in determining if the block is strong enough to withstand high levels of quantization. It is possible to maintain multiple thresholds and quantize each image block that satisfies a threshold condition with the corresponding matrix. To decompress the image, each block needs to be de-quantized using the same matrix with which it was quantized.

If the nzp of the image block is less than the threshold, it is deemed to be robust and the elements of the already quantized block are quantized by an additional factor. This factor is always chosen to be a multiple of “2”, *i.e.*, 2p (p = 1, 2, 3..), such that the additional quantization *i.e.*, division, can be accomplished by “p” simple right shift operations.If none of the threshold conditions are met then “p” is set to “0” and no additional quantization is carried on the block. Thus, “p = 0” indicates the corresponding image block to be sensitive to quantization as shown in **Figure 2**.

For the current set of images chosen, the focus is to achieve high compression with least possible error. It has been found that the threshold value of “15” along with p = 1, *i.e.*, an additional quantization factor of “2”, gives the best results in an 8 × 8 matrix. Hence, if the last non-zero element position of a transformed image block after being quantized with fixed quantization matrix is less than “15”, then all the elements of the already quantized block undergo a single right shift to be additionally quantized by factor “2”. If the condition is not satisfied, there is no additional quantization. Thus, the blocks robust to quantization undergo double the quantization than the sensitive blocks.

However, it must be noted that without the variable quantization information, the receiver cannot distinguish the blocks that underwent additional quantization from the normal blocks. Thus, the variable quantization factor needs to be transmitted along with the compressed data to ensure faithful reconstruction. This can be achieved by the encoding technique described below.

Encoding: The block resulting from quantization is rearranged into a linear sequence and is subjected to encoding to reduce the amount of information to be transmitted. Usually, *run length encoding* (RLE) is employed which could be followed by any other encoding technique to further condense the data if desired.

Conventional RLE takes advantage of the fact that image data after transformation and quantization has significant number of zeros. RLE encodes by giving only the non-zero elements of the sequence. Each non-zero element (E) is preceded by the number of zeros (Z) before itself in the sequence. End of the sequence is signified by setting both Z and E to zero. It can be seen that (E) can never be zero unless it is the end of the sequence.

In the modified RLE scheme, the last two codes (Z and E) at the end of block sequence, not only indicate the end of the block, but also its corresponding quantization factor. So at the end of the block sequence, E is set to zero indicating end of sequence and Z is set to “p” giving the variable quantization code. **Figure 3** is a comparison of the outcomes of conventional and modified RLE carried on a data sequence.

At the decoder, the block is reconstructed and the variable quantization code ***“p”*** is identified as (Z) whose corresponding (E) is set to zero at the end of the sequence. The resulting block undergoes “p” left shifts which is equivalent to being multiplied by “2p” before being de-quantized with the fixed matrix as shown in **Figure 4**.

Table 3 gives a comparison of the arithmetic complexity of the proposed fast ITT with those for the fast DCTs in [3]-[5] and ICT in [8]. It also presents the arithmetic complexity when variable quantization is used for ICT and ITT.

The crucial observation of the comparison in **Table 3** is that the computationally intensive multiplication operations are completely eliminated by using the proposed ITT implementation for both 1D and 2D cases. There is a minor increase in the number of addition operations for ITT when compared to the rest of the approaches. Additionally, ITT employs shift operations which are not used by the other implementations. The nominal increase in complexity arising from an increase in the number of addition and shift operations for ITT is justified by the elimination of multiplication operations which consume more resources. It is seen that the variable quantization technique demands only an increase in the number of shift operations to compensate for the improved compression performance. The minor increase in computational complexity due to increase in the number of shift operations can be totally justified by the increased compression achieved with no additional bit transmis-

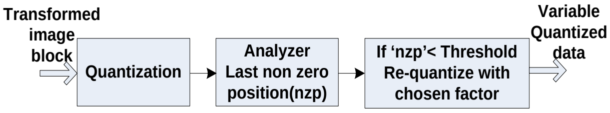

Transformed image block
Quantization
Analyzer Last non zero position(nzp)
If ‘nzp’< Threshold Re-quantize with chosen factor
Variable Quantized data

Figure 2. Variable quantization scheme.

