Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2012, Article ID 879626, 10 pages
doi:10.1155/2012/879626

Research Article

Improved Algorithm for ODCT Computation of a Running Data Sequence

S. Akhter, V. Karwal, and R. C. Jain

ECE Department, Jaypee Institute of Information Technology, Noida 201307, India

Correspondence should be addressed to S. Akhter, shamim.akhter@jiit.ac.in

Received 23 May 2012; Revised 16 October 2012; Accepted 3 November 2012

Academic Editor: Chi Ko

Fast windowed update algorithms capable of independently updating the odd discrete cosine transform (ODCT) and odd discrete sine transform (ODST) of a running data sequence are analytically developed. In this algorithm, to compute the ODCT coefficients of a real-time sequence, we do not require the ODST coefficients. Similarly, the ODST coefficients of the shifted sequence can be calculated without using ODCT coefficients. The running input data sequence is sampled using a rectangular window. However, this idea can be easily extended for other windows. The update algorithm derived herein can be used to compute the transform coefficients of the shifted sequence as new data points are available. The complexity of developed algorithm is $O(N)$. The validity of algorithm is tested by MATLAB simulations.

1. Introduction

The DCT transform [1] has been extensively used in various digital image coding schemes and image compression standards. It is known that in most practical cases, the DCT-based schemes outperform in terms of compression ratio. In any real-time data processing system, a new data point is input at every sampling instant. One method to find the DCT can be to compute it for the entire data vector or data matrix. Although this looks easy, it is a cumbersome task. Another method is to reduce the dataset and then compute DCT coefficients for a smaller dataset. Hence, a part of incoming data stream is sampled using window of size N, and its transform coefficient is computed. An example is a tapped delay line adaptive filter [2]. In this case as new samples arrive, the transform of the updated data is computed. One computationally attractive method given by Yip and Rao [3] uses the shift property of the discrete transform wherein the transform of the shifted data can be obtained from respective transforms (both DCT and DST) of the unshifted data plus the data points shifting in, That is, the transform of the updated sequence is obtained without having to perform complete N-point transform at every instant of time. However, this scheme of Yip and Rao [3] had a drawback of dependency between DCT and DST coefficients.

Update algorithm initially developed was limited to one-point update at a time [3, 4]. Later algorithms for r-point, $1 \leq r \leq N-1$, and update for DCT, and DST were developed [5, 6] but all these algorithms needed simultaneous update of DCT and DST coefficients, that is, to compute the updated DCT coefficients, the DST coefficients need to be calculated simultaneously and vice versa. Thereafter, independent update algorithms were developed in [5] based on second-order shift properties of DCT and DST coefficients, these algorithms were limited to one-point update for even discrete cosine transform (EDCT) and even discrete sine transform (EDST). This concept was extended to include r new points at a time for EDCT/EDST in [7–9]. The DCT and DST transforms of type I–IV form a group of-so-called even DCT/DST. These transforms are very well studied and a number of efficient implementation techniques exist in the literature.

Another group of transforms, namely, DCT/DST V–VIII, are called "Odd" DCT/DST. Not much work has been reported in developing efficient implementation techniques for odd DCT/DST computation [10]. In this paper, we give an efficient algorithm for real-time computation of transform coefficients capable of independently updating odd DCT and odd DST coefficients. We have derived r-point independent update algorithm for independently processing ODCT and ODST coefficients in the presence of rectangular window. The algorithms are analytically derived and the results verified with MATLAB simulations for various different values of sequence length N and different data points. Our interest in this work is based on new application of odd DCT/DST-VI/VII transforms with a possible application in ISO/IEC/ITU-T high efficiency video coding (HEVC) standards [10].

To compute the transform of an infinite input sequence, first a suitable sequence of length N is extracted. To calculate the effect of inclusion of r new points, update algorithms can be used for fast computation of transform coefficient of the updated sequence. In this paper, we have derived the update algorithms for ODCT type-II and ODST type-II that have the capability of independently updating the ODCT and ODST coefficients. These algorithms would be useful in applications where we need only the ODCT coefficients or only the ODST coefficients. These algorithms provide lower complexity implementation as compared to the previously available algorithms as we do not need to calculate and store values of both ODST and ODCT coefficients.

The ODCT-II and ODST-II of input signal $f(x)$ of length N is defined by

$$C(k) = \frac{2}{\sqrt{2N-1}} P_k \sum_{x=0}^{N-1} P_{x+1} f(x) \cos\frac{2x+1}{2N-1}k\pi \quad \text{for } k = 0,1,\ldots,N-1,$$
$$S(k) = \frac{2}{\sqrt{2N-1}} \sum_{x=0}^{N-2} f(x) \sin\frac{2x+1}{2N-1}k\pi \quad \text{for } k = 1,\ldots,N-1, \tag{1}$$

where

$$P_j = \begin{cases} \dfrac{1}{\sqrt{2}}, & \text{for } j = 0 \text{ or } N \\ 1, & \text{otherwise.} \end{cases} \tag{2}$$

$C(k)$ and $S(k)$ are the ODCT and ODST coefficients, respectively.

Modifying the original equations of ODCT and ODST to include a sequence pointer n [5], we get

$$C(n,k) = \frac{2}{\sqrt{2N-1}} P_k \sum_{x=0}^{N-1} P_{x+1} f(n-N+x) \cos\frac{2x+1}{2N-1}k\pi, \tag{3}$$

$$S(n,k) = \frac{2}{\sqrt{2N-1}} \sum_{x=0}^{N-2} f(n-N+x) \sin\frac{2x+1}{2N-1}k\pi. \tag{4}$$

The input data $f(x)$ in the above expressions is indexed from $f(n-N)$ to $f(n-1)$.