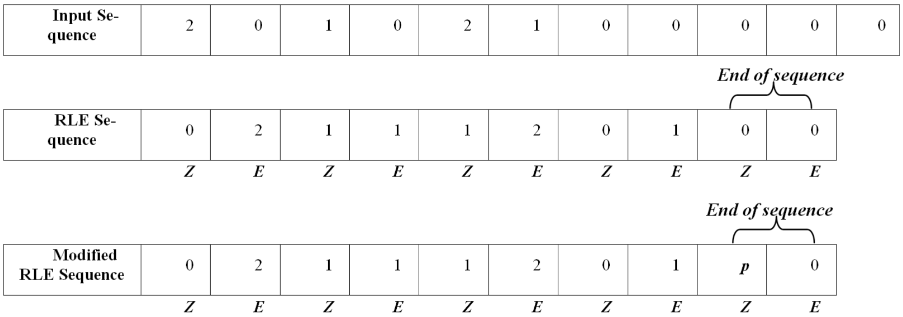

Input Sequence 2 0 1 0 2 1 0 0 0 0 0
End of sequence
RLE Sequence 0 2 1 1 1 2 0 1 0 0
Z E Z E Z E Z E Z E
End of sequence
Modified RLE Sequence 0 2 1 1 1 2 0 1 p 0
Z E Z E Z E Z E Z E

Figure 3. Conventional and modified RLE encoding.

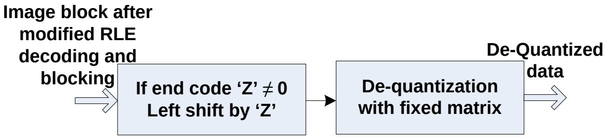


Image block after modified RLE decoding and blocking
If end code 'Z' ≠ 0 Left shift by 'Z'
De-quantization with fixed matrix
De-Quantized data

Figure 4. Variable de-quantization scheme.

Table 3. Comparison of computational complexity.

Input type	Operation	DCT [3]	DCT [4]	DCT [5]	ICT [8]	ITT [7]	ICT (var)	ITT (var)
8 point 1D array	Add	NA	28	29	NA	44	NA	44
	Multiply	NA	11	5	NA	0	NA	0
	Shift	NA	NA	NA	NA	29	NA	29 + 8p
8 point 2D array	Add	454	448	464	448	704	448	704
	Multiply	94	176	144	352	0	352	0
	Shift	6	NA	NA	128	464	128+64p	464 + 64p

In the above table, "2p" is the quantization factor for the image block and (var) represents transform with variable quantization.

sion by using variable quantization.

For the purpose of comparison between ITT and ICT, compression and decompression of image blocks of size 8 × 8 are carried out using a scheme similar to the one followed by Joint Photographic Expert Group (JPEG) format [11] as shown in **Figure 5** and **Figure 6**, respectively.

5. Simulation

MATLAB software is chosen to simulate the image compression scheme illustrated in Section IV using both ITT and ICT with and without variable quantization. A variety of input images are selected to analyze and compare the compression results obtained using the two transforms. The quality of compression achieved for an image depends as much on its spatial and frequency content as on the technique adopted for compression. Measures like s*pectral activity measure* **(*SAM*)** and *spatial frequency measure* **(*SFM*)** are employed to assess the structural nature of the image [12]. SAM gives a measure of image predictability in the spectral domain, and is defined as the ratio of the arithmetic mean to the geometric mean of the DFT coefficients.

It is given by

$$\mathrm{SAM}=\frac{\frac{1}{M\cdot N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left|F(i,j)\right|^{2}}{\left\{\prod_{i=0}^{M-1}\prod_{j=0}^{N-1}\left|F(i,j)\right|^{2}\right\}^{\frac{1}{M\cdot N}}} \tag{24}$$

where M and N are the numbers of pixels in the horizontal and vertical directions and $F(i, j)$ is the $(i,j)^{\mathrm{th}}$ DFT

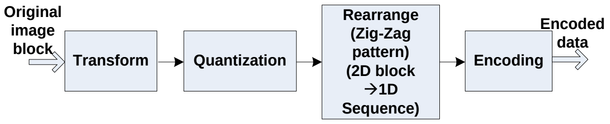


Figure 5. Compression scheme.

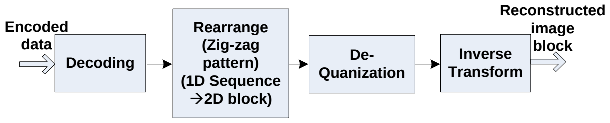

Encoded data
Decoding
Rearrange (Zig-zag pattern) (1D Sequence →2D block)
De-Quanization
Inverse Transform
Reconstructed image block

Figure 6. De-Compression scheme.

coefficient of the original image

SAM of an image is directly proportional to the predictability of the image and has a dynamic range of [1, ∞).

SFM indicates the overall activity level in a picture, defined by the *row frequency*, $R_f$and the *column frequency*, C_f.

It is given by

$$\text{SFM} = \sqrt{R_f^2 + C_f^2} \tag{25}$$

$$R_f = \sqrt{\frac{1}{M \cdot N} \sum_{i=1}^{M} \sum_{j=2}^{N} \left(x_{ij} - x_{ij-1} \right)^2} \tag{26}$$

$$C_f = \sqrt{\frac{1}{M \cdot N} \sum_{j=1}^{M} \sum_{i=2}^{N} \left(x_{ij} - x_{i-1j} \right)^2} \tag{27}$$

where, x_{ij} represents the $(i,j)^{th}$ pixel value of the original image and M and N are the numbers of pixels in the horizontal and vertical directions.SFM is directly proportional to the number of high frequency components in the image.

In general, images with a large SFM and a small SAM are difficult to code, since they have high frequency content and are not very predictable.

The level of compression of the image is varied by adjusting the factor multiplying the quantization matrix used during the quantization phase of the compression. If the factor is high, the quality of the compression decreases, whereas the amount of compression increases. In order to rate the performance of the different compression techniques, compression performance curves are plotted between the evaluation parametersPSNR vs. bpp (bits per pixel) and RMSE vs. CR (compression ratio). In the PSNR vs. bpp plots, the higher the curve the better is the performance, whereas in the RMSE vs. CR plots, the lower the curve the better is the quality of compression.

In [12] [13], the authors have proposed a set of quality parameters composed of objective measurements for picture quality evaluation which are expressed as in **Table 4**.

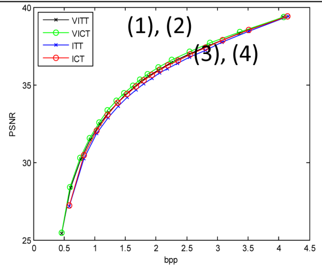
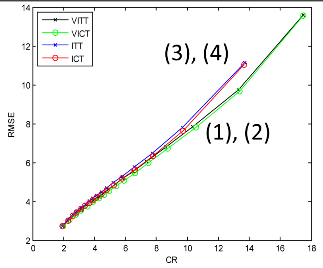
Ideally, when the reconstructed image is a close replica of the input image before compression, the correlation-based measures (NK, SC) should be close to unity, whereas the Human Visual System (HVS)-based measures (NAE, LMSE) and the difference-based measures (AD, MD) should be close to zero.

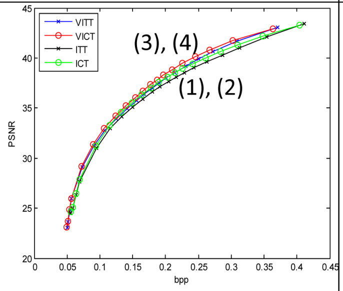
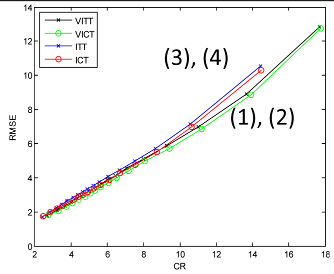
By varying the quantization factor, the image is subjected to different levels of compression. The PQE measures as well as the performance evaluation measures are calculated at each level of compression. Using the performance plots of PSNR vs. bpp and RMSE vs. CR along with the PQE measures, compression for each of the input images is interpreted. In the performance plots in **Figure 7**, the curves (1) and (2) in black and green correspond to ITT and ICT based compression techniques with variable quantization, whereas curves (3), and (4) in blue and red correspond to ITT and ICT based compression techniques with fixed quantization respectively