This paper is organized as follows. In Section 2, the r-point independent update algorithm is derived analytically for the ODST independent ODCT-II, in the presence of rectangular window. ODCT independent ODST-II update algorithm can be derived similarly and results is given for brevity. In Section 3, MATLAB implementation is discussed followed by Section 4 where comparison of computational complexities is given in terms of number of multiplications and additions followed by Section 5 that lists the conclusion.

2. Derivation of the r-Point Update Algorithm for Independent ODCT-II

Let $C(n+r,k)$ for $k = 0,1,\ldots,N-1$ represent the updated ODCT-II coefficients including the effect of r new data points. From (3), we get

$$C(n+r,k) = \frac{2}{\sqrt{2N-1}} P_k \sum_{x=0}^{N-1} P_{x+1} f(n-N+x+r) \times \cos\frac{2x+1}{2N-1}k\pi. \tag{5}$$

Defining $a = 2/\sqrt{2N-1}$ and fragmenting the summation into two parts, $x = 0$ to $N-r-1$ and $x = N-r$ to $N-1$ yields,

$$\begin{aligned} C(n+r,k) = {} & aP_k \sum_{x=0}^{N-1-r} P_{x+1} f(n-N+x+r) \times \cos\frac{2x+1}{2N-1}k\pi \\ & + aP_k \sum_{x=N-r}^{N-1} P_{x+1} f(n-N+x+r) \times \cos\frac{2x+1}{2N-1}k\pi. \end{aligned} \tag{6}$$

The term P_{x+1} is equal to unity in the first summation term for the given range. Substituting $y = x + r$ in first term and $i = x + r - N$ in the second term, we get

$$\begin{aligned} C(n+r,k) & = aP_k \sum_{y=r}^{N-1} f(n-N+y) \cos\left(\frac{2(y-r)+1}{2N-1}k\pi\right) \\ & + aP_k \sum_{i=0}^{r-1} P_{i-r+N+1} f(n+i) \cos\left(\frac{2(i-r+N)+1}{2N-1}k\pi\right). \end{aligned} \tag{7}$$

Breaking the 1st term into two summations, we get

$$\begin{aligned} C(n+r,k) & = aP_k \sum_{y=0}^{N-1} f(n-N+y) \cos\left(\frac{2(y-r)+1}{2N-1}k\pi\right) \\ & - aP_k \sum_{y=0}^{r-1} f(n-N+y) \cos\left(\frac{2(y-r)+1}{2N-1}k\pi\right) \\ & + aP_k \sum_{i=0}^{r-1} P_{i-r+N+1} f(n+i) \cos\left(\frac{2(i-r+N)+1}{2N-1}k\pi\right). \end{aligned} \tag{8}$$

Therefore,

$$\begin{aligned}
&C(n+r,k)\\
&\quad= aP_k\sum_{y=0}^{N-2} f(n-N+y)\cos\left(\frac{2(y-r)+1}{2N-1}k\pi\right)\\
&\qquad+ aP_k(-1)^k f(n-1)\cos\frac{2\pi rk}{2N-1}\\
&\qquad- aP_k\sum_{y=0}^{r-1} f(n-N+y)\cos\left(\frac{2(y-r)+1}{2N-1}k\pi\right)\\
&\qquad+ aP_k\sum_{i=0}^{r-1} P_{i-r+N+1} f(n+i)\cos\left(\frac{2(i-r+N)+1}{2N-1}k\pi\right).
\end{aligned} \tag{9}$$

Introducing term P_{y+1} in the first summation since it is equal to unity for the given range, we get

$$\begin{aligned}
&C(n+r,k)\\
&\quad= aP_k\sum_{y=0}^{N-2} P_{y+1} f(n-N+y)\cos\left(\frac{2(y-r)+1}{2N-1}k\pi\right)\\
&\qquad+ aP_k(-1)^k f(n-1)\cos\frac{2\pi rk}{2N-1}\\
&\qquad- aP_k\sum_{y=0}^{r-1} f(n-N+y)\cos\left(\frac{2(y-r)+1}{2N-1}k\pi\right)\\
&\qquad+ aP_k\sum_{i=0}^{r-1} P_{i-r+N+1} f(n+i)\cos\left(\frac{2(i-r+N)+1}{2N-1}k\pi\right).
\end{aligned} \tag{10}$$

After modifying the 1st summation term, we have

$$\begin{aligned}
&C(n+r,k)\\
&\quad= aP_k\sum_{y=0}^{N-1} P_{y+1} f(n-N+y)\cos\left(\frac{2(y-r)+1}{2N-1}k\pi\right)\\
&\qquad+ aP_k(-1)^k\left(1-\frac{1}{\sqrt{2}}\right) f(n-1)\cos\frac{2\pi rk}{2N-1}\\
&\qquad- aP_k\sum_{y=0}^{r-1} f(n-N+y)\cos\left(\frac{2(y-r)+1}{2N-1}k\pi\right)\\
&\qquad+ aP_k\sum_{i=0}^{r-1} P_{i-r+N+1} f(n+i)\cos\left(\frac{2(i-r+N)+1}{2N-1}k\pi\right).
\end{aligned} \tag{11}$$

Expanding the first summation term yields

$$\begin{aligned}
&C(n+r,k)\\
&\quad= aP_k\sum_{y=0}^{N-1} P_{y+1} f(n-N+y)\cos\frac{(2y+1)k\pi}{2N-1}\cos\frac{2rk\pi}{2N-1}\\
&\qquad+ aP_k\sum_{y=0}^{N-1} P_{y+1} f(n-N+y)\sin\frac{(2y+1)k\pi}{2N-1}\sin\frac{2rk\pi}{2N-1}\\
&\qquad+ aP_k(-1)^k\left(1-\frac{1}{\sqrt{2}}\right) f(n-1)\cos\frac{2rk\pi}{2N-1}\\
&\qquad- aP_k\sum_{y=0}^{r-1} f(n-N+y)\cos\left(\frac{2(y-r)+1}{2N-1}k\pi\right)\\
&\qquad+ aP_k\sum_{i=0}^{r-1} P_{i-r+N+1} f(n+i)\cos\left(\frac{2(i-r+N)+1}{2N-1}k\pi\right).
\end{aligned} \tag{12}$$

first term can be written as $C(n,k)\cos(2rk\pi/(2N-1))$. Modifying the 2nd term in summation from $y=0$ to $N-1$ as $y=0$ to $N-2$, we get

$$\begin{aligned}
&C(n+r,k)\\
&\quad= C(n,k)\cos\frac{2rk\pi}{2N-1}\\
&\qquad+ aP_k\sum_{y=0}^{N-2} P_{y+1} f(n-N+y)\\
&\qquad\qquad\times\sin\frac{(2y+1)k\pi}{2N-1}\sin\frac{2rk\pi}{2N-1}\\
&\qquad+ aP_k(-1)^k\left(1-\frac{1}{\sqrt{2}}\right) f(n-1)\cos\frac{2\pi rk}{2N-1}\\
&\qquad- aP_k\sum_{y=0}^{r-1} f(n-N+y)\cos\left(\frac{2(y-r)+1}{2N-1}k\pi\right)\\
&\qquad+ aP_k\sum_{i=0}^{r-1} P_{i-r+N+1} f(n+i)\cos\left(\frac{2(i-r+N)+1}{2N-1}k\pi\right).
\end{aligned} \tag{13}$$

It can be observed in 2nd term that P_{y+1} is unity for $y=0$ to $N-2$. Using (4), this term can be rewritten as follows:

$$\begin{aligned}
&C(n+r,k)\\
&\quad= C(n,k)\cos\frac{2rk\pi}{2N-1} + P_k S(n,k)\sin\frac{2rk\pi}{2N-1}\\
&\qquad+ (-1)^k aP_k\left(1-\frac{1}{\sqrt{2}}\right) f(n-1)\cos\frac{2rk\pi}{2N-1}
\end{aligned}$$

$$- aP_k \sum_{y=0}^{r-1} f(n-N+y)\cos\left(\frac{2(y-r)+1}{2N-1}k\pi\right)$$
$$+ aP_k \sum_{i=0}^{r-1} P_{i-r+N+1} f(n+i)\cos\left(\frac{2(i-r+N)+1}{2N-1}k\pi\right). \quad (14)$$

Replacing

$$A_r = \cos\frac{2rk\pi}{2N-1}, \qquad B_r = \sin\frac{2rk\pi}{2N-1}. \quad (15)$$

Putting $y = r - x - 1$ in the 1st summation term, and $i = r - x - 1$ in the 2nd summation term, we get

$$\begin{aligned} C(n+r,k) &= A_r C(n,k) + B_r P_k S(n,k) \\ &\quad + (-1)^k aP_k A_r\left(1-\frac{1}{\sqrt{2}}\right) f(n-1) \\ &\quad - aP_k \sum_{x=0}^{r-1} f(n-N-1+r-x)\cos\frac{2x+1}{2N-1}k\pi \\ &\quad + aP_k(-1)^k \sum_{x=0}^{r-1} P_{N-x} f(n-1+r-x)\cos\frac{2xk\pi}{2N-1}. \end{aligned} \quad (16)$$

Rewriting the last summation term from $x = 1$ to $r - 1$ and adding $x = 0$th term, we get,

$$\begin{aligned} C(n+r,k) &= A_r C(n,k) + B_r P_k S(n,k) \\ &\quad + (-1)^k aP_k A_r\left(1-\frac{1}{\sqrt{2}}\right) f(n-1) \\ &\quad - aP_k \sum_{x=0}^{r-1} f(n-N-1+r-x)\cos\frac{2x+1}{2N-1}k\pi \\ &\quad + aP_k(-1)^k \sum_{x=1}^{r-1} P_{N-x} f(n-1+r-x)\cos\frac{2xk\pi}{2N-1}. \\ &\quad + aP_k(-1)^k P_N f(n-1+r) \end{aligned} \quad (17)$$

P_{N-x} in the 5th term is equal to unity for the given range of summation. Rewriting the 5th term with the summation limits changed from $x = 1$ to $r - 1$ as $x = 0$ to $r - 1$ and subtracting $x = 0$th term, we get

$$\begin{aligned} C(n+r,k) &= A_r C(n,k) + B_r P_k S(n,k) \\ &\quad - aP_k \sum_{x=0}^{r-1} f(n-N-1+r-x)\cos\frac{2x+1}{2N-1}k\pi \\ &\quad + aP_k(-1)^k \sum_{x=0}^{r-1} f(n-1+r-x)\cos\frac{2xk\pi}{2N-1} \\ &\quad + (-1)^k aP_k A_r\left(1-\frac{1}{\sqrt{2}}\right) f(n-1) \\ &\quad + (-1)^k aP_k\left(\frac{1}{\sqrt{2}}-1\right) f(n-1+r). \end{aligned} \quad (18)$$

The expression given in (18) shows that for finding ODCT coefficient of r-point updated signal, we require both ODCT and ODST coefficients of previous data points. We need a way to derive ODST independent update equation. For making independent ODCT update, we calculate $C(n-r,k)$ as follows, which is ODCT coefficient of r-point update of previous data points:

$$C(n-r,k) = \frac{2}{\sqrt{2N-1}} P_k \sum_{x=0}^{N-1} P_{x+1} f(n-N+x-r) \times \cos\frac{2x+1}{2N-1}k\pi. \quad (19)$$