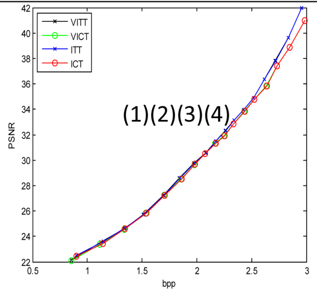
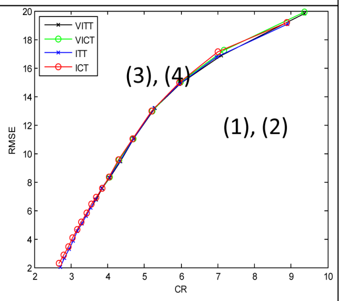
A. Lena

The popular image of Lena [14] has medium predictability and considerable amount of spectral activity. The PSNR and RMSE plots of ITT are almost overlapping with those of ICT in both the variable and non variable quantization cases. It can also be noted that the compression evaluation parameters, namely, CR and bpp are better for variable quantization. The rest of the quality evaluation measures discussed in **Table 4** are not too

	Input image	PSNR vsbpp plot	RMSE vs CR plot

(a)	
	Lena.jpg [14]
	Size = 512×512 pixels
	SAM = 1352.3
	SFM = 14.0436
			

(b)	
	Cameraman.tif [14]
	Size : 512×512 pixels
	SAM = 1794.7
	SFM = 14.36
			

(c)	
	Assorted patterns.jpg [14]
	Size : 500×500 pixels
	SAM = 19.53
	SFM = 63.6
			

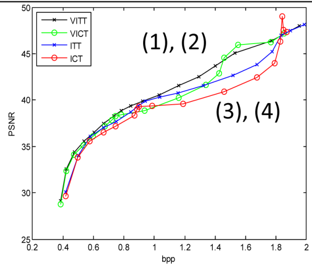
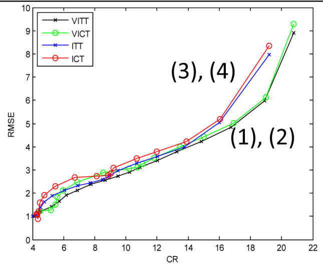
(d)	
	Intercede image.jpg [15]
	Size : 1280×720pixels
	SAM = 65.05
	SFM = 7.88
			

Figure 7. Various images with their compression performance plots.

deviant from their ideal values except for MD. There is an increase in MD when images are compressed with variable quantization using ITT or ICT. However, this can be ignored since an increase in MD does not necessarily mean overall poor quality of the image. From the RMSE vs. CR plot of Figure 7(a) it can be observed that RMSE curve for compression using variable quantization is significantly lower than that with normal quantization. Similarly, from the PSNR vs. bpp plot of Figure 7(a) it is observed that the PSNR is higher at higher compression levels with variable quantization using either ITT or ICT than that with normal quantization. From this analysis, it can be inferred that compression using variable quantization is beneficial for images with moderate predictability and considerable frequency content.

B. Cameraman

The image of Cameraman [14] is highly predictable. The RMSE and the PSNR plots in Figure 7(b) convey that the performance of ICT is better than that of ITT in this case. The rest of the PQE measures are almost same

Table 4. Objective PQE measures.

Correlation-based measures	
$NK = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} x_{i,j} x'_{i,j}}{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(x_{i,j}\right)^2}$	$SC = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(x_{i,j}\right)^2}{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(x'_{i,j}\right)^2}$
HVS-based measures	
$NAE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left\|x_{i,j} - x'_{i,j}\right\|}{\sum_{i=1}^{M}\sum_{j=1}^{N}\left\|x_{i,j}\right\|}$	$LMSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(x_{i,j} - x'_{i,j}\right)^2}{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(x_{i,j}\right)^2}$
Difference-based measures	
$AD = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left\|x_{i,j} - x'_{i,j}\right\|$	$MD = \max\left(\left\|x_{i,j} - x'_{i,j}\right\|\right)$

Note: $x_{i,j}$ and $x'_{i,j}$ represent the (i, j)th pixel value of the original and reconstructed images, respectively. M is the number of rows in the image; N is the number of columns in the image.

for ITT and ICT, except that the difference parameters, namely, AD and MD of ICT are marginally better than that of ITT. So it can be inferred that ICT offers a slightly better compression performance than ITT.