Substituting $y = x - r$ in the above equation yields

$$C(n-r,k) = aP_k \sum_{y=-r}^{N-r-1} P_{y+r+1} f(n-N+y)\cos\left(\frac{2(y+r)+1}{2N-1}k\pi\right). \quad (20)$$

Breaking the summation into two terms, we get

$$\begin{aligned} C(n-r,k) &= aP_k \sum_{y=0}^{N-r-1} P_{y+r+1} f(n-N+y)\cos\left(\frac{2(y+r)+1}{2N-1}k\pi\right) \\ &\quad + aP_k \sum_{y=-r}^{-1} P_{y+r+1} f(n-N+y)\cos\left(\frac{2(y+r)+1}{2N-1}k\pi\right). \end{aligned} \quad (21)$$

Changing the first summation limit to $y = 0$ to $N - r - 2$, and adding $y = (N - r - 1)$th term, we get,

$$\begin{aligned} C(n-r,k) &= aP_k \sum_{y=0}^{N-r-2} P_{y+r+1} f(n-N+y)\cos\left(\frac{2(y+r)+1}{2N-1}k\pi\right) \\ &\quad + aP_k \sum_{y=-r}^{-1} P_{y+r+1} f(n-N+y)\cos\left(\frac{2(y+r)+1}{2N-1}k\pi\right) \\ &\quad + aP_k\frac{1}{\sqrt{2}} f(n-r-1)(-1)^k. \end{aligned} \quad (22)$$

P_{y+r+1} is unity in the range of 1st summation as well as in 2nd summation. Introducing P_{y+1} in the first term which is unity in the given range, we get,

$$\begin{aligned} C(n-r,k) &= aP_k \sum_{y=0}^{N-r-2} P_{y+1} f(n-N+y)\cos\left(\frac{2(y+r)+1}{2N-1}k\pi\right) \\ &\quad + aP_k \sum_{y=-r}^{-1} f(n-N+y)\cos\left(\frac{2(y+r)+1}{2N-1}k\pi\right) \\ &\quad + aP_k\frac{1}{\sqrt{2}} f(n-r-1)(-1)^k. \end{aligned} \quad (23)$$

Changing the summation of the first term from $y = 0$ to $N - r - 1$ and subtract $y = (N - r - 1)$th term, we get

$$\begin{aligned}
C(n-r,k) &= aP_k \sum_{y=0}^{N-r-1} P_{y+1} f(n-N+y) \cos\left(\frac{2y+2r+1}{2N-1}k\pi\right) \\
&\quad + aP_k \sum_{y=-r}^{-1} f(n-N+y) \cos\left(\frac{2(y+r)+1}{2N-1}k\pi\right) \\
&\quad + aP_k f(n-r-1)\left(\frac{1}{\sqrt{2}}-1\right)(-1)^k.
\end{aligned} \tag{24}$$

Solving further yields

$$\begin{aligned}
C(n-r,k) &= aP_k \sum_{y=0}^{N-1} P_{y+1} f(n-N+y) \cos\frac{2y+1+2r}{2N-1}k\pi \\
&\quad - aP_k \sum_{y=N-r}^{N-1} P_{y+1} f(n-N+y) \cos\left(\frac{2(y+r)+1}{2N-1}k\pi\right) \\
&\quad + aP_k \sum_{y=-r}^{-1} f(n-N+y) \cos\left(\frac{2(y+r)+1}{2N-1}k\pi\right) \\
&\quad + aP_k f(n-r-1)\left(\frac{1}{\sqrt{2}}-1\right)(-1)^k.
\end{aligned} \tag{25}$$

Expanding 1st term, substituting $y = N - x - 1$ in the 2nd term and $y = -(1+x)$ in the 3rd term, we have

$$\begin{aligned}
C(n-r,k) &= aP_k \sum_{y=0}^{N-1} P_{y+1} f(n-N+y) \cos\frac{2y+1}{2N-1}k\pi \ \cos\frac{2rk\pi}{2N-1} \\
&\quad - aP_k \sum_{y=0}^{N-1} P_{y+1} f(n-N+y) \sin\frac{2y+1}{2N-1}k\pi \ \sin\frac{2rk\pi}{2N-1} \\
&\quad - aP_k \sum_{x=0}^{r-1} P_{N-x} f(n-1-x) \\
&\qquad \times \cos\left(\frac{2(N-1-x+r)+1}{2N-1}k\pi\right) \\
&\quad + aP_k \sum_{x=0}^{r-1} f(n-N-1-x) \cos\left(\frac{2x+1-2r}{2N-1}k\pi\right) \\
&\quad + aP_k f(n-r-1)\left(\frac{1}{\sqrt{2}}-1\right)(-1)^k.
\end{aligned} \tag{26}$$

Using the definition of ODCT and ODST as given in (3) and (4) and substituting $A_r = \cos(2rk\pi/(2N-1))$, $B_r = \sin(2rk\pi/(2N-1))$ yields

$$\begin{aligned}
C(n-r,k) &= A_r C(n,k) - P_k B_r S(n,k) \\
&\quad + aP_k f(n-r-1)\left(\frac{1}{\sqrt{2}}-1\right)(-1)^k \\
&\quad - (-1)^k aP_k \sum_{x=0}^{r-1} P_{N-x} f(n-1-x) \\
&\qquad \times \left(A_r \cos\frac{2xk\pi}{2N-1} + B_r \sin\frac{2xk\pi}{2N-1}\right) \\
&\quad + aP_k \sum_{x=0}^{r-1} f(n-N-1-x) \\
&\qquad \times \left(A_r \cos\frac{2x+1}{2N-1}k\pi + B_r \sin\frac{2x+1}{2N-1}k\pi\right).
\end{aligned} \tag{27}$$

Adding (18) and (27) results in

$$\begin{aligned}
C(n+r,k) &= 2A_r C(n,k) - C(n-r,k) \\
&\quad + aP_k(-1)^k\left(\frac{1}{\sqrt{2}}-1\right) \\
&\qquad \times [f(n-1-r) - f(n-1)A_r + f(n-1+r)] \\
&\quad + aP_k \sum_{x=0}^{r-1} [f(n-N-1-x)A_r - f(n-N-1+r-x)] \\
&\qquad \times \cos\frac{2x+1}{2N-1}k\pi \\
&\quad + aP_k(-1)^k \\
&\qquad \times \sum_{x=0}^{r-1} [f(n-1+r-x) - P_{N-x} f(n-1-x)A_r] \\
&\qquad \times \cos\frac{2xk\pi}{2N-1} \\
&\quad - aP_k B_r (-1)^k \sum_{x=0}^{r-1} P_{N-x} f(n-1-x) \sin\frac{2xk\pi}{2N-1} \\
&\quad + aP_k B_r \sum_{x=0}^{r-1} f(n-N-1-x) \sin\frac{2x+1}{2N-1}k\pi \\
&\qquad \text{for } k = 0, 1, \ldots, N-1,
\end{aligned} \tag{28}$$

where $C(n-r,k)$ represent the ODCT coefficients of the previous time-step sequence, $C(n,k)$ represents the ODCT coefficients in the current time step sequence, and $C(n+r,k)$ represents ODCT coefficients of r-point updated sequence.

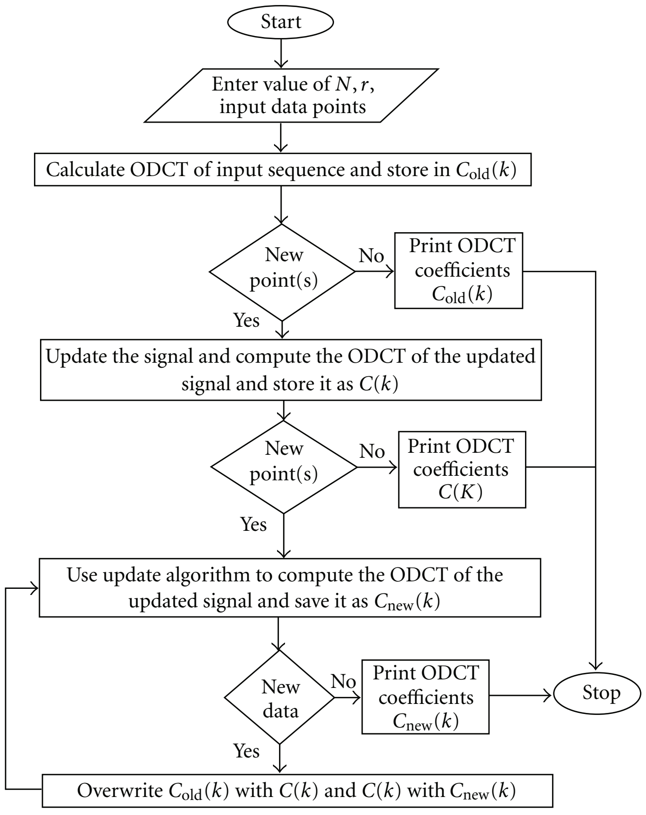
Equation (28) yields ODCT-II independent update equation that is independent of the ODST coefficients hence it can be used for fast computation of ODCT coefficients in the presence of r new data points. The above derived algorithm can be used to compute the ODCT transform for any value of sequence length N and value of r ranging from 1 to $N-1$. From (28), it can be easily seen that while computing the ODCT coefficients of updated sequence, we require ODCT coefficients of the two previous times-step sequence.

Therefore, there will be two stages for computing coefficients for updated sequence. One is the preprocessing stage where we calculate coefficients of two previous time-step data points using the conventional definition of the transform, and the second is the update stage where update algorithm is used to calculate coefficients of the shifted sequence. Once the pre-processing stage is over, the algorithm enters into update stage and the last two time-step coefficients are saved in $C(n-r,k)$ and $C(n,k)$. The algorithm remains in update stage thereafter. The conventional definitions are to be used only once in pre-processing stage and thereafter update algorithm is used each time new data points are shifted in the input sequence.

The signal point $f(n-N-1-x)$ represents the oldest data points shifted out and $f(n-N-1+r-x)$ represents recent data shifted out. The signal point $f(n-1+r-x)$ indicates the newest data points shifted in and $f(n-1-x)$ shows earlier data entered into. Analogous ODCT-II independent ODST-II fast update algorithms in the presence of r new data points for $1 \le r \le N-1$ can be similarly derived and the final result is listed below. The detailed derivation of ODST-II fast update algorithm is excluded for brevity as

$$\begin{aligned} S(n+r,k) &= 2A_r S(n,k) - S(n-r,k) - a(-1)^k B_r f(n-1) \\ &\quad + a\sum_{x=0}^{r-1}\left[f(n-N-1+r-x)+f(n-N+x-r)\right] \\ &\quad\quad \times \sin\frac{2x+1}{2N-1}k\pi \\ &\quad - a(-1)^k \sum_{x=1}^{r-1}\left[f(n-1+r-x)+f(n-1-r+x)\right] \\ &\quad\quad \times \sin\frac{2xk\pi}{2N-1} \quad \text{for } k=1,\ldots,N. \end{aligned} \tag{29}$$

$S(n-r,k)$ is the ODST coefficients of the previous time-step sequence, $S(n,k)$ is the ODST coefficients in the current time step, and $S(n+r,k)$ represents the updated ODST coefficients in the presence of r-points shifted in. In (29), the signal points $f(n-N-1+r-x)$ represent recent data shifted out and $f(n-N-r+x)$ represent the oldest data points shifted out. The signal point $f(n-1+r-x)$ indicates data point shifted in the previous time-step and $f(n-1-r+x)$ indicates newest data point shifted in.