It can also be noticed from the same plots that compression with variable quantization offers better results than that with fixed quantization using either ITT or ICT. It can also be stated that variable quantization offers a wider range of compression, without much deterioration in quality than that offered by standard quantization, as can be observed from the corresponding RMSE vs. CR plot. Thus, it can be deduced that a better compression performance can be attained by using variable quantization for images with considerable frequency content and ICT over ITT for images with high predictability.

C. Assorted Patterns

The assorted patterns image [14] has low SAM indicating unpredictability. It is observed that though the correlation-based measures and the HVS-based measures are close to their ideal values, they are not as close as they were for the more predictable images using either ITT or ICT. Similarly, the AD and MD values are also higher which can be attributed to the lower predictability of the image. It can be observed from **Figure 7(c)** that the RMSE plot of ITT is lower than that of ICT, whereas the PSNR plot is higher indicating a better quality of compression.

The plots also indicate that compression using variable quantization is very similar to that obtained using normal quantization as the curves for variable and non-variable quantization based compression overlap for ITT as well as ICT. This can be attributed to the high SFM value of the image, which indicates the presence of intricate detail. So abiding by the law of variable quantization, the majority of blocks cannot withstand high levels of quantization and so undergo usual quantization. Thus, it can be stated that variable quantization and standard quantization based compression techniques yield similar results for images with high SFM and that ITT-based compression is superior to that of the ICT-based compression.

D. Intercede image

The intercede image given in **Figure 7(d)** is the difference between two concurrent frames of a video of horses running in a beach [15]. As the image has less predictability, the PQE parameters are slightly deviant from their ideal parameters at high rates of compression. At any given CR, it can be observed from **Figure 7(d)** that the RMSE of ITT is lower than that of ICT and the PSNR curve of ITT is above that of ICT over a major range of bpp. Hence, it is logical to deduce that ITT is a better choice over ICT for compression of the image.

From the RMSE vs. CR plot it can be seen that compression with variable quantization offers better quality than that offered with standard quantization on account of its better PSNR and low RMSE even at high CR irrespective of the transform used. Moreover, PSNR of the proposed compression method is consistently better than that with the conventional method. Based on these observations, it can be inferred that variable ITT offers the best compression result followed by variable ICT, ITT and ICT at higher compression ratios for the intercede image.

6. Conclusions

In this work, ITT of size 8 × 8 has been realized using a multiplier-free fast implementation and its complexity has been shown to be less than that of popular fast DCT and ICT implementations. The nature of images more suited for compression using ITT and using ICT have been identified based on their SFM and SAM characteristics. Through simulations it has been shown that the compression performance of ITT is about the same as that of ICT in most of the cases. Though the compression performance of ITT can be slightly inferior to that of ICT for images with high predictability (SAM), it provides a better performance than ICT does for images with low SAM, as illustrated for inter-coding images (**Figure 7(d)**). This could find a major application in the field of video compression where the concept of interceding is used.

The performances of ITT and ICT with and without variable quantization have also been compared using the PQE parameters and the performance curves. When variable quantization is employed, the image blocks are discriminated based on their sensitivity to quantization. The blocks that are robust to quantization undergo double the quantization than the rest of the blocks. The additional computational complexity resulting from making the quantization variable has been shown to be negligible involving just a shift operation. The transmission overhead resulting from the requirement to transmit the variable quantization information has been avoided by the modified run length encoding that has been introduced. It has been found that the performance of variable quantization-based compression is either superior to or on par with that of the standardized compression technique for the images analyzed. Images with low or moderate frequency content were found to profit the most from variable quantization. This results in a better quality reconstructed image for a given compression ratio when variable quantization is employed. It can also be interpreted that higher compression ratios are possible for a fixed level of expected quality in the reconstructed image when variable quantization is used.

While the strength of fast DCT designs lie in exploiting trigonometric properties, fast ITT designs can be developed by manoeuvring its polynomial properties. Since polynomial theory is very well-developed, there is a possibility of developing more efficient ITT implementations.

The variable quantization technique proposed can be further enhanced by differentiating the image blocks into more than two sets based on their sensitivity to quantization. Further research may be carried out to optimize the threshold value used to distinguish between the image blocks based on the type of image being compressed.

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada and the Regroupement strategique en Microelectronique du Quebec (ReSMIQ), and is based on an invited paper presented at the Ninth International Conference on Information, Communications and Signal Processing, Tainan, Taiwan, December 2013.

References

[1] Mukundan, R., Ong, S.H. and Lee, P.A. (2001) Image Analysis by Tchebichef Moments. *IEEE Transactions on Image Processing*, **10**, 1357-1364. http://dx.doi.org/10.1109/83.941859

[2] Nakagaki, K. and Mukundan, R. (2007) A Fast 4 × 4 Forward Discrete Tchebichef Transform Algorithm. *IEEE Signal Processing Letters*, **14**, 684-687.

[3] Feig, E. and Winograd, S. (1992) Fast Algorithms for the Discrete Cosine Transform. *IEEE Transactions on Signal Processing*, **40**, 2174-2193. http://dx.doi.org/10.1109/78.157218

[4] Loeffler, C., Ligtenberg, A. and Moschytz, G.S. (1989) Practical Fast 1-D DCT Algorithms with 11 Multiplications. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, **2**, 988-991. http://dx.doi.org/10.1109/icassp.1989.266596

[5] Arai, Y., Agui, T. and Nakajima, M. (1988) A Fast DCT-SQ Scheme for Images. *Transactions-IEICE*, **E-71**, 1095-1097.

[6] Ishwar, S., Meher, P.K. and Swamy, M.N.S. (2008) Discrete Tchebichef Transform—A Fast 4 × 4 Algorithm and Its Application in Image/Video Compression. *IEEE International Symposium on Circuits and Systems*, Seattle, 18-21 May 2008, 260-263. http://dx.doi.org/10.1109/ISCAS.2008.4541404

[7] Prattipati, S., Iswar, S., Swamy, M.N.S. and Meher, P.K. (2013) A Fast 8 × 8 Integer Tchebichef Transform and Comparison with Integer Cosine Transform. *IEEE International Midwest Conferences on Circuits and Systems*, Colubus, 4-7 August 2013, 1294-1297.

[8] (2011) Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 7*th Meeting*, Geneva, 21-30 November 2011, 21-30.

[9] Kakarala, R. and Bagadi, R. (2009) A Method for Signalling Block-Adaptive Quantization in Baseline Sequential JPEG. *IEEE TENCON*, Singapore City, 23-26 January 2009, 1-6.

[10] Mukundan, R. (2006) Transform Coding Using Discrete Tchebichef polynomials. *Proceedings of IASTED International Conference on Visualization Imaging and Image Processing*, Palma de Mallorca, Spain, 29-31 August 2006, 270-275.

[11] Pennebaker, W.B. and Mitchell, J.L. (1992) JPEG Still Image Data Compression Standard. Kluwer Academic Publishers, Norwell.

[12] Grgic, S., Mrak, M. and Grgic, M. (2001) Comparison of JPEG Image Coders. *Proceedings of* 3*rd International Symposium on Video Processing and Multimedia Communications*, *VIPromCom*-2001, Zadar, Croatia, 13-15 June 2001, 79-85.

[13] Eskicioglu, A.M. and Fisher, P.S. (1995) Image Quality Measures and Their Performance. *IEEE Transactions on Communications*, **43**, 2959-2965. http://dx.doi.org/10.1109/26.477498

[14] (2013) Digital Image Processing Database. http://www.imageprocessingplace.com/root_files_V3/image_databases.htm

[15] (2013) Footage: Small World Productions, Inc.; Tourism New Zealand. Producer: Gary F. Spradling. Music: Steve Ball.