Start
Enter value of N, r, input data points
Calculate ODCT of input sequence and store in $C_{\text{old}}(k)$
New point(s)
No
Print ODCT coefficients $C_{\text{old}}(k)$
Yes
Update the signal and compute the ODCT of the updated signal and store it as $C(k)$
New point(s)
No
Print ODCT coefficients $C(K)$
Yes
Use update algorithm to compute the ODCT of the updated signal and save it as $C_{\text{new}}(k)$
New data
No
Print ODCT coefficients $C_{\text{new}}(k)$
Stop
Yes
Overwrite $C_{\text{old}}(k)$ with $C(k)$ and $C(k)$ with $C_{\text{new}}(k)$

Figure 1: Flow chart of independent update algorithm.

Steps for computing ODCT r-point independent update are given in Figure 1. Signal $f(x-r)$represents the initial data points, $f(x)$ is the signal after r-point shift and $f(x+r)$ represents the signal after another r-point shift.

Independent update algorithm can be implemented in two major steps: one is the pre-processing step wherein we use the conventional ODCT definitions to compute the transforms and once we have the transform of two time steps then we can use the independent update algorithm, analytically derived earlier in this section. The pre-processing step is used only once to compute the two time-step transforms then the algorithm enters the update stage and remains in this stage thereafter.

(a) Preprocessing Step. Compute the ODCT coefficients values of $f(x-r)$ and $f(x)$ from the definition and save these values as $C_{\text{old}}(k)$ and $C(k)$, respectively.

(b) Update Step. This involves the computation of ODCT coefficients of r-point shifted data, $f(x+r)$, using update algorithm listed in (28).

Step 1. Using the values of $C_{\text{old}}(k)$ and $C(k)$ (computed in part (a)) and the new data pointed shifting in, we compute the ODCT coefficients of the new r-point shifted data using (28) and save it as $C_{\text{new}}(k)$.

Step 2. Overwrite $C_{\text{old}}(k)$ with $C(k)$, and $C(k)$ with $C_{\text{new}}(k)$.

```
root2 = sqrt(2.0);N = 8,r = 2,sign = −1;
root_2N = sqrt(2.0 ∗ N − 1.0);a = 2.0/root_2N;
for k = 0: N − 1;
if (mod(k, N)==0)
    pk = 1.0/root2;
else pk = 1.0;
end;
M = N + N − 1;
theta = 2 ∗ r ∗ k ∗ pi/M;Ar = cos(theta);Br = sin(theta);
sign = −sign;
p1 = 0.0;p2 = 0.0;p3 = 0.0;p4 = 0.0;p5 = 0.0
 for x = 0 : r − 1;
        angle1 = (x + x + 1) ∗ k ∗ pi/M;
        angle2 = (x + x) ∗ k ∗ pi/M;
        sin 1 = sin(angle1);
        sin 2 = sin(angle2);
        cos 1 = cos(angle1);
        cos 2 = cos(angle2);
 if (x==0)
        pk1 = 1.0/root2;
else
        pk1 = 1.0;
 end;
 p1 = p1 + pk1 ∗ (fnew(r − x)) ∗ sin 2;
 p2 = p2 + (fold(r − x)) ∗ sin 1;
 p3 = p3 + (−fold(2 ∗ r − x) + Ar ∗ fold(r − x)) ∗ cos 1;
 p4 = p4 + (fnew(2 ∗ r − x) − pk1 ∗ Ar ∗ fnew(r − x)) ∗ cos 2;
 end;
p5 = (1.0/sqrt(2.0) − 1) ∗ sign ∗ (f(N) − fnew(r) ∗ Ar + fnew(2 ∗ r));
term1 = a ∗ pk ∗ (−sign ∗ Br ∗ p1 + Br ∗ p2 + p3 + sign ∗ p4 + p5);
C_NEW(k + 1) = 2 ∗ Ar ∗ C(k + 1) − C_old(k + 1) + term1
end;
```

Algorithm 1

Repeat the update step for each time new r-point data is entered.

The flow chart for this algorithm is given in Figure 1. Similarly, the ODCT independent ODST independent update algorithm can be implemented in the presence of rectangular window.

3. MATLAB Implementation

The independent update algorithms derived in (28) and (29) for ODCT and ODST, respectively, were tested in MATLAB for different values of N and r. The code snippet is given in Algorithm 1 for $r = 2$ and $N = 8$.

In the program shown in Algorithm 1 program $C_{\text{old}}(k)$ represents the oldest ODCT coefficient of input signal say $F = [1\ 2\ 3\ 4\ 5\ 6\ 7\ 8]$. Two new data points "9" and "10" are shifted in, are the modified data array becomes $F = [3\ 4\ 5\ 6\ 7\ 8\ 9\ 10]$ and its coefficients are denoted by $C(k)$. This is the pre-processing stage. When two new data points "11" and "12" are shifted in, the coefficients $C_{\text{new}}(k)$ of the shifted sequence are calculated using the update algorithm derived in (28). Array "*fold*" is used for storing the four data points shifting out and array "*fnew*" is used for storing the four new data point shifting in.

After simulating the program shown in Algorithm 1, C_{NEW} value calculated using the update algorithm is found to be the same as that computed from definition of ODCT for the shifted sequence $F = [5\ 6\ 7\ 8\ 9\ 10\ 11\ 12]$. This verifies the algorithm.

Similarly, the code for ODST-II independent update algorithm was written to test the correctness of the derived algorithm.

4. Performance Analysis

The computation of running ODCT and ODST based on independent update algorithm is expected to provide certain advantages over simultaneous update algorithm proposed by Murthy and Swamy [2], and Sherlock and Kakad [6] for EDCT/EDST. The comparison is done on the basis of total number of operations (multiplications and additions) used for the computation of transform coefficients for r-point updated data. If ODCT-II is to be computed for $N = 8$ using basic definition as given by (3), then it requires 46 multiplication and 56 addition operations. Similarly for ODST-II computation from definition requires 33 multiplication and 38 addition operations.

The expression for simultaneous update of ODST-II is given below

$$\begin{aligned} S(n+r,k) &= A_r S(n,k) - B_r C(n,k) \\ &\quad + a\sum_{x=0}^{r-1} f(n-N-1+r-x)\sin\frac{2x+1}{2N-1}k\pi \\ &\quad - a(-1)^k \sum_{x=0}^{r-1} f(n-1+r-x)\sin\frac{2xk\pi}{2N-1} \\ &\qquad \text{for } k=1 \text{ to } N-1. \end{aligned} \tag{30}$$

Computation of ODST coefficients of the updated sequence, for sample length $N = 8$, using simultaneous update algorithm requires $14r+7$ multiplications and $14r$ additions. The above values are computed after expanding (30) for different values of r. We have done analysis for $r = 1$ and 2, and it can be extended for other values of r also.

For $r = 1$,

$$S(n+1,k) = A_1 S(n,k) - B_r C(n,k) + af(n-N)\sin\frac{\pi k}{2N-1}. \tag{31}$$

Computing the ODST coefficients $S(1)$ to $S(7)$ using (31) requires 21 multiplication and 14 addition operations. Similarly for $r = 2$, we have

$$\begin{aligned} S(n+2,k) &= A_2 S(n,k) - B_2 C(n,k) \\ &\quad + a\left[f(n-N+1)\sin\frac{\pi k}{2N-1} + f(n-N)\sin\frac{3\pi k}{2N-1}\right] \\ &\quad - a(-1)^k f(n)\sin\frac{2\pi k}{2N-1}. \end{aligned} \tag{32}$$

Computing the ODST coefficients $S(1)$ to $S(7)$ using (32) requires 35 multiplication and 28 addition operations. In general we require $14r + 7$ multiplication and $14r$ addition operation for computing all ODST coefficients for any value of r.

Similarly, computational requirement for simultaneous ODCT-II coefficients computation using (18) can be derived and analysis for $r = 1$ and 2 are discussed below.

For $r = 1$,

$$\begin{aligned} C(n+1,k) &= A_1 C(n,k) + B_1 P_k S(n,k) \\ &\quad - aP_k f(n-N)\cos\frac{\pi k}{2N-1} + (-1)^k aP_k \frac{1}{\sqrt{2}} f(n) \\ &\quad + (-1)^k aP_k\left(1-\frac{1}{\sqrt{2}}\right) f(n-1)A_1. \end{aligned} \tag{33}$$

Table 1: Comparison of power consumption.

Type	Multiplications (P_{cm})	Additions (P_{ca})
From definition	368	448
From simultaneous update algorithm	305	320
From independent update algorithm	201	320

Table 2: Comparison of multiplication operations required.

r	Simultaneous update	Independent update algorithm
1	$8N$	$5N$
2	$12N$	$7N$
3	$16N$	$9N$

For computing the ODCT coefficients $C(0)$ to $C(7)$ using (33), we require 37 multiplication and 32 addition operations. Similarly for $r = 2$, we have

$$\begin{aligned} C(n+2,k) &= A_2 C(n,k) + B_2 P_k S(n,k) - aP_k f(n-N+1) \\ &\quad \times \cos\frac{\pi k}{2N-1} \\ &\quad + (-1)^k aP_k \frac{1}{\sqrt{2}} f(n+1) + (-1)^k aP_k f(n)\cos\frac{2\pi k}{2N-1} \\ &\quad + (-1)^k aP_k\left(1-\frac{1}{\sqrt{2}}\right) f(n-1)A_2 - aP_k f(n-N) \\ &\quad \times \cos\frac{3\pi k}{2N-1}. \end{aligned} \tag{34}$$

Computing the ODCT coefficients $C(0)$ to $C(7)$ using (34) requires 51 multiplication and 48 addition operations. In general, for computing ODCT coefficients, we require $14r + 23$ multiplication and $16r + 16$ addition operation. It is to be noted that simultaneous update requires both ODCT and ODST coefficients to be updated simultaneously, so overall requirement is $28r + 30$ multiplications and $30r + 16$ additions plus multiplications and additions operations that are required during the pre-processing stage.

Multiplication and addition operation required for ODCT-II independent update algorithm are given below.

For $r = 1$,

$$\begin{aligned} C(n+1,k) &= 2A_1 C(n,k) - C(n-1,k) + aP_k(-1)^k \frac{1}{\sqrt{2}} f(n) \\ &\quad + aP_k(-1)^k\left(\frac{1}{\sqrt{2}}-1\right) f(n-2) - aP_k(-1)^k \\ &\quad \times \left(\sqrt{2}-1\right) f(n-1)A_1 \\ &\quad + aP_k\cos\frac{k\pi}{2N-1}\left[f(n-N-1) - f(n-N)\right] \\ &\qquad \text{for } k = 0,1,\ldots,N-1. \end{aligned} \tag{35}$$

Table 3: Comparison of memory requirement.

r	Coefficient		Input sequence		Transform domain data		Total memory requirement	
	Simultaneous update	Independent update	Simultaneous update	Independent update	Simultaneous update	Independent update	Simultaneous update	Independent update
1	$6N$	$5N$	$N+1$	$N+2$	$2N$	$2N$	$9N+1$	$8N+2$
2	$10N$	$7N$	$N+2$	$N+4$	$2N$	$2N$	$13N+2$	$10N+4$
3	$14N$	$9N$	$N+3$	$N+6$	$2N$	$2N$	$17N+3$	$12N+6$

In general, we require $14r + 25$ multiplications and $32r + 16$ additions for independent update algorithm for ODCT computation.

The comparative analysis for power consumption for different approaches for ODCT/ODST computation is given below. To compute ODCT-II coefficient of r-point shifted data using the definition requires using 46 multiplications and 56 additions operation, r times. For example, to compute ODCT coefficient for $N = 8$ data point using conventional definition for 7 time shifted data points requires 46 multiplications and 56 additions, eight times. If power consumption by one multiplication operation is P_{cm} and that by one addition operation is P_{ca}, then total power consumed by multiplications is $368P_{\mathrm{cm}}$ and that by additions is $448P_{\mathrm{ca}}$.

For the same sample length of $N = 8$ and for the same set of ODCT/ODST coefficient computation, $r = 7$ and hence simultaneous update algorithm require using 121 multiplications and 128 additions for ODCT update and 105 multiplications and 98 additions for ODST update. Additional 79 multiplications and 94 additions are required during the pre-processing stage for calculating the ODCT/ODST coefficients from definition. Hence power consumed by multiplications is $305P_{\mathrm{cm}}$ and that by additions is $320P_{\mathrm{ca}}$.

Similarly, for the same sample length of $N = 8$ and for the same set of ODCT coefficient computation $r = 6$, independent update algorithm requires 201 multiplications and 320 additions. Hence power consumed by multiplications is $201P_{\mathrm{cm}}$ and that by additions is $320P_{\mathrm{ca}}$. Table 1 shows the power consumed during addition and multiplication operation, for $N = 8$ and $r = 7$. The computation of running ODCT and ODST based on independent update algorithm provides advantage over the simultaneous update for different value of r. Comparison for computational burden between independent update algorithm and simultaneous update algorithm for different value of r-point update is given in Table 2.

From (31), for $r = 1$, ODST computation requires $3N$ multiplication and $2N$ addition operation. Similarly from (33), for $r = 1$, ODCT computation requires $5N$ multiplication and $4N$ addition operations. So we can say that simultaneous update algorithm requires total $8N$ multiplication and $6N$ addition operations. Similarly from (35), we can see that ODCT independent update algorithm for $r = 1$ requires $5N$ multiplication and $6N$ addition operations. In general, the simultaneous update algorithm requires $(4r + 4)N$ multiplication operations while independent update algorithm requires $(2r + 3)N$ multiplication operations, whereas the additions operation requirement for both algorithms are $(4r + 2)N$. From Table 2, it can be seen that independent update algorithm requires less number of multiplications as compared to those required for simultaneous update algorithm, and hence we have less power consumption. Next we consider the memory requirement. Memory is needed in both the algorithms to store the following:

(a) the coefficients of the recursive equations,

(b) transform result,

(c) input signal samples.

Table 3 shows the memory requirement for both the update algorithms. It can be seen that simultaneous update requires more storage as compared to that required in independent update, since in simultaneous update algorithm both ODCT as well as ODST coefficients are to be stored. Thus it can be concluded that independent update algorithm is better in terms of storage requirement as compared to simultaneous update algorithm.

5. Conclusion

The use of recursive method is very appropriate for machine-based computation. The shift properties of DCT and DST transforms gives a set of recursive equations that can be used for updating DCT and DST coefficient of a running data sequence. We have given an efficient algorithm for computation of ODST-II and ODCT-II coefficients in the presence of rectangular window. In the given algorithm, the computation of ODCT coefficients does not require simultaneous computation of ODST coefficients. The recursive equations are analytically derived and tested using MATLAB. The algorithms developed in this paper are an improvement over existing update algorithms as it provides independence between ODCT and ODST coefficient's computation, thereby reducing computation and memory requirement. The algorithm derived is tested for different values of sample length N and r. The complexity of the analytically derived independent update algorithms is of the order of $O(N)$.

References

[1] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*, Academic Press, 1990.

[2] N. R. Murthy and M. N. S. Swamy, "On the computation of running discrete cosine and sine transform," *IEEE Transactions on Signal Processing*, vol. 40, no. 6, pp. 1430–1437, 1992.

[3] P. Yip and K. R. Rao, "On the shift property of DCT's and DST's," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 3, pp. 404–406, 1987.

[4] L. N. Wu, "Comments on 'On the shift property of DCT's and DST's," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 1, pp. 186–188, 1990.

[5] J. Xi and J. F. Chicharo, "Computing running DCT's and DST's based on their second-order shift properties," *IEEE Transactions on Circuits and Systems I*, vol. 47, no. 5, pp. 779–783, 2000.

[6] B. G. Sherlock and Y. P. Kakad, "Transform domain technique for windowing the DCT and DST," *Journal of the Franklin Institute*, vol. 339, no. 1, pp. 111–120, 2002.

[7] B. G. Sherlock, Y. P. Kakad, and A. Shukla, "Rapid update of odd DCT and DST for real-time signal processing," in *14th Signal Processing, Sensor Fusion, and Target Recognition*, vol. 5809 of *Proceedings of SPIE*, pp. 464–471, Orlando, Fla, USA, March 2005.

[8] V. Karwal, B. G. Sherlock, and Y. P. Kakad, "Windowed DST-independent discrete cosine transform for shifting data," in *Proceeding of the 20th International Conference on Systems Engineering*, pp. 252–257, Coventry, UK, September 2009.

[9] V. Karwal, *Discrete cosine transform-only and discrete sine transform-only windowed update algorithms for shifting data with hardware implementation [Ph.D. thesis]*, University of North Carolina at Charlotte, 2009.

[10] R. K. Chivukula and Y. A. Reznik, "Fast computing of discrete cosine and sine transforms of type VI and VII," in *34th Applications of Digital Image Processing*, vol. 8135 of *Proceedings of SPIE*, San Diego, Calif, USA, August 2011.